資料１

2024年度 第１回 暗号技術検討会

令和７年３月２５日
９：００～１１：００
経済産業省本館１７階
第２特別会議室/ハイブリッド開催

議事次第

１．開会

２．議事
(1) 2024年度暗号技術評価委員会 活動報告について【報告】
(2) CRYPTREC暗号リスト仕様書の参照先変更について【報告】
(3) 耐量子計算機暗号ガイドライン／調査報告書の更新について【承認】
(4) 「量子コンピュータが共通鍵暗号の安全性に及ぼす影響」に関する外部評価報告書（案）について【承認】
(5) 2024年度暗号技術活用委員会 活動報告について【報告】
(6) 暗号鍵管理ガイダンス（Part 2）について【承認】
(7) 耐量子計算機暗号（PQC）への対応について【承認】
(8) 2025年度暗号技術評価委員会活動計画（案）について【承認】
(9) 2025年度暗号技術活用委員会活動計画（案）について【承認】
(10) 暗号技術検討会 2024年度 報告書（案）について【承認】
(11) その他

３．閉会

配付資料一覧

資料１	議事次第・配付資料一覧
資料２	暗号技術検討会 開催要綱（構成員・オブザーバ名簿）
資料３－１	2024年度 暗号技術評価委員会 活動報告
資料３－２	監視状況報告
資料３－３	CRYPTREC暗号リスト仕様書の参照先変更
資料３－４	2024年度暗号技術調査ワーキンググループ（耐量子計算機暗号）活動報告 （別紙）耐量子計算機暗号ガイドライン／調査報告書の更新 （別紙１）CRYPTREC暗号技術ガイドライン（耐量子計算機暗号）（案） （別紙２）CRYPTREC耐量子計算機暗号の研究動向調査報告書（案）
資料３－５	量子コンピュータが共通鍵暗号の安全性に及ぼす影響に関する技術動向調査 （別紙）2024年度外部評価報告書（量子コンピュータが共通鍵暗号の安全性に及ぼす影響）（案）
資料４－１	2024年度 暗号技術活用委員会 活動報告
資料４－２	2024年度暗号鍵管理ガイダンスWG 活動報告
資料４－３	暗号鍵管理ガイダンス（Part 2）の概要
資料４－４	暗号鍵管理ガイダンス（Part 2）
資料５	耐量子計算機暗号（PQC）への対応について
資料６	2025年度暗号技術評価委員会活動計画（案）
資料７	2025年度暗号技術活用委員会活動計画（案）
資料８	暗号技術検討会 2024年度 報告書（案）

以上

資料２

「暗号技術検討会」開催要綱

１　名　称

本検討会は「暗号技術検討会」（以下「検討会」という。）と称する。

２　開催の趣旨・目的

検討会は、デジタル庁統括官、総務省サイバーセキュリティ統括官及び経済産業省商務情報政策局長が有識者の参集を求め、暗号技術の普及による情報セキュリティ対策の推進を図る観点から、専門家による意見等を聴取することにより、デジタル庁、総務省及び経済産業省における施策の検討に資することを目的として開催する。

３　検討事項

（１）CRYPTREC暗号リスト掲載暗号技術の監視

（２）CRYPTREC暗号リスト掲載暗号技術の安全性及び信頼性確保のための調査・検討

（３）CRYPTREC暗号リストの改定に関する調査・検討

（４）CRYPTREC暗号リスト掲載暗号技術の普及促進及び暗号技術の利用促進・産業化に向けた取組の検討

（５）その他、システム全体のセキュリティ確保のために必要となる活動の検討等、暗号技術の評価及び利用に関すること

４　構成等

（１）検討会の構成は、別紙１のとおりとする。

（２）検討会には、座長１名を置く。

（３）座長は、構成員の互選により定める。

（４）座長は、検討会構成員の中から顧問及び座長代理を指名できる。

（５）構成員の任期は委嘱時に定めるものとし、再任を妨げないものとする。

５　運営

（１）座長は、検討会の議事を掌握する。

（２）座長が、緊急の理由によりやむを得ず不在となった場合、座長代理が座長に代わり議事を掌握する。

（３）関係する政府機関等で、座長が特に認めたものについては、オブザーバとして検討会に出席することができる。

（４）座長が必要と認めるときは、暗号技術の提案者、関連する利害関係者その他の参考人から意見を聴取することができる。

（５）座長は、検討会が調査する事項について特に専門的な調査を行う必要があると認めるときは、委員会等を置くことができる。
（６）座長は、必要があると認めるときは電子メールによる審議を行うことができる。なお、この審議を行った場合は、次の検討会において当該審議の結果を報告するものとする。
（７）その他検討会の運営に関し必要な事項は、座長が定めるところによる。

６　スケジュール

検討会は、年度内に１回以上開催する。

７　開催方法

検討会は、集合開催を原則とするが、必要に応じ、その一部又は全部をオンラインにより開催することができることとする。

８　議事・資料等の取扱い

別紙２のとおりとする。

９　庶　務

検討会の庶務は、デジタル庁デジタル社会共通機能グループ、総務省サイバーセキュリティ統括官室及び経済産業省商務情報政策局サイバーセキュリティ課において処理する。

（令和４年３月３０日　最終改訂）

別紙１

暗号技術検討会　構成員・オブザーバ名簿

2025.3.25現在

構成員

阿部　正幸	日本電信電話株式会社　社会情報研究所　フェロー
石井　義則	一般社団法人情報通信ネットワーク産業協会　常務理事
上原哲太郎	立命館大学　情報理工学部　情報理工学科　教授
國廣　　昇	筑波大学　システム情報系　教授
高木　　剛	東京大学　大学院情報理工学系研究科　数理情報学専攻　教授
田村　裕子	日本銀行　金融研究所　情報技術研究センター　企画役
手塚　　悟	慶應義塾大学　グローバルリサーチインスティテュート　特任教授
本間　尚文	東北大学　電気通信研究所　教授
松井　　充	三菱電機株式会社　開発本部　主席技監
松浦　幹太	東京大学　生産技術研究所　教授
松本　　勉	国立研究開発法人産業技術総合研究所　フェロー 横浜国立大学　先端科学高等研究院　上席特別教授
松本　　泰	特定非営利活動法人日本ネットワークセキュリティ協会　フェロー
向山　友也	一般社団法人テレコムサービス協会　技術・サービス委員会　副委員長
吉田　博隆	国立研究開発法人産業技術総合研究所 サイバーフィジカルセキュリティ研究センター　研究チーム長
渡邊　　創	国立研究開発法人産業技術総合研究所 サイバーフィジカルセキュリティ研究センター　副研究センター長

（五十音順、敬称略）

オブザーバ

内閣官房内閣サイバーセキュリティセンター　内閣参事官
個人情報保護委員会事務局　参事官
警察庁　長官官房　技術企画課　情報セキュリティ対策室長
総務省　自治行政局　住民制度課長
総務省　自治行政局　住民制度課　マイナンバー制度支援室長
法務省　民事局　商事課長
外務省　大臣官房　情報通信課長
財務省　大臣官房　文書課　業務企画室長
文部科学省　大臣官房　政策課　サイバーセキュリティ・情報化推進室長
厚生労働省　大臣官房参事官（サイバーセキュリティ・情報システム管理担当）
経済産業省　イノベーション・環境局　国際電気標準課長
防衛省　整備計画局　サイバー整備課　AI・サイバーセキュリティ政策調整官
国立研究開発法人情報通信研究機構　サイバーセキュリティ研究所長
国立研究開発法人産業技術総合研究所　サイバーフィジカルセキュリティ研究センター　首席研究員
独立行政法人情報処理推進機構　セキュリティセンター長
一般財団法人日本情報経済社会推進協会　デジタルトラスト評価センター　副センター長
公益財団法人金融情報システムセンター　監査安全部長

別紙２

暗号技術検討会の公開について

１　会議の公開について

（１）民間企業の暗号技術（既製品を含む）の解読方法等について議論を行う可能性があり、当事者又は第三者の権利、利益や公共の利益を害するおそれがあるため、検討会は原則非公開とする。

（２）検討会の出席者は、検討会において知り得た情報で、当事者又は第三者の権利、利益や公共の利益を害するおそれがあるものについては、検討会の出席者及び座長が特に認めた者以外に漏えいしてはならないものとする。

２　検討会の資料の公開について

（１）検討会の資料については、原則公開とする。

（２）ただし、検討会の資料を公開することにより、当事者又は第三者の権利、利益や公共の利益を害するおそれがある場合は、検討会は資料の公開を延期又は非公開とすることができる。

（３）資料は、ホームページ（cryptrec.go.jp）への掲載その他の方法により公開するものとする。

３　議事概要の公開について

（１）議事概要については、原則公開とする。

（２）ただし、議事概要を公開することにより、当事者又は第三者の権利、利益や公共の利益を害するおそれがある場合は、議事概要の該当部分を削除した上で公開することができる。

（３）議事概要は、ホームページ（cryptrec.go.jp）への掲載その他の方法により公開するものとする。

資料３－１

2024 年度　暗号技術評価委員会活動報告（案）

1.　活動目的

CRYPTREC 暗号リストに掲載されている暗号技術や電子政府システム等で利用される暗号技術の安全性維持及び信頼性確保のために安全性及び実装に係る監視及び評価を行う。

2.　活動概要

2024 年度活動計画に沿って以下の内容を行った。

1)　暗号技術の安全性及び実装に係る監視及び評価

暗号技術の安全性に係る監視・評価 及び 実装に係る技術の監視・評価を実施した。

①　CRYPTREC暗号リストの監視

国際会議等で発表されるCRYPTREC暗号リストの安全性及び実装に係る技術（暗号モジュールに対する攻撃とその対策も含む）に関する監視を行い、会議やMLを通して報告した。

・　2024年度は、電子政府推奨暗号リストの安全性に懸念を持たせるような事態は生じていない。今年度実施の監視報告の詳細については、CRYPTREC Report 2024で報告する。

・　CRYPTREC暗号リスト仕様書の参照先変更について、参照先の確認を行い、更新がある参照先について旧バージョンとの差分を調査し、参照先の更新を実施した。

②　電子政府推奨暗号リストからの運用監視暗号リストへの降格、並びに、推奨候補暗号リスト及び運用監視暗号リストからの危殆化が進んだ暗号の削除に係る検討

CRYPTREC暗号リストの安全性に係る監視活動を継続的に行い、急速に危殆化が進んだ暗号技術やその予兆のある暗号技術の安全性について評価を行う。また、リストからの降格や削除、注釈の改訂が必要か検討を行う。

・　現時点では、電子政府推奨暗号リストからの降格、推奨候補暗号リストおよび運用監視暗号リストから削除は行っていない。

③　CRYPTREC注意喚起レポートの発行

CRYPTREC暗号リストの安全性及び実装に係る技術の監視活動を通じて、国際会議等で発表された攻撃等の概要や想定される影響範囲、対処方法について早期に公開することが望ましいと判断された場合、注意喚起レポートを発行する。

・　現段階では、注意喚起レポートの発行は行っていない。

④　推奨候補暗号リストへの新規暗号（事務局選出）の追加に係る検討

標準化動向に鑑み電子政府システム等での利用が見込まれると判断される暗号技術の追加を検討する。

・　追加が必要となる暗号技術は無かった。

⑤　新技術等に関する調査及び評価

将来的に有用になると考えられる技術やリストに関わる技術について、安全性・性能評価を行う。必要に応じて、暗号技術調査ワーキンググループによる調査・評価、または、外部評価による安全性・性能評価などを行う。

(ア)NISTのPQC標準化において第4ラウンド、および、署名に関する追加公募／選定が進行中であることから、引き続き、暗号技術調査ワーキンググループ（耐量子計算機暗号）（PQC WG）を設置して、耐量子計算機暗号に関する最新動向を把握する。また、「素因数分解の困難性に関する計算量評価」、「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新についてもPQC WGで検討し、更新を行う。

・　2023年度の暗号技術評価委員会において、調査報告書・ガイドラインの記載内容は、2024年9月30日までの情報を可能な限り調査して掲載する、および、NISTの標準化動向を調査しつつ、その内容を調査報告書、ガイドラインに反映すること等、調査報告書、および、ガイドラインの執筆方針が決定された。

・　執筆方針に従って、2024年度版調査報告書・ガイドラインを作成した。（資料3-4_別紙1、資料3-4_別紙2）

・　「素因数分解の困難性に関する計算量評価」、「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新を行った。2024年度については、大きな変動はなかった。

（イ）2023年度に承認された「大規模な量子コンピュータに対する共通鍵暗号系の安全性に関する動向調査」を実施した。

・　本動向調査は2019年度に行われた「量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価」の更新版となる。

・　「量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価」2024年度版の案を作成した。（資料3-5_別紙）

2)　暗号技術の安全な利用方法に関する調査（技術ガイドラインの整備、学術的な安全性の調査・公表等）

暗号技術を利用する際の技術面での注意点に関する調査、新技術の安全性・性能に関する調査・評価を行う。

- 2-(1)-⑤-(ア）の中の一部：2024年度版PQC調査報告書・ガイドラインの作成（資料3-4_別紙1、資料3-4_別紙2）

  PQCの利活用に関する調査を実施し、PQC調査報告書・ガイドラインに記載した。

- 2-(1)-⑤-(イ）との重複項目：大規模な量子コンピュータに対する共通鍵暗号系の安全性に関する動向調査

  量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価を実施した。

3.　スケジュールおよび各委員会における活動

1)　第 1 回暗号技術評価委員会（2024 年 7 月 9 日：オンライン）の活動内容

- PQC WG について、引き続き設置されることが承認された。そして、PQC WG の 2024 年度活動計画を審議し、原案のまま承認された。
- 「量子コンピュータが共通鍵暗号の安全性に及ぼす影響」について外部評価を行うことが審議され、承認された。
- 監視状況に関する報告が行われた。現在の CRYPTREC 暗号リストに掲載の技術には問題がないことが確認された。

2)　第 2 回暗号技術評価委員会（2025 年 3 月 3 日：オンライン）の活動内容

- （本委員会の後記載予定）

4.　評価委員会の構成（敬称略）

委員長	高木　剛	（東京大学）
委員	青木　和麻呂	（文教大学）
委員	岩田　哲	（名古屋大学）
委員	上原　哲太郎	（立命館大学）
委員	大東　俊博	（東海大学）
委員	國廣　昇	（筑波大学）
委員	四方　順司	（横浜国立大学）
委員	手塚　悟	（慶応義塾大学）
委員	花岡　悟一郎	（産業技術総合研究所）

委員	藤﨑　英一郎	（北陸先端科学技術大学院大学）
委員	本間　尚文	（東北大学）
委員	松本　勉	（産業技術総合研究所・横浜国立大学）
委員	松本　泰	（日本ネットワークセキュリティ協会）
委員	山村　明弘	（秋田大学）

資料３－２

監視状況報告

1. 監視活動報告

2024 年度第一回暗号技術評価委員会（2024 年 7 月 4 日）から 2024 年度第二回暗号技術評価委員会（2025 年 3 月 3 日）までに、表 1 に示す国際会議について各種調査を行い、暗号解読技術等に関する研究動向を収集した。

表 1 調査対象国際会議

学会名・会議名		開催国・都市	期間
PQCrypto 2024	The 15th International Conference on Post-Quantum Cryptography	イギリス・オックスフォード	2024 年 6 月 12 日~14 日
Crypto 2024	The 44th Annual International Cryptology Conference	アメリカ・サンタバーバラ	2024 年 8 月 18 日～22 日
TCC 2024	The 22nd Theory of Cryptography Conference 2024	イタリア・ミラノ	2024 年 12 月 2 日～6 日
Asiacrypt 2024	The 30th International Conference on the Theory and Application of Cryptology and Information Security	インド・コルカタ	2024 年 12 月 9 日～13 日

また、2024 年度第一回暗号技術評価委員会における対象学会である PKC2024 について、監視活動報告が為されなかった暗号解析の研究発表が存在した。対象となっている暗号は CRYPTREC 暗号リスト記載の暗号ではない。追加調査をおこなった結果についても、本監視状況報告で述べる。

表 2 PKC2024 についての詳細情報

学会名・会議名		開催国・都市	期間
PKC 2024	The 27th International Conference on Practice and Theory in Public Key Cryptography	オーストラリア、シドニー	2024 年 4 月 15 日～4 月 17 日

2. 解読技術等の動向

各国際会議における報告等より、具体的な暗号の攻撃に関する発表を抽出し、CRYPTREC 暗号リスト記載の暗号の安全性に直接関わる技術動向（2.1）およびその他の

注視すべき技術動向（2.2）について分析を行った。

2.1. CRYPTREC 暗号リスト記載の暗号に直接関わる解読技術動向

CRYPTREC 暗号リスト（電子政府推奨暗号リスト）掲載の暗号に関して報告する。

共通鍵暗号 AES については、Asiacrypt 2024 で 3 編の新しい解読論文が発表されている。特に 6 ラウンド AES-128 に対する識別攻撃が改良され、計算量が2^{84}から$2^{76.57}$まで削減された。報告された攻撃の成功確率は 60%である。また、13 ラウンド AES-256 に対する差分中間一致攻撃が改良され、時間計算量2^{240}、データ量2^{89}、メモリ量2^{144}に削減された。これは現状で最良の計算量である。また、TPKC と呼ばれる新種の衝突攻撃が報告された。これは 2 つの異なる秘密鍵による暗号化を行った結果、生成された 2 つの暗号文が衝突するという攻撃であり、特定平文に対する攻撃を Fixed-TPKC、攻撃者の選択した平文に対する攻撃を Free-TPKC と定められている。結果、2/5/6 ラウンドの AES-128/192/256 に対する Fixed-TPKC と、5/7/9 ラウンドの AES-128/192/256 に対する Free-TPKC が報告された。

また、ハッシュ関数 SHA2 および SHA3 について新たな攻撃が報告された。SHA2 については、Eurocrypt 2024 で報告された 31 ラウンド SHA-256 の衝突攻撃のメモリ計算量の改善が Asiacrypt 2024 で与えられた。結果として、現実的な計算量（64 スレッドを使用して約 1.2 時間）と小さいメモリ量（2^{10}オーダー）での衝突メッセージペアが発見された。この手法を 31 ラウンド SHA-512 に適用した結果、既存攻撃と比較して、時間計算量が$2^{115.6}$から$2^{94.7}$へ、メモリ計算量が$2^{77.3}$から2^{35}へ、それぞれ改善された。また SHA3 についての衝突攻撃が Crypto 2024 で発表された。特に攻撃可能なラウンド数が更新され、SHA3-384 については 5 ラウンドまで、SHAKE256 については 6 ラウンドまで、攻撃可能であることが発表された。なお、その他の SHA-3 インスタンスについての衝突攻撃についても既存計算量が削減されており、特に 4 ラウンド SHA3-512 と 5 ラウンド SHA3-224/SHA3-256/SHAKE128 に対して既存の衝突攻撃が改善された。

いずれも現状まだ十分なラウンド数のマージンがあり、現実的な攻撃とは至っていない。

2.1.1. 共通鍵暗号に関する解読技術

Key Collisions on AES and Its Applications [Asiacrypt 2024]

Kodai Taiyama, Kosei Sakamoto, Ryoma Ito, Kazuma Taka, Takanori Isobe

AES に対する鍵衝突攻撃と AES の Davies-Meyer ハッシュモード（AES-DM）に対する衝突攻撃への応用に関する論文である。ブロック暗号に対する鍵衝突とは、2 つの

異なる秘密鍵を使用して対象となる平文を暗号化した結果、生成された 2 つの暗号文が衝突するという概念である。既存研究では CIPHERUNICORN-A、MULTI2、SC2000 などに対する鍵衝突攻撃が報告されているが、AES に対する鍵衝突攻撃は報告されていない。また、ブロック暗号に対する鍵衝突攻撃は DM ハッシュモードに対する衝突攻撃へと応用可能であるが、AES-DM に対する（セミフリースタート）衝突攻撃もまた報告されていない。

本研究では、最初に新しいタイプの鍵衝突であるターゲット平文鍵衝突（TPKC: Target-Plaintext Key Collision）を定義する。TPKC は一般的な鍵衝突とよく似ており、2 つの異なる秘密鍵を使用してある特定の平文を暗号化した結果、生成された 2 つの暗号文が衝突するという概念である。TPKC はさらに Fixed-TPKC と Free-TPKC という 2 つの問題に分割され、前者は特定の平文が事前に与えられている条件下で鍵衝突を発見する問題、後者は特定の平文を自由に決められるという条件下で鍵衝突を発見する問題である。なお、Fixed-TPKC と Free-TPKC はそれぞれ DM ハッシュモードの衝突とセミフリースタート衝突に等価な概念となる。次に、これら 2 つの問題を解くための新しい自動解析ツールを開発する。このツールは、AES ライクなハッシュ関数に対する強力な解析手法の 1 つとして知られているリバウンド攻撃をベースとしている。具体的には、リバウンド攻撃を効率的に実行するために、グラフ理論と SAT ソルバーによる差分特性探索手法を組み合わせ、Fixed-TPKC と Free-TPKC の最適な衝突パターンを探索するツールとなっている。

提案したツールを AES に適用した結果、2/5/6 ラウンドの AES-128/192/256 に対する Fixed-TPKC、5/7/9 ラウンドの AES-128/192/256 に対する Free-TPKC を発見し、それぞれ AES-DM に対する衝突攻撃とセミフリースタート衝突攻撃に応用可能であることを示した。特に、9 ラウンド AES-256-DM に対するセミフリースタート衝突攻撃は理論的に2^{30}の計算量で実行可能であり、実際の衝突メッセージペアを示すことで提案ツールの有効性を証明した。

The Boomerang Chain Distinguishers: New Record for 6-Round AES [Asiacrypt 2024]

Xueping Yan, Lin Tan, Hong Xu, Wenfeng Qi

AES の識別攻撃に関する論文である。2016 年まで、AES の識別攻撃は高々4 ラウンドが限界であったものの、Eurocrypt 2017 で Grassi らは multiple-of-8 と呼ばれる性質を発見し、この性質を利用することで 5 ラウンド AES の識別攻撃を2^{32}の計算量で実行できることを示した。その後、Asiacrypt 2019 で Bardeh らは exchange 攻撃を提案し、6 ラウンド AES の識別攻撃を2^{84}の計算量で実行できることを示した。現状、AES の識別攻撃は Bardeh らの提案手法が最良であると知られている。

これらの既存研究に基づき、本研究では以下の 3 点について考察する。まず、5 ラウ

ンドAESに対する識別攻撃と比較すると、6ラウンドAESに対する識別攻撃は非常に多くの計算量が必要となるため、このギャップを埋めることが可能であるかという課題に対処する。次に、既存手法では適当的平文暗号文設定における広いデータ空間を有効活用できるという利点があり、この利点を最大限に有効活用した新しい解析手法を提案する。最後に、AESに対する強力な解析手法の1つであるブーメラン攻撃に着目し、既存手法では1つのブーメラン特性にのみ依存していたが、複数のブーメラン特性を利用して攻撃を改善できないかを検討する。

本研究では2種類のブーメラン識別攻撃を提案した。1つ目はRe-Boomerang手法と呼ばれ、2つの関連性の高いブーメラン特性を組み合わせることで識別攻撃を強化するものとなっている。この攻撃は切り詰めブーメラン攻撃とexchange攻撃を組み合わせることで実現可能である。2つ目はRe-Boomerang手法の拡張でBoomerang Chain手法と呼ばれる。Re-Boomerang手法では2つのブーメラン特性を利用したが、この2つのブーメラン特性の間にさらに複数のブーメラン特性を追加することで、攻撃にかかるデータ量の削減を目的としたものとなっている。これらの提案手法を6ラウンドAESに適用した結果、計算量を2^{84}から$2^{76.57}$まで削減することに成功し、AESの識別攻撃に関して新しい記録を達成した。なお、攻撃成功確率は60%である。

Generic Differential Key Recovery Attacks and Beyond [Asiacrypt 2024]

Ling Song, Huimin Liu, Qianqian Yang, Yincen Chen, Lei Hu, Jian Weng

ブロック暗号に対する差分攻撃と矩形攻撃（Rectangle Attack）の汎用的攻撃手法の提案論文である。これらの攻撃に関して多くの改善手法が提案されてきた。特に、Crypto 2023で提案された差分中間一致（差分MitM）手法がさらなる改善をもたらし、切り詰め差分MitM手法への拡張など、さらなる発展が進んでいる。特にこれらの攻撃における鍵推測戦略が異なることに着目する。差分攻撃では最初に差分を満たす可能性がある平文暗号文ペアを生成した後、いくつかの鍵ビットを推測するという戦略を採ることが一般的である。一方で、矩形攻撃では平文暗号文ペアまたは平文暗号文カルテットを生成する前にいくつかの鍵ビットを推測することが可能であり、これにより事前に誤ったペアまたはカルテットをフィルタリングできることから、効率的な鍵推測が可能となる。

本研究の貢献は以下の3点である。1つ目は、汎用的古典差分攻撃（GCDA: Generic Classical Differential Attack）と呼ばれる新しい攻撃手法の開発である。差分攻撃における鍵推測の順序を改善し、誤った平文暗号文ペアを可能な限り事前にフィルタリングすることで、計算量の削減が可能となった。2つ目は、一般化差分MitM攻撃（GDMA: Generalized Differential MitM Attack）と呼ばれる新しい攻撃手法の開発であり、差分攻撃における最も強力な差分MitM手法を改善することに成功した。具体的には、差分 MitM 手法はある特定の鍵推測戦略のみをサポートしているが、任意の鍵推測戦略

をサポートするような汎用的手法へと拡張した。3 つ目は、汎用的矩形 MitM 攻撃（GRMA: Generic Rectangle MitM Attack）と呼ばれる新しい攻撃手法の開発であり、差分 MitM 手法を矩形攻撃に組み込みことで新しい矩形攻撃手法を実現した。

3 種類のブロック暗号（AES-256、KATAN-32、SKLINNYe-64-256v2）に適用することで提案手法の有効性を実証した。AES-256 に対しては GCDA と GDMA を適用し、関連鍵設定の下で 12/13 ラウンド AES-256 の計算量を改善した。例えば、13 ラウンド AES-256 に対しては、計算量2^{240}、データ量2^{89}、メモリ量2^{144}で攻撃が実行可能であり、既存の差分 MitM 攻撃の結果を大幅に改善する結果となった。その他、KATAN-32 に対しては GDMA、SKINNYe-64-256v2 に対しては GRMA が適用され、それぞれ既存結果を改善することに成功した。

2.1.2. ハッシュ関数に関する解読技術

Probabilistic Linearization: Internal Differential Collisions in up to 6 Rounds of SHA-3 [Crypto 2024]

Zhongyi Zhang, Chengan Hou, Meicheng Liu

SHA-3 の衝突攻撃に関する論文である。SHA-3 の衝突攻撃に関する研究では主に差分攻撃と内部差分攻撃をベースとした手法が提案されている。前者の手法は FSE 2012 で提案されたターゲット差分アルゴリズム（TDA: Target Difference Algorithm）をベースに、後者は FSE 2013 で提案されたターゲット内部差分アルゴリズム（TIDA: Target Internal Difference Algorithm）をベースに発展してきた。これらのアルゴリズムでは SHA-3 の S-box を線形化し、線形化された方程式から連立方程式を組み立てた後、この連立方程式を解く、という手順となっている。既存の線形化手法（linearization）では連立方程式を決定論的に組み立てるため、線形方程式の数が多すぎる傾向となり、大きいサイズの capacity を持ち、かつ安全性レベルの高い SHA-3 インスタンス（例えば、ラウンド数の多い SHA3-384、SHA3-512、SHAKE256、など）において TDA/TIDA を適用したとしても、現実的な時間内に連立方程式を解くことが困難となる。

この問題を解決するために、本研究では最大差分密度部分空間（maximum difference density subspace）と呼ばれる新しい概念を導入した確率的線形化手法（probabilistic linearization）を提案するとともに、この提案手法を TIDA に導入してアルゴリズムの一般化を行った。従来手法とは異なり、提案手法では連立方程式を確率論的に組み立てることが可能となり、結果として連立方程式における線形方程式の数を減らすことが可能となる。これは衝突を引き起こすためのデータ量の削減に繋がる。確率論的な手法であるため、連立方程式の解が必ずしも正しい入力差分を示すとは限らない。このため、複数の連立方程式を組み立て、これらの解に正しい入力差分が含まれているかを判定することで攻撃を成功させることが可能となる。

提案手法を全ての SHA-3 インスタンスに適用した。結果として、SHA3-384 と SHAKE256 に対する衝突攻撃では攻撃可能ラウンド数を更新することができ、それぞれ 5 ラウンドと 6 ラウンドまで攻撃が可能であることを示した。また、その他の SHA-3 インスタンスについては既存の計算量を削減することに成功した。具体的には、4 ラウンド SHA3-512 と 5 ラウンド SHA3-224/SHA3-256/SHAKE128 に対して既存の衝突攻撃を改善した。

The First Practical Collision for 31-Step SHA-256 [Asiacrypt 2024]

Yingxin Li, Fukang Liu, Gaoli Wang, Xiaoyang Dong, Siwei Sun

SHA-256 と SHA-512 の衝突攻撃に関する論文である。Eurocrypt 2013 で Mendel らは SHA-256 の衝突攻撃が 64 ステップ中 31 ステップまで実行可能であることを示した。この攻撃は$2^{65.5}$の計算量と2^{34}のメモリ量を必要とする。その後、Eurocrypt 2024 で Li らは Mendel らの 31 ステップ SHA-256 の衝突攻撃を改善した。Li らの攻撃は$2^{49.8}$の計算量と2^{48}のメモリ量を必要とする。現状において Li らの攻撃が最良であるものの、依然として 31 ステップ SHA-256 の衝突攻撃はメモリ量の観点で現実的であるとは言えない。また、計算量を$\mathcal{T}$、メモリ量を$\mathcal{M}$、ワードサイズをn（SHA-256 は$n = 32$）とすると、これらの既存攻撃は$\mathcal{T} \times \mathcal{M} \approx 2^{3n}$という条件を満たす必要があり、Li らの攻撃は最良のタイムメモリトレードオフを実現している。このため、Li らの手法でメモリ量を削減して、計算量を増加させる場合、現実的な時間内での衝突発見がより困難となる。このような制約事項を取り除き、31 ステップ SHA-256 の衝突攻撃に対して現実的な計算量とメモリ量で実行可能な攻撃手法を開発することが本研究のモチベーションとなる。

本研究では 31 ステップ SHA-256 に対するメモリ効率の高い攻撃手法を新たに開発する。新しい攻撃手法も既存手法と同様に中間一致技術を使用する。既存手法では、内部状態における 3 つのワード(A_{-1}, A_{-2}, A_{-3})を同時に一致させるという条件があり、この条件が上記の制約事項に影響を及ぼしていた。そこで、提案手法では、1 つのワードA_{-1}のみを一致させるという条件に緩和し、残りの 2 つのワード(A_{-2}, A_{-3})を on-the-fly で有効性チェックを行うこととした。その結果、最適な場合において$2^{1.5n}$の計算量を達成しつつ、メモリ量を$2^{0.5n}$まで削減することに成功した。提案手法を 31 ステップ SHA-256 に適用した結果、現実的な計算量（64 スレッドを使用して約 1.2 時間）と無視できるくらい小さいメモリ量（2^{10}オーダー）で衝突メッセージペアを発見した。同様に、提案手法を 31 ステップ SHA-512 に適用した結果、計算量を$2^{115.6}$から$2^{94.7}$へ、メモリ量を$2^{77.3}$から$2^{35.2}$へ、それぞれ改善することに成功した。

2.2. その他の注視すべき技術動向

共通鍵暗号については、差分線形攻撃や中間一致攻撃、およびそれらに用いられる共通鍵暗号解読プリミティブがいくつか進展し、認証 Ascon などを含む様々な共通鍵暗号解読論文が発表されている。

公開鍵暗号・署名においては、NIST 耐量子計算機暗号標準化において大きな動向があったので、ここで簡潔に報告する。

2024 年 8 月 13 日、NIST は 3 つの耐量子計算機暗号（CRYSTALS-Kyber、CRYSTALS-Dilithium、Sphincs+）の標準を定め、それに従い、これらの名前を変更した。CRYSTALS-Kyber は ML-KEM（Module-Lattice-Based Key-Encapsulation Mechanism）と名付けられ、FIPS 203 にて標準化された。CRYSTALS-Dilithium は ML-DSA（Module-Lattice-Based Digital Signature Algorithm）と名付けられ、FIPS 204 にて標準化された。Sphincs+ は SLH-DSA（Stateless Hash-Based Digital Signature Algorithm）と名付けられ、FIPS 205 にて標準化された。FALCON も FIPS 206 にて標準化される予定であり、ドラフトがリリースされている。

また 2024 年 10 月 24 日、追加の電子署名公募がラウンド 2 に移行し、候補の数が 14 件まで絞られた。現在の候補は、HAWK、CROSS、LESS、Mirath、MQOM、PERK、RYDE、SDitH、MAYO、QR-UOV、SNOVA、UOV、SQIsign、FAEST である。

以上の電子署名の標準化に関連して、例年と比べ、耐量子計算機電子署名に関する攻撃論文が多く見られた。攻撃論文の中には、以上の候補に関する安全耐性について報告しているものもある。詳細については、§2.2.2 以降を参考せよ。

より詳しくは、§2.1 で報告した論文以外にも、使用される機会が多い、もしくは今後多くなると予想される暗号に関する解析報告として、次の論文が発表された。

2.2.1. 共通鍵暗号に関する解読技術

Generic MitM Attack Frameworks on Sponge Constructions [Crypto 2024]

Xiaoyang Dong, Boxin Zhao, Lingyue Qin, Qingliang Hou, Shun Zhang, Xiaoyun Wan

スポンジ構造型ハッシュ関数に対する中間一致（MitM: Meet-in-the-Middle）攻撃手法を用いた原像攻撃と衝突攻撃に関する論文である。MitM 攻撃手法は Merkle-Damgard 構造型ハッシュ関数に対して幅広く適用されてきた実績があるものの、スポンジ構造型ハッシュ関数に対する適用例はあまり多くない。また、Merkle-Damgard 構造型ハッシュ関数に対する MitM 攻撃手法をスポンジ構造型ハッシュ関数に対して直接的な応用が難しいという問題があった。

本研究ではこのような問題を解決し、スポンジ構造型ハッシュ関数に適用可能な

MitM 原像攻撃と MitM 衝突攻撃のための汎用的な攻撃フレームワークを提案した。本フレームワークは NIST LWC 標準暗号の Ascon-Hash、NIST PQC 標準暗号の SPHINCS⁺-Haraka、ISO/IEC 標準暗号の PHOTON と SPONGENT、などの様々なスポンジ構造型ハッシュ関数に適用され、攻撃フレームワークとしての有効性が示された。具体的には次のとおりである。Ascon-Hash に対し、原像攻撃は 12 ラウンド中 5 ラウンドまで、衝突攻撃は 4 ラウンドまで実行可能であることが示された。なお、原像攻撃は初めての成果であり、衝突攻撃は従来手法による 2 ラウンドの攻撃を改善した。また、SPHINCS⁺-Haraka に対し、5 ラウンド中 4 ラウンドまでの原像攻撃が示された。これは従来手法による 3.5 ラウンドの攻撃を改善した。PHOTON と SPONGENT、その他のスポンジ構造型ハッシュ関数に関しても同様、初めての成果を示す、もしくは従来手法の改善に成功している。

Revisiting Differential-Linear Attacks via a Boomerang Perspective with Application to AES, Ascon, CLEFIA, SKINNY, PRESENT, KNOT, TWINE, WARP, LBlock, Simeck and SERPENT [Crypto 2024]

Hosein Hadipour, Patrick Derbez, Maria Eichlseder

共通鍵暗号プリミティブの差分線形識別子を効率的に探索するための汎用的かつ自動化されたツールの開発に関する論文である。従来の差分線形攻撃では、対象となるプリミティブEを差分攻撃パートE_uと線形攻撃パートE_ℓの 2 つのパートに分割し、これらのパートが完全に独立であるという仮定の下で計算量が評価されてきた。しかしながら、実際にはこれらのパートには依存関係があるため、従来の計算量評価が正確ではないという問題があった。この問題を解決するための代表的な手法として sandwich フレームワークがある。このフレームワークでは、対象となるプリミティブを 2 つのパートだけでなく、これらの依存関係を考慮するための中間パートE_mを導入する。つまり、プリミティブを$E = E_\ell \circ E_m \circ E_u$と分割して解析することにより、計算量をより正確に評価できるようになった。差分線形攻撃と類似した攻撃手法であるブーメラン攻撃では、この中間パートE_mを詳細に分析するためのブーメラン接続表（BCT: Boomerang Connectivity Table）フレームワークが提案され、このフレームワークを拡張するような様々な応用手法もまた提案されてきた。一方で、差分線形攻撃においても BCT フレームワークと同様に差分線形接続表（DLCT: Differential-Linear Connectivity Table）フレームワークが提案されているが、BCT フレームワークのような拡張手法が提案されていないという課題がある。特に、複数ラウンドをカバーする中間パートE_mを許容するような手法が未解決な課題として残されている。また、差分線形識別子を自動的に探索するツールが提案されているものの、ARX 暗号のための専用ツールであり、ブロックサイズが小さいバリアントにしか適用できないという汎用性の面で課題がある。

本研究ではこれらの課題を解決する。具体的には、BCT フレームワークの拡張手法

を参考に、複数ラウンドをカバーする中間パートE_mを許容するような新しい接続表を定義し、DLCT フレームワークを拡張した。また、この拡張した DLCT フレームワークを使用し、制約プログラミング（CP）と混合整数線形計画法（MILP）に基づく差分線形識別子探索のための汎用的な自動化ツールを開発した。提案ツールが様々なプリミティブに適用可能であることを示すために、AES、Ascon、CLEFIA を含む 11 種類のプリミティブに適用され、それぞれ既存の差分線形識別子を改善できることが示された。

Speeding up Preimage and Key-Recovery Attacks with Highly Biased Differential-Linear Approximations [Crypto 2024]

Zhongfeng Niu, Kai Hu, Siwei Sun, Zhiyu Zhang, Meiqin Wang

スポンジ構造型ハッシュ関数に対する原像攻撃とスポンジ構造型認証暗号に対する鍵回復攻撃の高速化手法に関する論文である。代表的なスポンジ構造型ハッシュ関数・認証暗号として Ascon と Sparkle がある。これらの暗号学的置換に対して差分線形（DL: Differential-linear）識別が最も有効な解析手法の 1 つであると知られているが、これらの識別攻撃が原像攻撃や鍵回復攻撃のようなより現実的な攻撃へと拡張された研究はない。特に、Sparkle のハッシュ関数/XOF バリアントである Esch/XOEsch に対する原像攻撃はまだ報告されておらず、Sparkle の認証暗号バリアントである Schwaemm に対する鍵回復攻撃は設計者らによる評価しか報告されていない。

本研究では、最初に原像攻撃を高速化させるための高バイアスな DL 識別子に基づく新しい解析フレームワークを提案する。基本的なアイデアは次のとおりである。関数Fの像（image）$\mathcal{O}$が与えられ、xと$x \oplus \delta$がそれに対応する原像かをチェックする。$y = F(x)$を計算した結果、$y = \mathcal{O}$の場合は原像xが得られるため攻撃成功となる。一方、$y \neq \mathcal{O}$の場合は DL 近似を応用して確率的に$F(x \oplus \delta)$の計算をスキップさせる。このスキップできる計算量だけ高速化が可能となり、これには複数個の高バイアスな DL 識別子が必要となる。同様のアイデアを使用し、鍵回復攻撃を高速化させるための解析フレームワークも提案された。後者のフレームワークでは、高バイアスな関連鍵差分線形識別子を使用して単一鍵設定での鍵回復攻撃を実現している。つまり、単一鍵設定での安全性を主張するためには、関連鍵設定での解析も考慮する必要があることを意味する。

これらのフレームワークはスポンジ構造型ハッシュ関数（Ascon-Hash）と XOF（XOEsch、Ascon-XOF）に対する原像攻撃とスポンジ構造型認証暗号（Schwaemm）に対する鍵回復攻撃に適用された。Ascon-Hash に対しては 12 ラウンド中 4 ラウンドまで原像攻撃に成功するが、これは設計者の主張する安全性レベルを超えた結果であることに注意が必要である。XOEsch に対しては 7 ラウンド中 2.5 ラウンドまで、Ascon-XOF に対しては 12 ラウンド中 4 ラウンドまで原像攻撃に成功した。また、Schwaemm に対しては 7 ラウンド中 4.5 ラウンドまで鍵回復攻撃に成功した。

New Approaches for Estimating the Bias of Differential-Linear Distinguishers [Crypto 2024]

Ting Peng, Wentao Zhang, Jingsui Weng, Tianyou Ding

共通鍵暗号プリミティブの差分線形（DL: Differential-Linear）バイアスを厳密に評価するための汎用的な手法に関する論文である。前述のとおり、DL 解析では対象となるプリミティブEを$E = E_\ell \circ E_m \circ E_u$と分割して解析する手法である。ここで、$E_u$は差分パート、$E_m$は中間パート、$E_\ell$は線形パートである。2017 年に Blondeau らは「DL 解析は、理論的な意味において、多次元線形攻撃もしくは切り詰め差分攻撃のいずれかとして考えることができる」と主張し、これらの変換方法について議論した。しかし、このアイデアはその後の進展がなく、理論的な意味での主張に留まっている。

本研究では、DL バイアスと切り詰め差分確率における関連性を形式的に表現する新しい公式を提示し、Blondeau らの理論を拡張させる。この拡張のために、切り詰め差分分布表（TDT: Truncated Differential Distribution Table）を提案する。TDT は単一の S-box に関する切り詰め差分分布、もしくは複数ラウンドの切り詰め差分分布を表現するものである。TDT を利用することで、切り詰め差分確率の評価を大幅に加速化することが可能となる。新しい公式と TDT に基づき、DL バイアスを厳密に見積もるための 2 つの新しいアプローチを提案する。1 つは中間パートE_mが複数ラウンドである際の DL バイアスを計算するためのものであり、もう 1 つは中間パートE_mが 1 ラウンドのみである際のバイアスを計算するためのものである。対象となるプリミティブの解析困難性に合わせてアプローチを使い分けることとなる。

これらのアプローチを 5 つの共通鍵暗号プリミティブ（Ascon、Serpent、KNOT、AES、CLEFIA）に適用した。Ascon に対しては、既存研究において 4/5 ラウンド DL 識別子の理論値と実験値に大きな乖離があったという問題に対処する。提案アプローチを適用することで理論値の改善に成功し、理論値と実験値がほぼ一致することを示した。また、新しい 6 ラウンド DL 識別子を発見し、既存の DL 識別子の中で最良の結果を達成した。AES に対しては、3 ラウンド DL 識別子をある条件の下で網羅的に探索し、その結果をベースとして 4/5 ラウンド DL 識別子へと拡張した。AES の 5 ラウンド DL 識別子は初めての結果である。その他、Serpent、KNOT、CLEFIA に対しても同様の結果が得られた。

2.2.2. 公開鍵暗号に関する解読技術

An Improved Practical Key Mismatch Attack Against NTRU [PQCrypto 2024]

Zhen Liu, Vishakha, Jintai Ding, Chi Cheng, Yanbin Pan

NTRU に対する鍵不一致攻撃に関する論文である。NTRU の復号は、一変数多項式である秘密鍵fと暗号文cに対して積$a = c * f \bmod q$を計算することで行われる。ここで、NTRU

の公開パラメータの一部である$q \ll p$なる互いに素な正整数p, qと素数Nについて、剰余環を$\mathcal{R}_p \coloneqq \mathbb{F}_p[x]/(x^N - 1), \mathcal{R}_q \coloneqq \mathbb{F}_q[x]/(x^N - 1)$とした際、$a = c * f \bmod q$は剰余環$\mathcal{R}_q$で一致することを意味する。実際の復号は、$\mathcal{R}_p$において$f$の逆元となる多項式$f_p$を利用して、$a * f_p^{-1} \bmod p$で行われる。この復号が成功するためには$a$の係数が$[-q/2\,, q/2)$の範囲に収まる必要があるが、NTRU ではパラメータを調整することでこれを実現している。

ここで、正整数nを用いて操作された暗号文$c_i = c + n * p * x^i$を同じ鍵で復号すると、$c_i * f = a + n * p * x^i * f \bmod q$が成り立つが、もし$a + n * p * x^i * f$の係数が$[-q/2\,, q/2)$にある場合、同様の手法で平文が復号される。係数がこの範囲を超える（オーバーフロー）する位置から秘密鍵fの情報を復元するというアプローチが、Hoffstein ら(2000)が提案した鍵不一致攻撃の基本的な手法であるが、係数が$[-q/2\,, q/2)$の範囲からはみ出る箇所が 1 か所でしか起きないという強い仮定があった。

本論文では、その仮定を取り除いた攻撃が提案された。また、NTRU に対する鍵不一致攻撃には Qin ら(Asiacrypt 2021)によりクエリ回数の下限が知られているが、この攻撃のクエリ回数はほぼその下界を満たしている。実験では、NIST 耐量子標準化プロジェクトに提案された NTRUEncrypt と NTRU-HPS への攻撃を行い、実際に提案パラメータでの鍵回復を成功させている。

Adaptive Attacks Against FESTA Without Input Validation or Constant-Time Implementation [PQCrypto 2024]

Tomoki Moriya, Hiroshi Onuki, Maozhi Xu, Guoqing Zhou

同種写像ベースの公開鍵暗号である FESTA に対する適応的攻撃についての論文である。FESTA とは、Castryck-Decru による SIDH 攻撃（Eurocrypt 2023）において用いられた Kani の定理を活用した、トラップドア関数およびそれを用いた公開鍵暗号方式である。FESTA は CIST（Computational Isogeny with Scaled-Torsion）問題を安全性の根拠としている。

本論文では、秘密鍵を公開鍵及び入出力から割り出す攻撃を提案している。この攻撃において想定しているシナリオは下記のようなものである。攻撃者 Bob はまず、秘密鍵を持っている人物 Alice に誤った処理で作成した暗号文を提示する。この暗号文は正しい処理で作られたものと同じ型を持っている。Alice はこの暗号文を復号するが、ここで仮に「復号処理が実行できたかできなったか」の情報が Bob に渡った場合、Bob は Alice の秘密鍵を割り出すことができることが本論文で報告された。

FESTA の暗号化処理において、Bob の入力の一部である2×2変換行列Bは、公開パラメータとして設定されている可換部分群$\mathcal{M}_b \subset \mathrm{GL}_2(\mathbb{Z}/2^b\mathbb{Z})$に属していることが想定されている。しかしながら著者らは、Bが$\mathcal{M}_b$に属していない場合に復号プロトコルがどのように振舞うかを観察することで、上述の適応的攻撃を導いた。

したがって、変換行列Bが指定された可換部分群$\mathcal{M}_b$に属するかをチェックする工程はス

キップできないと指摘された。さらにこの攻撃シナリオの場合、Alice は復号の過程において、「復号に失敗する」という例外 1 と、「復号に成功したがBが$\mathcal{M}_b$に属さなかった」という例外 2 のどちらかを得る。この例外出力結果はもちろんのこと、この例外が出力される時間が異なる場合、サイドチャネル攻撃が可能であることが指摘された。以上の観点から、FESTA は注意深く実装されるべきであると報告された。

Solving the Tensor Isomorphism Problem for special orbits with low rank points: Cryptanalysis and repair of an Asiacrypt 2023 commitment scheme [Crypto 2024]

Valerie Gilchrist, Laurane Marco, Christophe Petit, Gang Tang

テンソル同型問題（Tensor Isomorphism Problem、以下 TIP）と、D'Alconzo ら（Asiacrypt 2023）によるコミットメントスキームの解析に関する論文である。TIP は 3-テンソル（すなわち立方行列）間の同型写像を求める計算問題であり、NIST PQC 標準化プロジェクトの追加署名候補 MEDS の安全性の根拠となっている。

本論文では、特別な 3-テンソルに対する TIP を解く多項式時間アルゴリズムが提案された。この問題は D'Alconzo らによるコミットメントスキームの安全性の根拠となっていたので、本論文のタイトルにあるコミットメントスキームが安全でないことが報告された。

TIP は 3 つの一般線型群の直積からの群作用に関する問題として記述される。著者らは、当該コミットメントスキームで用いられる 3-テンソルの軌跡が低いランクのテンソルを持つこと、さらにその点については非自明な固定化部分群を導けることが、本攻撃において本質的な観察であると報告した。著者らは方式の修正として、ランダムテンソルを用いた新たな方式を提案している。

2.2.3. 電子署名に関する解読技術

Breaking Parallel ROS: Implication for Isogeny and Lattice-Based Blind Signatures [PKC2024]

Shuichi Katsumata, Yi-Fu Lai, Michael Reichle

ブラインド署名の安全性に関する論文である。ブラインド署名はその匿名性から、署名の偽造に関する安全性をどう定義するのかに関して議論があった。本論文では、ℓ-同時偽造不可能性（ℓ-concurrent unforgeability）という安全性が導入された（ここでℓは 2 以上の自然数）。これはℓ回の署名セッションが終了した後に攻撃者が$\ell + 1$個以上の正当な署名を生成できないという安全性の定式化であり、既存の署名方式がそれを満たすかどうかを著者らは議論した。

この種の安全性の議論は、Schnorr（ICICS 2001）により導入されたROS_ℓ問題（Random inhomogeneities in an Overdetermined Solvable system of linear equations）と、ROS_ℓ問題が解ければ Schnorr 署名のℓ-concurrent unforgeability が破られるという事実に端を発

する。セキュリティパラメータλに対して、$\ell = \mathrm{polylog}(\lambda)$では$ROS_\ell$問題が解けることがWagner（CRYPTO 2002）により示されていたが、ℓ-同時偽造不可能性についてはそもそも$\ell = \mathrm{poly}(\lambda)$に対する多項式時間攻撃が Benhamouda ら（Eurocrypt2021)で知られていた。本論文では、元々の問題の変種として parallel $ROS_{\ell,\omega,C}$問題とその多項式時間攻撃を提案し、近年提案された parallel repetition 型のブラインド署名に対しても同様の攻撃が可能であることが示された。具体的同種写像ベースの CSI-Otter、格子ベースの Blaze+、BlindOR がターゲットとなっており、それぞれに対応する parallel $ROS_{\ell,\omega,C}$問題の具体的な形と解析が与えられている。

The Blockwise Rank Syndrome Learning Problem and Its Applications to Cryptography [PQCrypto 2024]

Nicolas Aragon, Pierre Briaud, Victor Dyseryn, Philippe Gaborit, Adrien Vinçotte

耐量子計算機署名形式の候補である、ROLLO および RQC に関する攻撃論文である。これらは Rank-based cryptography とよばれる符号ベース暗号の一種に属し、特定の階数（rank）を持つ符号語を発見する問題の困難性を安全性の根拠とする。より詳細には、ROLLO は LRPC 方式に、RQC は RQC 方式に属する。これらの方式は、Rank Support Learning 問題に基礎を置く多重シンドローム（multi-syndrome）アプローチにより効率化されていたが、著者らは Asiacrypt 2023 において Song らにより導入されたブロックワイズエラー（blockwise error）アプローチにより、公開鍵および署名長の長さをさらに大きく削減できることを指摘したうえで、この二つのアプローチを組み合わせた新たなアプローチを提案した。これは著者らが導入した Blockwise Rank Syndrome Learning 問題に基礎を置く。この新たなアプローチにより一般的な RQC 方式および LRPC 方式における公開鍵と暗号文の長さを大きく改善した。

加えて本論文では、上記 Song らによって導入された l-RD 問題について新たな攻撃を導入し、ROLLO-I に対する攻撃を実施した。結果、128/192/256 ビットセキュリティの方式に対して、主張されていた安全性は 145/225/281 ビットセキュリティであったが、著者らの攻撃によりこれが 116/166/224 ビットセキュリティに低下した。

Cryptanalysis of the SNOVA Signature Scheme [PQCrypto 2024]

Peigen Li, Jintai Ding

耐量子計算機署名形式である SNOVA に対する攻撃論文である。SNOVA は多変数方程式の求解困難性に基づいた署名であり、NIST 耐量子計算機電子署名の第 2 ラウンド候補である。特に、構造付きの UOV 型の署名としてその署名サイズの小ささと処理性能が注目されているものの、行列を変数として並べた写像を用いる非可換環的な構成をしており、その安全性は詳しく解析されていなかった。SNOVA の開発者らによれば、パラメータ(v, o, q, l)の SNOVA は、パラメータ(l^2v, l^2o, q)を持つ$\mathbb{F}_q$上の UOV として捉えることが可能であり、そ

の性質を用いてパラメータが設定されていた。

本論文では、提案パラメータを持つ SNOVA の攻撃計算量は開発者が主張するよりも低いことが報告された。著者らは、SNOVA の構成を注意深く取り扱うことで、同じパラメータの SNOVA がパラメータ(lv, lo, q)を持つl^2o本の式を持つ UOV として捉えられることを明らかにし、攻撃の解析を行った。なお、この攻撃を適用した後でも攻撃計算量は引き続き指数的であり、パラメータの修正により安全性は確保されるとして NIST 追加署名の第 2 ラウンドに残っている。

また同様の手法により署名方式 NOVA の攻撃計算量も低くなることが判明している。

One Vector to Rule Them All: Key Recovery from One Vector in UOV Schemes [PQCrypto 2024]

Pierre Pébereau

耐量子計算機署名形式である MAYO を含む、UOV 型の署名方式に対する攻撃論文である。UOV 型の署名方式では、署名者が線形の連立方程式を解くことで署名可能であるのに対して、署名偽造を行う攻撃者は計算量的に困難な 2 次の連立方程式を解く必要があるというトラップドア構造を用いている。この困難性の差を生むのが”oil subspace”であり、秘密鍵に対応する。

本論文では、oil subspace の中の 1 本のベクトルと公開鍵の情報から、秘密鍵を多項式時間で復元するアルゴリズム、およびあるベクトルが oil subspace に含まれるかどうかを判定する多項式時間アルゴリズムが与えられた。また、同様の結果が UOV-like な署名方式である MAYO や VOX に対しても可能であるという結果を著者らは得ている。これらの結果は、UOV 型の署名の安全性が oil subspace の中の 1 本のベクトルを発見することの困難性に依存していることを示しており、サイドチャネル攻撃への対策を強化する必要があることが著者らにより報告された。

検証用コードは https://github.com/pi-r2/OneVector で公開されており、UOV の代表的なパラメータに対して数秒から 2.5 分で鍵が復元可能であるとしている。

Practical Key-Recovery Attack on MQ-Sign and More [PQCrypto 2024]

Thomas Aulbach, Simona Samardjiska, Monika Trimoska

署名方式 MQ-Sign に対する攻撃についての論文である。MQ-Sign は UOV 型の署名方式で、Shim らにより KpqC に提出されたアルゴリズムであり、鍵長を削減するために中心写像にスパースな構造を入れている。具体的には、中心写像を$F = F_V + F_{OV} + F_{LC}$と vinegar-vinegar 変数の項、oil-vinegar 変数の項、それ以外の線形・定数項に分割して表現したときに、F_V, F_{OV}をランダム多項式に基づくものとするか、スパースなものとするかにより 4 種の選択があり、F_Vに関する選択を 1 文字目の S/R で、F_{OV}に関する選択を 2 文字目の S/R で表現し、それぞれ MQ-Sign-SS,-RS,-SR,-RR と名付けている。

本論文では、F_{OV}がスパース構造を持つ MQ-Sign-SS および MQ-Sign-RS に対する鍵復元攻撃を EIP 問題として定式化し、そのスパース性から行列のランクが著しく落ちる事を利用して多項式時間での鍵復元攻撃を行っている。

さらに本論文では、MQ-Sign-SR に対する高速な偽造攻撃を提案している。こちらはF_Vのスパース性を利用することで変数の数を削減しており、グレブナー基底計算を用いた解析によればレベル I,III,V のパラメータの攻撃計算量はそれぞれ$2^{111}, 2^{170}, 2^{228}$となり、128,192,256 ビットセキュリティを持たないことが確認された。

多項式時間攻撃のスクリプトは https://github.com/mtrimoska/MQ-Sign-attack で公開されており、最も高いセキュリティレベル V のものでも 7 秒で鍵復元が可能であるとしている。

Practical and Theoretical Cryptanalysis of VOX [PQCrypto 2024]

Hao Guo, Yi Jin, Yuansheng Pan, Xiaoou He, Boru Gong, Jintai Ding

耐量子計算機署名形式 VOX についての攻撃論文である。VOX は UOV 系の署名スキームであり、NIST PQC 標準化プロジェクトの追加署名に提出されている。2023 年に Furue,Ikematsu による rectangular MinRank 攻撃が発見され、その後開発者によるパラメータのアップデートがあった。

本論文では、更新されたパラメータをもつ方式に対する MinRank 攻撃再評価を行い、現実的な時間内で鍵復元が可能であることが示された。VOX の特徴は中心写像 F の構成を少量のランダムな二次多項式と剰余環の構造を持つ多項式を混ぜて構成することで鍵サイズの削減と効率化を図っており、その特徴が MinRank 攻撃により効率的に抽出可能であることが改めて確認された形となる。

攻撃スクリプトは https://github.com/tuovsig/analysis で公開されており、古典的な Kipnis-Shamir 攻撃とグレブナー基底計算を組み合わせたものでも 2 秒程度で鍵復元が可能であるとしている。理論的な計算量評価も行っており、開発者らの想定とは逆に、レベル 1,3,5 のパラメータがそれぞれ 112,69,48 ビットセキュリティであると計算している。

FuLeakage: Breaking FuLeeca by Learning Attacks [Crypto 2024]

Felicitas Hörmann, Wessel P. J. van Woerden

耐量子計算機署名方式である FuLeeca についての解析論文である。FuLeeca は NIST PQC 標準化プロジェクトに提出された Hash-and-Sign フレームワークの符号ベース署名である。構成が格子署名 FALCON に似ていることから、FALCON の変種の解析で用いられた複数署名からの秘密鍵復元手法の適用可能性が本論文で検証された。

本論文では、FuLeeca に対する鍵回復攻撃が提案された。技術的には符号ベースの記述がなされている FuLeeca を格子ベースの記述に変換する事で FALCON を中心とする GPV フレームワークの格子署名に対する鍵復元アルゴリズムを適用している。攻撃者の持つ署

名数が少ない場合には鍵復元に用いられる格子の部分格子が抽出され、それに合わせて攻撃計算量が下がる。多い場合には学習アルゴリズムを用いて鍵復元を行う。顕著な結果として、最も高いセキュリティパラメータの場合でも 175,000 個の署名から 1 時間で復元可能な格子攻撃が実証されている。また、元々の方式で用いられていた準巡回符号の構造を用いることで、Cramer ら（Eurocrypt 2016）の量子多項式時間での short generator 復元攻撃が適用可能であるという 2023 年 7 月の pqc-forum での指摘が正しいことが確認された。

Rare structures in tensor graphs - Bermuda triangles for cryptosystems based on the Tensor Isomorphism problem [Asiacrypt 2024]

Lars Ran, Simona Samardjiska

NIST 耐量子計算機署名方式の候補である MEDS と ALTEQ の安全性に関連する、3-TI（3-Tensor Isomorphism）問題に関する解読論文である。3-TI 問題とは、有限体 $\mathbb{F}_q$ 上で与えられた、座標変換で移りあう二つの 3-テンソル（立方行列のこと）に対して、その座標変換を具体的に求める問題である。MEDS の安全性は、3-TI 問題と同値な問題である MCE（Matrix Code Equivalence、行列符号同値）問題に依拠している。ALTEQ の安全性は、3-TI 問題の特別な場合である ATFE（Alternting Trilinear Form Equivalence、交代三線型形式同値）問題に依拠している。先行研究として、MCE 問題に対しては Narayanan らによるグラフ理論的アルゴリズムが、ATFE 問題に対しては Ran らによる代数的アルゴリズムが知られており、MEDS と ALTEQ の安全性をわずかに低減させることが報告されていた。

本論文では、3 - TI 問題を解く新手法が報告された。著者らは、秘密鍵である座標変換を回復するために、"三角形"という新たな不変量を導入した。"三角形"とは、3-テンソルのグラフにおける長さ 3 の周期のことを指しており、確率約$1/q$ という非常に稀な確率で存在する。このアプローチはグラフベースのアルゴリズムで典型的だが、しかし通常の組合せ論的手法とは異なり、不変量を非線形方程式系としてモデル化しそれを解くという、純粋な代数的手法を用いて行われる。この非線形方程式系の求解には、tri-graded な多項式環に適応した Gröbner 基底法を用いている。

著者らは、この新たな不変量である"三角形"を探し出すアルゴリズムを提案し、その計算量の下限と上限を評価した。そして、"三角形"が存在する場合に機能する、MEDS と ALTEQ 双方への新たな暗号解析を与えている。特に（"三角形"が存在する）ALTEQ について、実用的な攻撃が提案された。この攻撃は全てのセキュリティレベルに対して少なくとも 60 ビットの改善を与え、特にレベル I のパラメータに対してわずか 1501 秒で秘密鍵を回復することが報告された。

Don't Use It Twice! Solving Relaxed Linear Equivalence Problems [Asiacrypt 2024]

Alessandro Budroni, Jesús-Javier Chi-Domínguez, Giuseppe D'Alconzo, Antonio J. Di Scala ,Mukul Kulkarni

NIST 耐量子計算機署名方式の候補である LESS 署名方式、MEDS 署名方式の安全性に関連する、LCE（Linear Code Equivalence）問題に関する解読論文である。

LCE 問題とは、二つの(n,k)-線型符号に対して与えられる生成行列G, G'について、$G' = SGQ$となるk次正則行列Sとn次単項行列Qを計算せよ、という問題であり、LESS 署名方式の安全性の根拠となっている。線型符号が行列の為す線形空間の部分空間である場合、LCE 問題は MCE（Matrix Code Equivalence）問題と等価である。この意味で本論文は、MCE 問題を安全性の根拠とする MEDS 署名方式とも関連している。

本論文は、この LCE 問題と MCE 問題に関して、秘密を回復するために必要なサンプル数について研究している。先行研究として、$G_i' = SG_iQ$なる対(G_i, G_i')が$i = 1, \dots, t$について与えられているとき、サンプル数tが$t = kn$であれば、(S, Q)が復元できることが知られていた。このサンプル数tを減らすのが本論文の目標である。加えて本論文では、組織形式 LCE 問題（LCE Systematic Form Version）も視野に入れることで、$G_i' = S_iG_iQ$のようなケースにQを復元する方式についても提案している。ここで組織形式 LCE 問題とは、G, G'が組織符号の生成行列であるとき、G'がGQの組織符号となる単項行列Qを計算する問題である。

結果として、サンプル数tが$t = \left\lfloor \frac{n^2}{k(n-k)} \right\rfloor + 1$、さらに$k = n/2$であるときは$t = 2$であれば、多項式時間計算量および多項式メモリ計算量で LCE 問題を解くアルゴリズムが提案された。著者らはサンプルG_i'のパリティ検査行列をH_i'としたとき、$(G_i \otimes H_i') \cdot vec(Q) = 0$が満たされることに着目した。ここで$vec(Q)$は$Q$の行ごとに成分を並べた列ベクトルである。サンプル数が$t = 2$の場合、著者らは Saeed のアイデアに着想を得て上記の線型方程式を解くことで、秘密鍵を導いている。結果として、LCE 問題に依拠する LESS 署名形式および MEDS 署名形式において、秘密鍵の再使用は避けるべきと結論された。

2.2.4. サイドチャネル攻撃に関連する解読技術

Fault Attack on SQIsign [PQCrypto 2024]

JeongHwan Lee, Donghoe Heo, Hyeonhak Kim, GyuSang Kim, Suhri Kim, Heeseok Kim, Seokhie Hong

耐量子計算機署名形式である SQISign に対するサイドチャネル攻撃である。SQISign は同種写像ベースのシグマプロトコルを Fiat-Shamir 変換により署名方式としたものである。

本論文では、署名生成関数の中のコミットメントフェーズにおいて、イデアルI_{com}と同種

写像ϕ_{com}の計算ルーチンに対する故障注入攻撃が提案された。特に、計算ルーチンの実行を制御することで平文に対する偽の署名を生成する手法と、署名の情報から分割統治法による秘密鍵の復元攻撃手法が提案された。I_{com}の計算ルーチンへの攻撃はリファレンス実装における copy 関数への故障注入であり、これにより結果がランダムな値となる。対策として、イデアルのノルムをチェックする機構を組み込むことが有効であるとしている。ϕ_{com}の計算ルーチンへの攻撃は for ループの実行を途中で打ち切るものであり、対策としてループ回数のカウンタチェックが有効であるとしている。

さらにこれらの脆弱性を利用して、2 種類の攻撃シナリオの下で具体的攻撃が検討された。deterministic SQISign に対しては、鍵を固定した署名生成オラクルへのアクセスが可能であり、署名生成の実行中に 1 回だけ故障注入が可能なシナリオが想定され、randomized SQISign は対しては、署名生成オラクルが用いる鍵が毎回異なるものであっても良いが、攻撃者が 2 回故障注入できるシナリオが想定されている。

Attacking ECDSA with Nonce Leakage by Lattice Sieving: Bridging the Gap with Fourier Analysis-based Attacks [Asiacrypt 2024]

Yiming Gao, Jinghui Wang, Honggang Hu, Binang He

ECDSA および Diffie-Hellman 鍵共有に対するサイドチャネル攻撃に関する論文である。特に本論文では、サイドチャネル攻撃のアプローチとして、HNP（Hidden Number Problem）について考察する。HNP を解くための 2 つの主要なアプローチとして、格子ベース攻撃とフーリエ解析ベース攻撃が知られている。十分に長いナンス（Nonce）部分列が漏洩している場合、格子ベース攻撃はより効率的であり、少ないサンプル数で攻撃可能である。しかし、例えばナンスのわずかな部分（例えば 1 ビット）のみが漏洩している場合には、従来手法による攻撃は困難であった。またフーリエ変換ベースの攻撃手法と比較して、入力の誤りについてのロバスト性が失われていた。

本論文では、これら 2 つのアプローチ間のアルゴリズム的なトレードオフを導入している。Albrecht と Heninger が EUROCRYPT2021 で提案した格子を修正するために、著者らはパラメータxを導入した新たな格子を提案した。漏洩したビット数をlとして、簡約された格子次元は約$\log_2 x / l$に比例して減少する。著者らはさらに、誤り入力を伴う HNP も著者らの格子ベース手法で解けるよう、アルゴリズムを拡張した。結果として、ECDSA に対する既存の最先端の格子ベースの攻撃が改良された。特に 1 ビット漏洩に対する以前の最良の格子ベース攻撃は 112 ビット以下の曲線上でのみ行われていたが、これを 160 ビット以下の曲線に改良した。

It's a Kind of Magic: A Novel Conditional GAN Framework for Efficient Profiling Side-channel Analysis [Asiacrypt 2024]

Sengim Karayalcin, Marina Krcek, Lichao Wu, Stjepan Picek, Guilherme Perin

AES-128 に対するサイドチャネル攻撃についての論文である。プロファイル型サイドチャネル攻撃では、攻撃対象のデバイスの完全なコピーであるプロファイルデバイス（profiling device）に、マスク（乱数）も込めてアクセスできることを仮定する。しかし著者らは、安全な製品のデバッグモードやテストモードには強力な保護を適用するため、現実的にプロファイルデバイスを取得することは困難であると指摘する。

本研究では、攻撃者がプロファイルデバイスにはアクセスできないが、全く異なるデバイスにアクセスできる場合の鍵回復攻撃を検討している。具体的な手順は以下のようである。攻撃者はまず、攻撃するデバイスとは全く別のデバイス（参照デバイスと呼ぶ）に攻撃アルゴリズムを実装する。参照デバイスは実際に攻撃するデバイスとは異なるものの、攻撃者にとってホワイトボックスであり、すべての内部状態とマスクにアクセスできると仮定される。次に、実際に攻撃するデバイスのデータを用いた、新しい CGAN（Conditional Generative Adversarial Network、条件付き敵対的生成ネットワーク）フレームワークを導入する。この CGAN フレームワークにおいて、生成器は、ホワイトボックス設定で得られる特徴量を模倣する。この模倣された内部的特徴量を入力としてサイドチャネル攻撃を行うことが、著者らの提案した手法である。

著者らは実験を行い、本手法がブラックボックスなサイドチャネル攻撃の有効性を向上させ、最悪ケースにおける安全性評価の結果と一致するか、または潜在的に上回るケースがあることを報告した。以上の結論を下に、著者らのフレームワークはプロファイルデバイスにアクセスする必要はなく、従ってより現実的な攻撃であると報告された。

ZKFault: Fault attack analysis on zero-knowledge-based post-quantum digital signature schemes [Asiacrypt 2024]

Puja Mondal, Supriya Adhikary, Suparna Kundu, Angshuman Karmakar

NIST 耐量子計算機署名方式の候補である LESS 署名方式、CROSS 署名方式、MEDS 署名形式についての故障注入攻撃についての論文である。これらの符号ベース署名方式は、秘密鍵の知識についての対話型ゼロ知識署名を Fiat-Shamir 変換することで、署名方式を得ている。LESS、CROSS、MEDS では、署名長を圧縮するために、参照木（Referece Tree）と著者らが呼んでいる木構造を利用している。例えば LESS 署名方式では、署名者と検証者の間で単項行列生成器を共有し、Goldreich-Goldwasser-Micali 構成を活用することで、チャレンジに対するレスポンス（すなわち署名）を、シード値と参照木のノードを指定することで圧縮する。

本論文では、この参照木に対する故障注入攻撃を研究している。著者らは、この参照木にビットフリップを発生させることで、秘密鍵が漏洩することを指摘した。この故障

注入攻撃は、参照木を用いて署名長を圧縮している上述3ケースに通用する。特にLESS方式及びCROSS方式について参照木の構造が研究され、全ての秘密鍵が漏洩するような単一故障注入攻撃が提案された。また著者らは、こういった故障注入攻撃を防ぐ方法と、その方法が署名長にどのような影響を与えるかを議論している。

2.2.5. 量子計算機による解読技術

A Subexponential Quantum Algorithm for the Semidirect Discrete Logarithm Problem [PQCrypto 2024]

Christopher Battarbee, Delaram Kahrobaei, Ludovic Perret, Siamak F. Shahandashti

半直積離散対数問題（Semidirect Discrete Logarithm Problem、以下SDLP）に関する量子解読論文である。SDLPとは、有限群（もしくは有限半群）Gの元gと、自己準同型$\phi: G \to G$が与えられたとき、$s_{g,\phi}(x) \coloneqq \phi^{x-1}(g)\phi^{x-2}(g)\cdots\phi(g)g$から$x$を復元する計算問題である。これは通常の有限アーベル群に対する離散対数問題を半直積$G \rtimes \mathrm{End}(G)$へと一般化したものであり、耐量子性を持つ暗号プリミティブとして期待されている。

本論文では、半群の加算族$\{G_p\}_p$が”簡単（easy）”であるという技術的な仮定の下、準指数時間および準指数量子クエリ数でSDLPを解くアルゴリズムが報告された。ここで$\{G_p\}_p$が簡単であるとは、群の位数がpの多項式オーダーかつ、群演算と自己準同型写像の評価が$O((\log p)^2)$時間で完了するものとして定義される。

この解読に向けたアプローチとして、著者らはSDLPの群作用離散対数問題（Group Action Discrete Logarithm Problem、以下GADLP）への量子帰着を研究した。本論文で使用されるGADLPインスタンスは巡回群作用についてのものである。著者らはまず、関数$s_{g,\phi}: \mathbb{Z}_{\geq 0} \to G$を観察し、有限個の正整数$n$を除き$s_{g,\phi}(n) = s_{g,\phi}(n+r)$が満たされる数$r$のことを周期（period）と呼び、$s_{g,\phi}(n) = s_{g,\phi}(n+r)$が満たされる最小の$n$を指数（index）と呼んだ。このように関数$s_{g,\phi}$は$G$における長さ$r$のサイクルを与えるが、このサイクルに自由かつ推移的な巡回群作用を明示的に与えることで、指数nの計算とGADLPに問題を帰着した。より具体的には、簡単な半群の加算族$\{G_p\}_p$に対して、一回のGADLPクエリで機能する、時間計算量$O((\log p)^4)$、正解確率$\Omega(1)$でSDLPを解くアルゴリズムを与えた。これにより簡単な半群の加算族に対するSDLPは、指数nの計算と、GADLPの量子アルゴリズムを与えることに帰着される。前者に関しては具体的な量子アルゴリズムを、後者についてはKuperberg もしくは Regev により与えられた GADLP のアーベル群隠れシフト問題（abelian hidden shift problem）への還元を援用することで、上述の結果を導出している。

Extending Regev's Factoring Algorithm to Compute Discrete Logarithms [PQCrypto 2024]

Martin Ekerå, Joel Gärtner

離散対数問題に対する量子解析論文である。本研究は、Regev[arXiv: 2308.06572]により与えられた素因数分解の量子アルゴリズムに関連がある。著者らは、Regev のアルゴリズムが Shor のアルゴリズムの多次元化であり、Shor のアルゴリズムが素因数分解問題と離散対数問題をどちらも解くことを観察したうえで、Regev のアルゴリズムを離散対数問題を解くアルゴリズムへと自然に拡張した。

Regev の素因数分解アルゴリズムでは、適当な$(a_1, \dots, a_d)$を用いて$\prod a_i^{z_i} \bmod N$の周期を発見するために、格子基底の中から良いベクトルを探し出す。本研究では同様に、インスタンス$(g, x = g^e)$からの離散対数の計算のため適当な群の要素$(g_1, \dots, g_{d-1})$を基底として、$x^{z_d}\prod g_i^{z_i} = 1$を満たす点$(z_1, \dots, z_d)$がなす格子の中から良いベクトルを選びだす。

結論として、巡回群$\mathbb{F}_p^*$上の離散対数問題を Shor のアルゴリズムよりも少ないリソースで解く量子アルゴリズムが構成された。しかし応用上価値の高い楕円曲線上の離散対数問題を解く量子アルゴリズムの構成は未解決として残されている。

Space-Efficient and Noise-Robust Quantum Factoring [Crypto 2024]

Seyoon Ragavan, Vinod Vaikuntanathan

素因数分解に対するの量子解析論文である。本論文は、2023 年に Regev(arXiv: 2308.06572)により発表された量子素因数分解アルゴリズムを、量子メモリ効率およびノイズ耐性の二点で改善した。

第一の結果として著者らは、素因数分解問題を解く$O(nlog\, n)$量子ビット、$O(n^{3/2}log\, n)$量子ゲートの量子回路を提案した。Regev のアルゴリズムは$O(n^{3/2})$量子ビット、$O(n^{3/2}log\, n)$量子ゲートであるため、量子ビット数が削減されている。Regev のアルゴリズムは Shor のアルゴリズム（$O(n)$量子ビット、$O(n^2log\, n)$量子ゲート）の一般化として得られた経緯を加味すると、著者らの結果は量子ビット・量子ゲートのトレードオフを改良したと言える。この結果を得るために著者らは、Kaliski(arXiv: 1711.02491)のフィボナッチ数列を用いた指数計算回路を適用し、in-place での剰余乗算を可能とした。さらにいくつかの量子ビット数・量子ゲート数のトレードオフも提案しており、乗算の方法によっては$(10.32 + o(1))n$量子ビット、$O(n^{5/2}\log n)$ゲート数のものも構成可能であると報告された。

第二の結果として、著者らは後処理アルゴリズムを改良した。元々Shor のアルゴリズムはノイズに弱いことが指摘されており、Regev アルゴリズムもその性質を引き継いでいた。この論文では、回路実行中にエラーがあった場合、その出力がランダムな d 次元の点となると仮定することでフィルタリング手法を提案し、良いサンプルのみを用いて後処理を行う事で成功率を上げている。結果として回路中の 1 論理ゲート当たりのノイズが$\tilde{O}(1/n^{1.5})$であれば十分な確率で素因数分解が可能であると報告された。

Quantum Complexity for Discrete Logarithms and Related Problems [Crypto 2024]

Minki Hhan, Takashi Yamakawa, Aaram Yun

離散対数問題に対する量子解析に関する論文である。本論文では generic group model、つまり群の構造やエンコーディングのような付加情報を用いずに、量子アルゴリズムおよび量子・古典ハイブリッドアルゴリズムの計算量について研究が行われている。

群Gの位数を$|G|$とするとき、Shor 型のアルゴリズムで離散対数問題を解く量子計算量は$O(\log|G|)$と記述できる。本論文では、離散対数問題を解く量子回路の深さは少なくとも$\Omega(\log|G|)$であることが示された。これは Shor のアルゴリズムが漸近的に最適な量子アルゴリズムであることを意味する。また、離散対数問題を解く量子・古典のハイブリッドアルゴリズムで（量子・古典に関わらず）Q回の群要素操作を行うアルゴリズムは、量子回路深さ$\Omega(\log\log|G| - \log\log Q)$の群操作を$\Omega(\log|G| / \log Q)$回必要とすることが示された。さらに量子回路以外の要素として、t個の群要素を保存する量子メモリとr個の群要素を保存する quantum random access classical memory(QRACM)を利用可能な場合、量子・古典ハイブリッドアルゴリズムは$\Omega(\sqrt{|G|})$回の群操作クエリかもしくは$\Omega(\log|G| / \log(tr))$回のクエリを必要とする。後者の下界は古典クエリに対する下界でもある。ここで、QRACM は古典データx_iを保存しつつも、量子的なアクセスにより操作$|i\rangle \otimes |0\rangle \to |i\rangle \otimes |x_i\rangle$を可能なデバイスとされる。

本論文ではこれらの結果を応用し、multiple DLP(クエリ…, g^{x_m}から$x_1, \ldots, x_m$を計算する問題)を個別に解くよりも同時に解く方が効率的となるアルゴリズムを示し、パスワード認証鍵共有(PAKE)の文脈で提唱されている quantum annoying property(Eaton, PQCrypto2021)が strong form では成り立たないことを報告した。この性質は、もしも方式が破られるならば PAKE のパスワードの推測ごとに 1 回の離散対数問題を解かなければならないとされるもので、方式の安全性を議論するために導入された。

Reducing the Number of Qubits in Quantum Information Set Decoding [Asiacrypt 2024]

Clémence Chevignard, Pierre-Alain Fouque, André Schrottenloher

シンドローム復号問題に関する量子解析論文である。本論文では、Bernstein が PQCrypto 2010 で提案したシンドローム復号問題の量子求解アルゴリズムである ISD（Information Set Decoding）の量子メモリコストの最適化を取り組んでいる。

Bernstein の ISD は、Prange の ISD に Grover の量子探索を組み合わせることで得られており、符号の長さがnであるとき、各反復処理で$O(n)$次元の線型系を解くことができる。耐量子計算機暗号では、nとして10^3から10^5の値が採用されている。このアルゴリズムにおいては、ガウスの消去法が、行列を記憶するのに$O(n^2)$のメモリを必要とするため、量子メモリ計算量のボトルネックとなっていた。

著者らはこの量子メモリを削減するため、Wiedemann によるスパース行列の逆行列アルゴリズムを活用した。このアルゴリズムでは、ベクトルへの行列作用のみを考慮すれば良いため、メモリの削減が実現される。結果として、$O(n)$量子メモリ、$O(n^3)$個の Toffoli ゲートを用いた深さ$O(n^2 \log n)$の ISD 量子回路が提案された。例えば最小の Classic McEliece については、以前の研究では 50 万量子メモリが必要だったが、著者らのアルゴリズムはこれを 18098 量子メモリまで削減した。

これらの方式はメモリについては最適化されているが、Toffoli ゲート数についてはそうではない。このトレードオフとして、$O(n \log^2 n)$量子ビットの下で、Toffoli ゲートの数を$O(n^2 \log^2 n)$に削減した方式も本論文で提案された。

Quantum Circuits of AES with a Low-depth Linear Layer and a New Structure [Asiacrypt 2024]

Haotian Shi, Xiutao Feng

AES-128 についての量子解析に関する論文である。量子解読アルゴリズムの中には、AES-128 の量子回路実装を必要とするものもある。この攻撃のコストを適切に評価するために、AES-128 の最適な量子実装が求められている。

本論文では、AES-128 の新たな量子回路が提案された。著者らは、Rijndael (AES) MixColumns 行列を表す CNOT 回路を、広さ（width）32 深さ（depth）16 から、広さ 32 深さ 10 に削減した。さらにパイプライン構造を圧縮した。パイプライン構造とは、Grover と Simon のアルゴリズムに基づく量子攻撃で反復して用いられる量子オラクルの実装であるが、従来のパイプライン構造はラウンド数に応じて width が大きくなってしまう。本論文では、ラウンド数に応じて width が増加しない圧縮パイプライン構造を提案した。これはアンシラ量子ビットが少ない場合に効果的である。これにより、AES-128 の量子回路の T-深さ（行あたりの T ゲートの数）を 60 から 33 に削減した。さらに入力レジスタを変更しない Sbox 回路を提案することで、鍵スケジュールにおいて鍵を保存するために割り当てられていたアンシラ量子ビットを削減した。

結果として、AES-128 の量子回路実装として、広さと Toffoli 深さ（行あたりの Toffoli ゲートの数）の積 TofD-W のコストが 130720 にまで削減された。これは現状知られている中で最良である。

2.2.6. 暗号プリミティブに関する解読技術

A Simpler and More Efficient Reduction of DLog to CDH for Abelian Group Actions [PKC 2024]

Steven D. Galbraith, Yi-Fu Lai, Hart Montgomery

可換群Gが集合Xに作用している場合に一般化された暗号プリミティブに関する論文

である。特に本論文では、効率的群作用(Efficient Group Action: EGA)と呼ばれる、任意の$g \in G, x \in X$に対して$gx \in X$が効率的に計算可能であるという仮定の下、離散対数問題および計算量的 Diffie-Hellman 問題の類似物である以下の二つの問題(Galbraith et al., Math Crypto. 2021)が取り組まれた：

GA-DLog: (x, gx)からgを計算せよ。

GA-CDH: (x, ax, bx)から$(ab)x$を計算せよ。

これらの計算問題の困難性について、上記 Galbraith らの研究では GA-Dlog から GA-CDH への量子帰着が、GA-CDH オラクルが perfect correctness を持つという条件の下で可能であることが示された。その後、Montgomery と Zhandry(J. Crypto, 2024)により、信頼性の低い GA-CDH オラクルを信頼性の高いオラクルに変換する手法が提案され、上記帰着問題は一定の解決を見たものの、帰着効率はオラクルの信頼率εに対して$1/\varepsilon^{21}$という非効率なものであった。

本論文で著者らは、Montgomery と Zhandry の帰着手法中で用いられる確率の評価アルゴリズムを単純化し、最終的な効率を$1/\varepsilon^{4}$まで向上させた。

An Algorithm for Efficient Detection of (N, N)-Splittings and Its Application to the Isogeny Problem in Dimension 2 [PKC2024]

Maria Corte-Real Santos, Craig Costello, Sam Frengley

2 次元超特殊（superspecial）同種写像問題に関する論文である。これは、正標数pの基礎体K上の種数 2 の超特殊曲線C, C'について、超特殊アーベル曲面であるヤコビ多様体間の同種ϕ: $\mathrm{Jac}(C) \to \mathrm{Jac}(C')$を計算する問題であり、種数 2 の同種写像ベース 2022 年の SIKE 攻撃に関連するトピックといえる。

2 次元超特殊同種写像問題を解く既知のアルゴリズムで最も良いものは、Costello-Smith(PQCrypto 2020)による 2 ステップアルゴリズムである。第 1 ステップでは$\mathrm{Jac}(C)$,$\mathrm{Jac}(C')$のそれぞれから（疑似）ランダムウォークを行い、超特異楕円曲線の積で表現される$E_1 \times E_2$と$E'_1 \times E'_2$へのパスを発見する。第 2 ステップでは、E_1とE'_1、E_2とE'_2間のパスを Delfs-Galbraith アルゴリズム(Des. Codes. Crypyo. 2016)により発見して合成することで、目的の同種写像を得る。このアルゴリズムは同種写像ベースの暗号解析に用いられたことから、この分野の基本的なアルゴリズムとして知られている。

一方、種数 2 の曲線Cのヤコビ多様体$\mathrm{Jac}(C)$が分裂している(split)とは、二つの楕円曲線E_1, E_2の直積の上への分離的K-同種写像 ϕ:$\mathrm{Jac}(C) \to E_1 \times E_2$が存在することをいう。この同種$\phi$をより詳細に研究するため、著者らは Bruin-Doerksen に従い、最適(N, N)-分裂（optimal (N, N)-splitting）という性質を研究した。ここで上記の同種ϕが最適(N, N)-分裂であるとは、ϕの次数が平方数N^2（ここでNはpと互いに素）であり、ϕの核が$\mathrm{Jac}(C)$に定まるN-Weil ペアリングに関して極大等方部分群かつ$(\mathbb{Z}/N)^2$ と同型、すなわち(N, N)-同種写像であり、かつ、ϕに射影及びヤコビ埋め込みを合成することで得ら

れる被覆$C \to E_1$が非自明な不分岐被覆を経由しない、すなわち最適である、ことと定義される。

本論文では、与えられた種数 2 の超特殊曲線について、Jac(C)が最適(N,N)-分裂であるかどうかを判定する高速なアルゴリズムが与えられた。これを用いることで上記疑似ランダムウォークの速度が飛躍的に上がり、pが 100-1000 ビットの場合に 25-160 倍の高速化を達成している。

Improved Provable Reduction of NTRU and Hypercubic Lattices [PQCrypto 2024]

Henry Bambury, Phong Q. Nguyen

NTRU 格子および超立方格子（hypercubic lattice）に対するブロックワイズ簡約（blockwise reduction）攻撃に関する論文である。NTRU 格子は NTRU および FALCON で、超立方格子は HAWK で、それぞれ用いられている。簡約攻撃の先行研究として、n次元超立方格子に対する SVP 問題については、(攻撃の成功が) 証明可能な簡約攻撃が Ducas（DCC 2023）により与えられ、$n/2$次元の格子 SVP 問題に帰着されていた。$2n$次元 NTRU 格子については帰着が与えられていなかったが、Gama、Howgrave-Graham、Nguyen（Eurocrypt 2006）により、より小さな次元の格子 SVP 問題に帰着できるだろうと予想されていた。

超立方格子と NTRU 格子は、処理の効率化のため、特徴的な幾何学的性質を有している。著者らはこれら格子基底の幾何学的な特徴を利用し、BKZ に代表されるブロックワイズな簡約アルゴリズムを提案した。特に NTRU について証明可能なブロックワイズ簡約アルゴリズムを初めて与え、上述の Gama、Howgrave-Graham、Nguyen による予想を解決した。また超立方格子については、先行研究である Ducas の証明可能簡約アルゴリズムの拡張および改良となっている。

この結果を得るために著者らは、超立方格子だけでなく、$\lambda_1(L)\lambda_1(L^\times) < 1 - 1/poly(n)$を満たす格子$L$についてブロックワイズ基底簡約を与えた。ここで$L^\times$は$L$の双対格子である。特に NTRU-HPS では、復号失敗を防ぐためのパラメータ調節がなされたが、その結果$\lambda_1(L)\lambda_1(L^\times) < 1/2$が満たされるので、著者らの方式が活用できる。さらに著者らは、Ducas の方式で用いられていた SVP サブルーチンを改良した。格子の次元をnとして、Ducas の方式では、ブロックサイズ$n/2$のほぼ正確な SVP オラクルを利用していたが、著者らの方式では SVP オラクルが$\sqrt{2}$-近似を許すものへと緩和されている。結果として、SVP サブルーチンの計算時間が $2^{\alpha n/2}$ $(\alpha < 1)$ ならば、$\mathbb{Z}^n$-格子同型性判定問題を$2^{\alpha n/2 + o(n)}$で解くことができると著者らは報告した。加えて著者らは、NTRU 格子について、提案攻撃と既存のヒューリスティック攻撃を比較した。本論文で提案された証明可能な攻撃は、$n/2 + \Theta(n/\log n)$のブロックサイズが必要である一方、ヒューリスティック攻撃はブロックサイズを$4n/9 + \Theta(n/\log n)$しか必要としない。著者らはこれをヒューリスティック解析と証明可能解析の計算量の間に差が出る興味深い例として報告した。

Properties of Lattice Isomorphism as a Cryptographic Group Action [PQCrypto 2024]

Benjamin Bencina, Alessandro Budroni, Jesús-Javier Chi-Domínguez, Mukul Kulkarni

格子同型問題（Lattice Isomorphism Problem、LIP）に関する論文である。格子同型問題とは、NIST 耐量子計算機署名形式の候補である HAWK などの安全性の根拠となっている数学的プリミティブである。

格子同型問題は 2 つの格子L_1とL_2の間が同型かどうかを判定する、もしくは同型を与える直交変換を計算せよという問題である。格子に付随する二次形式に本問題を換言すれば、二次形式が整数係数可逆線型変換（すなわち$\mathrm{GL}_n(\mathbb{Z})$の元）で移りあうかを判定する、もしくはその変換を計算せよという問題となる。これは二次形式の為す空間への$\mathrm{GL}_n(\mathbb{Z})$-作用の群作用暗号（group action cryptography）に他ならないが、本論文ではこの観点を LIGA（Lattice Isomorphisms as a Group Action）と整理して研究が行われている。実際には二次形式への群作用を推移的かつ忠実とするために、$\mathrm{GL}_n^{\pm}(\mathbb{Z}) \coloneqq \mathrm{GL}_n(\mathbb{Z})/\simeq_{\pm}$を LIGA では考慮する。ここで二つの行列$A, B$が$A \simeq_{\pm} B$であるとは$A \in \{B, -B\}$を指し、これは同値関係となる。この設定の下、二次形式Qの$U \in \mathrm{GL}_n^{\pm}(\mathbb{Z})$の作用は$U \star Q \coloneqq U^{\mathrm{T}} Q U$で与えられる。この下で本論文では、群作用暗号におけるプリミティブである下記の問題を考慮している。

- 一方向性（One-wayness）: $U \in \mathrm{GL}_n^{\pm}(\mathbb{Z})$ と二次形式 Q があたえられたとき、$(Q, U \star Q)$からUを求めよ。
- Weak-unpredictability：$U \in \mathrm{GL}_n^{\pm}(\mathbb{Z})$ と多項式個の二次形式Q_iおよびPが与えられたとき、$(Q_i, U \star Q_i)$ から$U \star P$を求めよ。
- Weak-pseudorandomness : $U \in \mathrm{GL}_n^{\pm}(\mathbb{Z})$ と多項式個の二次形式Q_iおよびR_iが与えられたとき$(Q_i, U \star Q_i)$と(Q_i, R_i)を識別せよ。

本論文では、LIGA の下では Weak-unpredictability と Weak-pseudorandomness は成り立たないことが示された。そのうえで著者らは、LIP は PRF や更新可能暗号には使用すべきではないと結論づけた。また weak unpredictability を破るための別系統の手法として、サンプル数（群作用の列の長さℓ）と計算時間のトレードオフを与えるアルゴリズムも提案されている。さらに著者らは今後暗号に使える可能性のある計算問題として、二次形式同士を結び付ける Transpose Quadratic Form Problem および Inverse Quadratic Form Problem を提案している。これらの問題は search-LIP 問題に帰着されるため、困難であると期待される。

Polynomial XL: A Variant of the XL Algorithm Using Macaulay Matrices over Polynomial Rings [PQCrypto 2024]

Hiroki Furue, Momonari Kudo

MQ 問題に対する解析論文である。MQ 問題はn変数m方程式の連立 2 次方程式の求解問題であり、多変数公開鍵暗号の安全性の根拠となっている。この連立 2 次方程式の求解問

題を解くアルゴリズムの一つとして、XL アルゴリズムが知られている。

本論文では、前処理としてk変数の値を仮置きすることで計算量を下げる変種である h-XL の改良として polynomial XL を提案し、アルゴリズムの解析と$n = m$の場合の計算量の上界を与えている。XL アルゴリズムの大まかな流れは元の方程式にある単項式を Macaulay 行列により線形化し、消去法によりグレブナー基底を計算、最後に各変数の値を順番に求めていくというものである。 提案アルゴリズムでは、固定されていない$n - k$変数の方程式から多項式環$\mathbb{F}_q[x_1, \ldots, x_k]$上の Macaulay 行列を生成し、計算処理を進めることでk変数の総当たりを回避し高速化している。著者らは計算量解析の実例として、$n = m =$ 20,40,60,80に対する MQ 問題の解析時間について、h-XL,h-WXL,Crossbred 等の他のアルゴリズムと著者らの方式を比較し、優位であることを示した。

A Systematic Study of Sparse LWE [Crypto 2024]

Aayush Jain, Huijia Lin, Sagnik Saha

sparceLWE 問題に関する論文である。sparceLWE 問題は、LWE 問題と sparseLPN 問題の性質を引き継いだ計算問題として、本論文で導入された。著者らは計算困難性と暗号応用について研究している。n, m, σ, pをそれぞれ LWE 問題の次元、サンプル数、ノイズパラメータ、法とする。自然数kに対してk-sparse LWE 問題(k-sLWE)を、$(A, sA + e \bmod p)$と(A, u)を識別する問題とする。ここでuはランダムなベクトル、Aの各行はk個の非ゼロ要素しか持たないものとし、それ以外は通常の LWE 問題と同様とする。

本論文では、k-sLWE 問題の基本的な性質として、LWE 格子$L = \{xA + p\mathbb{Z}^m : x \in \mathbb{Z}^{1\times m}\}$の最小距離と密度が通常の LWE 問題とどの程度離れているかが検証され、サンプル数mが十分に大きい場合には通常の LWE と同様の挙動を示すことが示されている。また、$m = O(n^2 \log n \log p)$と設定すると Gentry らの格子署名(STOC, 2008)で用いられているようなトラップドア行列でスパースなものがサンプリング可能であることが示されている。これらにより、鍵サイズの小さい暗号方式が構成可能である。

さらに著者らは安全性の検討も行っている。sparceLWE 問題は、$k = O(\log n)$であればスパース性から総当たりが可能である。また sparseLPN に対して有効であった攻撃は、エラーベクトルが疎でないことから単純な適用は難しいと著者らは考察したが、$Ax = 0$を満たすスパースなベクトルを見つけることで攻撃の成功率を上げることが可能だと示唆した。著者らは現実的に考えられる攻撃として、Aが疎であることから、特定の$L = \Theta(n)$変数のみが非ゼロであるようなサンプルのみを集めることで次元を下げる Dense-Minor 攻撃を検討した。著者らはn'次元のk-sLWE の Dense-Minor 攻撃が、n次元mサンプルの LWE 問題となるような(n', k)を逆算するスクリプトを、lattice-estimator をベースに構成している。

最後に応用として、k-sLWE ベースの準同型暗号が効率的に構成可能であることを示した。

Quantum Lattice Enumeration in Limited Depth [Crypto 2024]

Nina Bindel, Xavier Bonnetain, Marcel Tiepelt, Fernando Virdia

SVP（最短ベクトル問題）の計算量評価についての論文である。SVP を解く代表的なアルゴリズムには列挙法（ENUM）と篩法（sieve）があり、それぞれの古典・量子計算量が評価されている。

格子次元nに対して、列挙法の時間計算量が$2^{cn\log n}, c \approx 0.1$であるのに対して、篩法の時間計算量が$2^{O(0.292n)}$であることが知られており、篩法の方が理論的に高速であるだけではなく SVP challenge のような解読コンテストの結果から、実際の計算も高速であることが知られている。

一方で、量子計算量の解析では Aono ら（Asiacrypt2018）の量子木探索を用いることで列挙法の時間計算量の指数部分が半分となる$2^{0.5cn\log n}$量子回路計算量のアルゴリズムが知られている反面、篩法の量子計算量は$2^{O(0.265n)}$であるため、次元によっては逆転する可能性が示唆されていた。

本論文では、古典/量子ハイブリッドの列挙アルゴリズムを用いて、CRYSTALS-Kyber に対する詳細な計算量評価を行っている。このハイブリッドアルゴリズムでは、列挙法で用いる列挙木の根に近い部分を古典で、残りの部分を量子で行う事を仮定している。古典と量子を切り分ける深さは NIST PQC 標準化プロジェクトで求められる量子回路の最大深さの制限から行っている。本計算量評価は、量子誤り訂正によるオーバーヘッドも加味している。ただし、量子木探索を行う部分木の大きさはインスタンスごとに異なり、実際の計算量が multiplicative Jensen's gap と呼ばれる量(Def. 1)に依存する。この gap をいくつに取ると Kyber-512,768,1024 の各パラメータの計算量が NIST で求められる AES の（深さを制限した）解読計算量よりも小さくなるのかという議論と、gap の値がどの程度ありえそうなのかが Table 5 にまとめられている。どの設定でも Kyber-1024 では AES の解読計算量よりも大きくなると考えられるが、512,768 では設定によっては小さくなるため、より細かい検証が必要と考えられる。

Not Just Regular Decoding: Asymptotics and Improvements of Regular Syndrome Decoding Attacks [Crypto 2024]

Andre Esser, Paolo Santini

シンドローム復号問題（Syndrome Decoding Problem、以下 SDP）および正則シンドローム復号問題（Regular Syndrome Decoding Problem、以下 RSDP）に関する解析論文である。

線形符号のパリティ検査行列Hとシンドロームsが与えられたときに、通常の SDP は$He = s$を満たすハミング重みw以下のエラーベクトルeを発見することを目的とする。一方 RDSP は、暗号で用いられる特徴を捉えた条件付けとして、エラーベクトルを$e = (e_1, \dots e_{n/b})$とb次元ごとに区切ったときに、各$e_i \in \mathbb{F}_2^b$のハミング重みがw以下であることを保証するもので

ある。重みが極端に大きい場合には、多項式時間の計算アルゴリズムの存在が示される。

本論文では、Both と May(PQCrypto 2018)による Information Set Decoding 型の攻撃を用いたときの SDP と RSDP との計算量比較が行われている。k/nが 0.49 よりも小さい領域においては RSD の方が計算量が小さいことが示される。続いて本論文では、新たな RSDP 向けのアルゴリズムの提案が行われ、先行研究となる Briaud and Øygarden(Eurocrypt2023)と e Carozza, Couteau and Joux(Eurocrypt2023)のアルゴリズムとの比較が行われ、わずかながらではあるが計算量が減っていることが確認された。

応用として Liu ら(Eurocrypt 2024)の疑似相関生成器(PCG)、Hazay ら(CRYPTO 2018)のマルチパーティ計算、Boyle ら（ACM CCS2019）の OT の再解析を行い、最大で 30 ビット程度計算量を改良している。

Improved Algorithms for Finding Fixed-Degree Isogenies Between Supersingular Elliptic Curves [Crypto 2024]

Benjamin Bencina, Péter Kutas, Simon-Philipp Merz, Christophe Petit, Miha Stopar, Charlotte Weitkämper

同種写像問題に関する論文である。同種写像問題とは、与えられた$\mathbb{F}_{p^2}$上の 2 つの同種な超特異楕円曲線E_1, E_2に対し、同種写像を計算する問題である。同種写像として次数dが指定されているケースもあり、その場合の計算困難性も期待されている。

本論文では、次数dが正の実数ϵで$d \approx p^{1/2+\epsilon}$となるケースで、この問題の解析を古典アルゴリズムおよび量子アルゴリズム双方で行った。古典アルゴリズムについては、$1/2 < \epsilon < 3/4$の場合に、従来知られていた中間一致アルゴリズムを、メモリ計算量および時間計算量双方で改善した。量子アルゴリズムにおいては、$0 < \epsilon < 5/2$の場合に、時間計算量を改善した。

著者らは問題を解く戦略は Deuring 対応に基づいている。すなわち、E_1, E_2の自己準同型環である虚 2 次整環$\mathcal{O}_1, \mathcal{O}_2$に対する接続イデアル（connecting ideal）を計算し、同種写像の集合$\mathrm{Hom}(E_1, E_2)$のノルム形式(norm form)を計算するアプローチをとる。このノルム形式を用いて次数dを表現し、最終的に同種写像計算に必要な表現に引き戻す。

著者らは応用として Basso ら(ACNS 2024)の SIDH 型署名の再評価を行っており、128 ビットセキュリティとされているパラメータの計算量が約2^{109}であると報告した。検証用のコードは https://github.com/isogeny-finding/improved-isogeny-finding で公開されている。

Radical N√élu Isogeny Formulae [Crypto 2024]

Thomas Decru

N-根基同種写像(radical N-isogeny)を計算するアルゴリズムに関する論文である。近年、CSIDH においてN-根基同種写像の計算を用いることで高速化する研究が行われており（Castryck ら、Asiacrypt 2022）、CSIDH-512 に適用すると既存の研究よりも 35%程度高

速化が可能である。

本論文では、奇素数Nについて、N-根基同種写像を計算する$O(N)$計算量の公式が与えられた。一方で、公式の正しさは完全には証明されておらず、数学的な予想（Conjecture 1）を仮定しているが、証明に向けたいくつかのアイデアも与えられている。

Sparse Linear Regression and Lattice Problems [TCC2024]

Aparna Gupte, Neekon Vafa, Vinod Vaikuntanathan

格子暗号の安全性に関わる、スパース線形回帰（Sparse Linear Regression、以下 SLR）に関する論文である。基底追跡、LASSO、Dantzig セレクタといったℓ_1-緩和法により、デザイン行列が特別な場合、SLR を解くことができる。しかし一般のデザイン行列に対するアルゴリズムは知られていない。また現状知られている効率的なアルゴリズムについても、平均計算量は知られていない。

本研究では、格子に対する BDD（Bounded Distance Decoding、有界距離復号）問題に関する最悪計算量を仮定した上で、SLR の平均計算量を与えている。この SLR 問題と BDD 問題の関係性において、SLR 問題のデザイン行列の制限固有値（restricted eigenvalue）条件は、BDD 問題のインスタンスを定義する格子の基底に関する条件数に対応している。この帰着により、BDD 問題の困難性を仮定した上で、デザイン行列の分布において、SLR 問題が困難な領域が与えられた。

加えて本論文では、デザイン行列が通常のガウシアンで生成されているケースを考察している。この方式で生成されるデザイン行列が識別可能な領域にある場合、LASSO により SLR 問題を解くことができるが、本論文では、この行列が識別不能な領域にある場合、例え多くのスパース解を持っていたとしても、解を求めることが困難であることを示している。この困難性は、連続的 LWE（Continuous Learning With Errors）に関する最悪計算量仮定に基づいている。これは近年、最短ベクトル問題（Shortest Vector Problem）の最悪計算量仮定と等価であることが示されており、この意味で、格子暗号の困難性と SLR 問題を関連付けている。

Worst-Case to Average-Case Hardness of LWE: An Alternative Perspective [TCC2024]

Alexandra Veliche, Divesh Aggarwal, Leong Jin Ming

格子暗号の安全性に関する、LWE（Learning With Errors）問題の困難性に関する論文である。先行研究である Regev（STOC 2005）、Peikert（STOC 2009）、Brakerski-Peikert-Langlois-Regev-Stehle（STOC 2013）によって、GapSVP および BDD 問題の最悪計算量に関する困難性から、LWE 問題の平均計算量に関する困難性が導かれていた。しかし、導出された LWE 問題の平均計算量に関する困難性の評価が最良であるかは不明であった。

本論文では、計算量の代替えとして、PPT（確率的多項式時間、Probabilistic Polynomial Time）アルゴリズムにより達成できる最大の成功確率を基に、LWE 問題の困難性が研究された。この観点から、LWE の困難性と BDD の困難性の関係性がより明らかにされている。特に、BDD 問題を解く PPT アルゴリズムについてのとある困難性予想を仮定した上で、LWE 問題を解く PPT アルゴリズムの攻撃精度の上からの評価を改良した。さらに LWE 問題について、決定論的 PPT 攻撃の攻撃精度の最高値と、計算論的 PPT 攻撃のそれの関係性も与えられた。この方式は実用的な安全性解析のフレームワークに応用しうる。

Cryptanalysis of Rank-2 Module-LIP with Symplectic Automorphisms [Asiacrypt 2024]

Hengyi Luo, Kaijie Jiang, Yanbin Pan, Anyu Wang

格子同型問題（Lattice Isomorphism Problem）についての解読論文である。格子同型問題とは、NIST 耐量子計算機署名形式の候補である HAWK などの安全性の根拠となっている数学的プリミティブである。

Eurocrypt'24 において、Mureau らは、数体$\mathbb{K}$上の加群格子に対する格子同型問題（以下、module-LIP）を定義し、$\mathbb{K}$が総実体のとき、$\mathbb{K}^2$の階数 2 部分加群の LIP 問題を解くヒューリスティックな古典アルゴリズムを提案した。このアルゴリズムはほとんどの部分加群とほとんどの総実体に対し、数論的な仮定の下で機能していた。

本論文ではこのアルゴリズムが改良され、決定論的な古典多項式時間アルゴリズムでの解法が与えられた。このアルゴリズムは全ての階数 2 部分加群と全ての総実体に機能し、数論的仮定も必要ないと報告された。

この結果を得るためのテクニックとして著者らは、（擬）シンプレクティック自己同型という新たな加群の同型を導入した上で、CM 体上の module-LIP を解くことが、ある特定のシンプレクティック自己同型を見つけることに帰着できることを示した。さらに弱（擬）シンプレクティック自己同型を効率的に計算するアルゴリズムを与えた。体が総実体である場合、これはその望ましい特定のシンプレクティック自己同型となっているため、著者らの主張した結果が従う。加えて本論文では、弱シンプレクティック自己同型は、HAWK の偽造耐性の証明で用いられた omSVP 仮定を無効にすることが報告された。ただし本論文は、HAWK に対する直接的な攻撃を与えるものではない。

On the Spinor Genus and the Distinguishing Lattice Isomorphism Problem [Asiacrypt 2024]

Cong Ling, Jingbo Liu, Andrew Mendelsohn

格子同型問題（Lattice Isomorphism Problem、以下 LIP）についての解読論文である。LIP とは、NIST 耐量子計算機署名形式の候補である HAWK などの安全性の根拠

となっている数学的プリミティブであり、二つの与えられた整格子が整数環上で同型かを判断する問題である。これらが任意の素数pについてのp進整数環上及び実数体上でも同型であるとき、二つの整格子は「同じ種数を持つ」と呼ばれる。同じ種数を持たないのであれば同型ではないことから、種数は LIP を解く上で重要な概念である一方、同じ種数を持っている場合、同型であるかを判別することは依然として困難である。

著者らはこの問題に取り組むため、種数の概念をより細分化したスピノール種数（spinor genus）を研究した。二つの整格子が「同じスピノール種数を持つ」とは、ある固有直交変換γと、任意の素数pに対しp進整数環上のスピノールノルム 1 のある局所変換δ_pが存在して、$\gamma \cdot \delta_p$でp進整数環上の格子の同型が与えられることを言う。

本論文では、同じ種数を持つ二つの格子が同じスピノール種数を持つかどうかを判定する方法が、特別な場合に与えられた。特に、同じ種数を持つ二つの階数 2 の非等方（anisotropic）な整二次形式について、同じスピノール種数を持つかどうかを量子多項式時間で判定するための必要条件が与えられた。これは特別な場合の新たな LIP の解法を与えるが、HAWK では類数が 1 より大きい数体上の階数 2 のエルミート整二次形式が用いられていることから、本論文は直接的に HAWK を攻撃するものではないことも報告された。

資料３－３

CRYPTREC 暗号リスト仕様書参照先の変更

1. 経緯と目的
   CRYPTREC 暗号の仕様書の一覧
   https://www.cryptrec.go.jp/method.html
   に掲載されている Web ページが存在しなくなっている暗号方式、アップデートされている暗号方式が、3 例ほど存在している。これらについて、適切な Web ページで確認できるようにする。

2. CRYPTREC 暗号リスト仕様書の参照先変更
   - 公開鍵暗号　－　署名　ECDSA

暗号名	ECDSA (Elliptic Curve Digital Signature Algorithm)
関連情報 1	仕様 ・ SEC 1: Elliptic Curve Cryptography (September 20, 2000, Version 1.0) https://www.secg.org/SEC1-Ver-1.0.pdf
関連情報 2	仕様 ・ ANS X9.62-2005, Public Key Cryptography for The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA) ・ 参照 URL https://www.x9.org/ ・ 2020 年 9 月に新しくなり X9.62-2005 は参照不能となり、新規バージョン　ANS X9.142-2020 のみ参照可能

（事務局案）新バージョンの Web ページに変更して問題ないと考える。
理由：旧バージョン（X9.62）の内容と新バージョン(X9.142)の内容を比較したところ、アルゴリズムの変更はないため。

・ 旧バージョンとの差異（抜粋）
   - ハッシュ関数：
     旧　ハッシュ関数は SHA-1 のみ
     新　SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3 が使える。
     　　SHA-1 は以前の署名の検証のみに使用を限定。
   - 楕円のパラメータ
     旧　群のサイズ q=2^m（m が合成数）を許可
     新　群のサイズ q=2^m（m が合成数）を不許可
   - 乱数生成機
     旧　2 種類の乱数生成機（ハッシュ関数ベースとブロック暗号ベース）に限定
     新　2 種類に限定しない

● 公開鍵暗号 － 鍵共有 ECDH

暗号名	ECDH (Elliptic Curve Diffie-Hellman Scheme)
関連情報 1	仕様 ・ SEC 1: Elliptic Curve Cryptography (September 20, 2000, Version 1.0) ・ 参照 URL https://www.secg.org/SEC1-Ver-1.0.pdf
関連情報 2	仕様 ・ NIST Special Publication SP 800-56A Revision 2(May 2013), Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography において、C(2e,0s,ECC CDH)として規定されたもの。 ・ 参照 URL https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf 新しいバージョンあり https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf

現在、比較中。2025 年度第 1 回暗号技術評価委員会で報告いたします。

● 共通鍵暗号 － 128 ビットブロック暗号 AES

暗号名	AES
関連情報	仕様 ・ NIST FIPS PUB 197, Specification for the ADVANCED ENCRYPTION STANDARD (AES), November 26, 2001 ・ 参照 URL https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf ・ 2023 年 9 月に新しくなっている 参照 URL https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf

（事務局案）新バージョンの Web ページに変更して問題ないと考える。
理由：旧バージョンの内容と新バージョンの内容を比較したところ、エディトリアルな修正であり、アルゴリズムの変更はないため。

資料３－４

2024 年度暗号技術調査 WG（耐量子計算機暗号）活動報告（案）

1　2023 年度暗号技術調査 WG（耐量子計算機暗号）活動経緯と活動内容の概要

2020 年度第 2 回暗号技術検討会において、2021 年度から暗号技術評価委員会の活動計画として 2 年をかけて PQC の研究動向を調査し、ガイドラインを作成することが決定された。暗号技術評価委員会は 2021-2022 年度に暗号技術調査ワーキンググループ(耐量子計算機暗号)を設置し、ガイドライン及び調査報告書を作成、公開した。

その後も、PQC 関連の技術開発、標準化活動が世界的に活発であることから、引き続き、暗号技術調査ワーキンググループ（耐量子計算機暗号）（以下、PQC WG）を設置して下記 2 点の活動を行うことが 2023 年度第 1 回暗号技術評価委員会において承認された。

(1) NIST の PQC 標準化において第 4 ラウンドが進行中であることをはじめ耐量子計算機暗号に関する技術開発、標準化活動が引き続き世界的に活発であることから、動向を 2023 年度から 2 年間かけて調査・把握し、ガイドラインの改定を行う。

(2) 「素因数分解の困難性に関する計算量評価」、「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新についても検討し、更新する。

2　ＷＧ委員の構成（敬称略）

主査：國廣 昇（筑波大学）
委員：青木 和麻呂（文教大学）
委員：伊藤 忠彦（セコム株式会社）
委員：下山 武司（国立情報学研究所）
委員：高木 剛（東京大学）
委員：高島 克幸（早稲田大学）
委員：成定 真太郎（KDDI 総合研究所）
委員：廣瀬 勝一（福井大学）
委員：安田 貴徳（岡山理科大学）
委員：安田 雅哉（立教大学）

3　耐量子計算機暗号ガイドラインの作成

3.1　スケジュール（2023 年度第 1 回暗号技術評価委員会で承認）

年度	回	耐量子計算機暗号ガイドラインの議論・決定・報告
2023 年度	第 1 回 2023/9/13	✓　追記・改定の方針について議論 ✓　執筆担当者を議論
	第 2 回 2024/1/19	✓　追記・改定すべき項目及びその章立ての決定 ✓　調査の中間報告
2024 年度	第 1 回 2024/7/26	✓　中間報告、追加及び削除すべき暗号方式があれば議論
	第 2 回 2025/2/3	✓　内容の確定

3.2　2023 年度第 1 回 WG（9/13）での実施内容及び決定事項

- ガイドライン及び調査報告書の作成
  - 2022 年度版ガイドライン、調査報告書をベースに 2024 年度版ガイドライン、調査報告書を作成する。改訂扱いではなく新規の扱いとすることで合意した。
  - 「はじめに」の章は 2022 年度版では同一の内容であったが、2024 年度版では他の章と同様に調査報告書に詳細な記述、ガイドラインは抜粋とする。
  - 「PQC の活用方法」の章は 2022 年度版ではガイドラインのみであったが、2024 年度版からは調査報告書にも含めることで合意。それに合わせて内容を拡充し、ガイドラインには公知の事実のみを載せ、詳細は調査報告書に載せる。
  - 以下に例示したいくつかの暗号方式の扱いに関しては今後の動向を注視し、2023 年度第 2 回以降の WG で改めて議論を行うこととした。
    - NIST 標準化が決まっているが FIPS 文書が発行されていないため詳細が流動的なもの
    - NIST Additional Signatures 候補の中で、格子、符号、多変数、同種写像、ハッシュ関数のカテゴリに含まれるもの
    - MPC-in-the-Head など新たなカテゴリとして分類されているもの

- 記載すべき項目及び章立てと執筆担当者

  ※注：目次、概要の詳細は資料 3-4 別紙に記載

	執筆担当者
i. はじめに	事務局（青野）
ii. PQC の活用方法	伊藤委員
iii. 格子に基づく暗号技術	下山委員、安田（雅）委員
iv. 符号に基づく暗号技術	成定委員
v. 多変数多項式に基づく暗号技術	安田（貴）委員
vi. 同種写像に基づく暗号技術	高島委員
vii. ハッシュ関数に基づく署名技術	廣瀬委員

- 調査活動と執筆活動の方針
  - PQC の研究成果が発表される主要な国際会議 Crypto、Eurocrypt、Asiacrypt、PQCrypto を中心に、開発・標準化の動向に関しても 2024 年 9 月 30 日までの情報を可能な限り調査する。その他主要な動向があれば可能な限り取り上げる。

- 2023 年度第 2 回 PQC WG での調査内容の報告
  各章の執筆担当者が 2023 年度第 2 回 PQC WG において、その時点までの調査内容を報告する。

3.3 2023 年度第 2 回 WG（2024/1/19）での実施内容及び決定事項

- 各章の執筆担当者が 2023 年度第 2 回 PQC WG において、その時点までの調査内容を報告。各章の大まかな更新内容が確認された。
- 調査報告書及びガイドラインの執筆方針について以下の執筆方針が決定された。
  - 1 章についても他の章と同様に、調査報告書は専門的な内容、ガイドラインには調査報告書から技術的に複雑な内容を省略し抜粋した内容とする。
  - 米国で FIPS 化が決まっている方式に関して、2024 年 9 月 30 日までに正式版が出版された場合には FIPS 版と更新部分を、FIPS 化されない場合には出版時期によって対応が異なるが、Initial draft 版とその更新差分を記述する方針とする。
  - Additional Signatures の候補を各章に記載するかどうかは執筆担当者の判断とする。
  - MPC-in-the-Head、Additional Signatures 提案方式の中でガイドライン中の計算問題の分類に含まれないものについて、大きな動きは認知されていないことから新章とはしない。

3.4 2024 年度第 1 回 WG（2024/7/26）での実施内容及び決定事項

- 各章の執筆担当者が 2024 年度第 1 回 PQC WG において、その時点までの調査内容を報告。各章の大まかな更新内容が確認された。
- 調査報告書及びガイドラインの執筆方針について以下の執筆方針が決定された。
  - 「はじめに」の章内にあるセキュリティレベルの表現について修正する。
  - 2024 年 4 月に発表された LWE 問題の量子アルゴリズムに言及する文章を記述。
  - FIPS 化の決まっている方式に関しては 2023 年度第 2 回と同様の方針（その後、2024 年 8 月に正式に FIPS 203,204,205 が発表されたことで調査報告書とガイドラインの記述方針が確定された)。

3.5 2024 年度第 2 回 WG（2025/2/3）での実施内容及び決定事項

- 各章の執筆担当者から提出された原稿に基づき、執筆担当者から執筆内容の概要が説明され、相談事項が共有された。また、以下の事項が決定された。
  - 1 章：韓国 KpqC の最終方式 4 件の情報を追記する。ガイドライン内の「現在」を 2024 年 9 月 30 日であると明記する。
  - 5 章：MAYO のパラメータが空欄となっているが公開後に追記する。
  - 全体を見直し、2 月 20 日までに執筆担当者は担当事務局員と合意の取れた原稿を完成させ事務局に提出する。
- 以下の事項は今後の検討事項とし、事務局の案をメーリングリスト上で審議することとした。

- ガイドライン内で PQC という単語が指す範囲について、他の CRYPTREC 文書の内容との整合性も考慮し、公開鍵暗号のみに限るのではない定義を考える。
- 句読点の「、。」に関して、数式中の表現との混乱を避けるための方針を決める。

- 評価委員会後にコメントがあった場合の対応について確認された。
- メーリングリストでの審議、および、関連する修正を行った上で、PQC WG 案として、ガイドライン・調査報告書の公開を暗号技術評価委員会および暗号技術検討会に付議することが承認された。
- NIST の状況を鑑みて、2025 年度以降も PQC WG を設置し、ガイドライン／調査報告書の更新を行うことが要望された。

3.6 2024 年度メール審議（2025/2/3～）実施内容及び決定事項

- 第 2 回 WG において決定された「PQC という単語がさす範囲」と「句読点の利用方針」について事務局案を作成し、メール審議を行った。
  - PQC を「古典での実装が可能かつ耐量子性を持つことを技術的に判断可能な暗号技術」とする。また、これまで「耐量子性」と表現していた用語を「耐量子計算機性」に変更することで合意。
  - 句読点は「,」「。」を用いる。ただし数式内などで全角の「,」を用いると不自然になる場合には半角「,」を用いる方針で合意。

4. 「素因数分解の困難性に関する計算量評価」、「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新

4.1 2024 年度予測図の更新

- 「今後の予測図の取り扱い」に基づいて予測図の更新を行った(図 1、2)。素因数分解問題の困難性および楕円曲線上の離散対数問題の困難性に関する計算量評価に大幅な進展はなかったため、2024 年 6 月・11 月のベンチマーク結果を追加した。

＜今後の予測図の取り扱い＞

(1) いわゆるムーアの法則を仮定して外挿線を年度末からプラス20年後まで従来どおり直線で引き、評価に大きな変動がないと考えられる限りにおいては、安全サイドに倒した評価※として予測図を当面の間更新していく。

〈今後の公開鍵暗号のパラメータ選択〉

(2) 公開鍵暗号のパラメータ選択に関する対応方針については、安全性以外にも相互接続性など、運用上の観点もあるため、暗号技術評価委員会だけではなく、暗号技術検討会、暗号技術活用委員会や関係各所などを含めて検討する。

※各国・国際標準化機関等により示されている主要な暗号技術の安全性基準と比較すると、より現状に即した評価となっており、危殆化時期は他機関等が規定している暗号技術の利用期限よりも先に延びている。

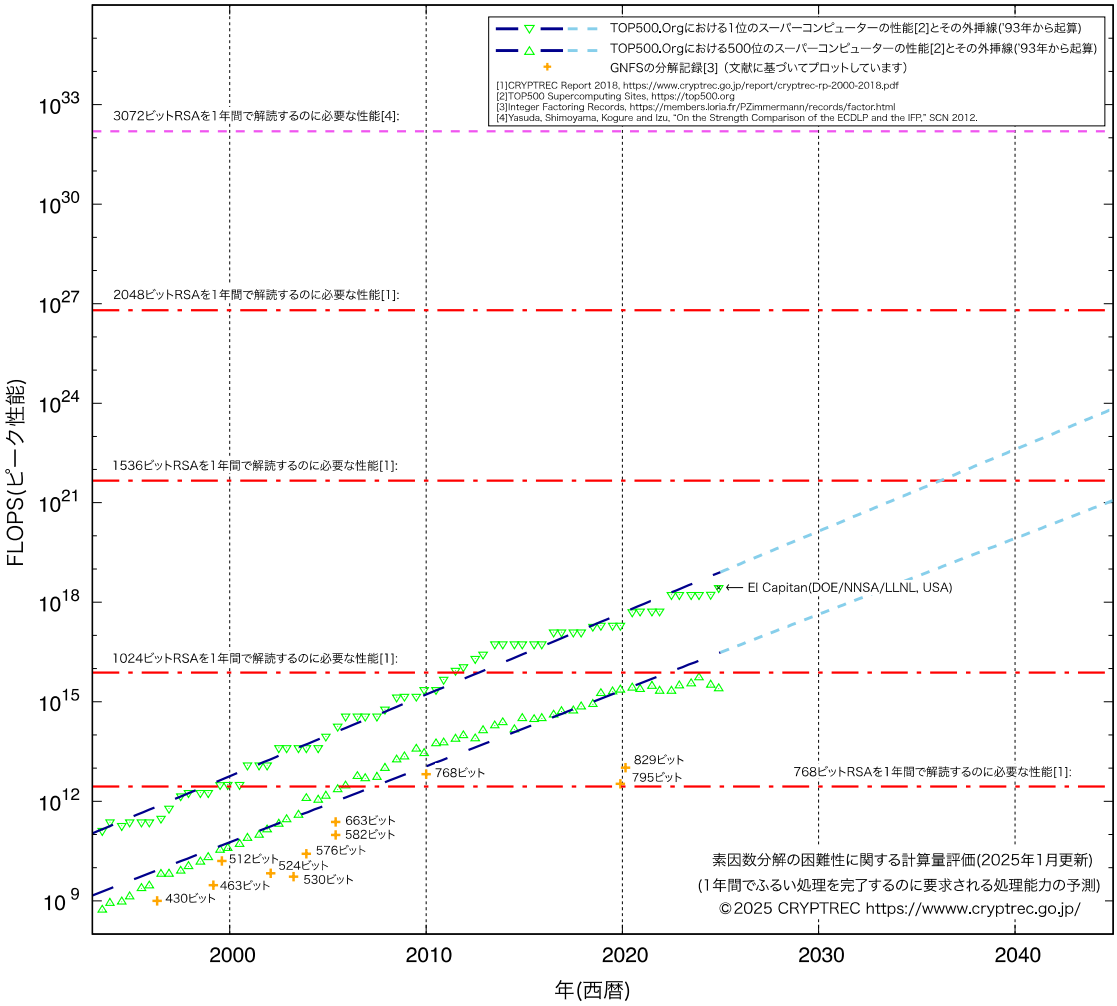


図 1：素因数分解の困難性に関する計算量評価（2025 年 1 月更新）[1]

[1] スーパーコンピューターの性能の伸びに関する外挿線は僅かであるが鈍化してきている。

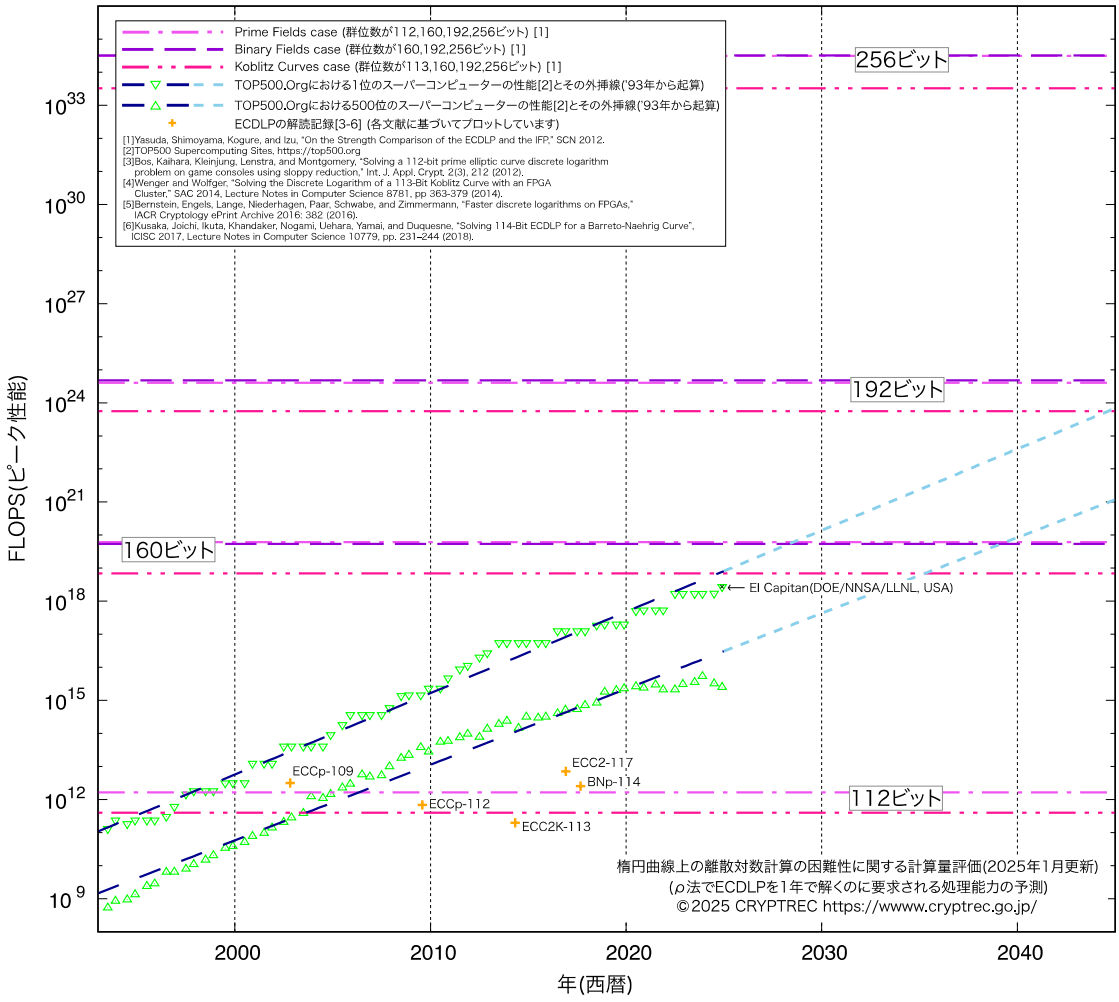


図 2：楕円曲線上の離散対数計算の困難性に関する計算量評価(2025 年 1 月更新)[2]

以上

[2] スーパーコンピューターの性能の伸びに関する外挿線は僅かであるが鈍化してきている。

資料３－４別紙

耐量子計算機暗号ガイドライン／調査報告書の更新

1. **【報告】**2024 年度版 PQC ガイドライン・調査報告書の概要

2024 年度版 PQC ガイドライン・調査報告書の概要を示す。なお、PQC ガイドラインは、PQC 調査報告書から技術的に複雑な内容を省略し、抜粋した内容となっている。このため、（１）に示す章立て（目次)、および、ガイドラインと調査報告書の概要は同じである。

（１）目次

2024 年度 PQC ガイドライン・調査報告書の目次を以下に示す。

章	章タイトル	目次
1	はじめに	1.1 暗号の安全性に影響のある量子コンピュータの開発状況 1.2 耐量子計算機暗号（PQC） の必要性について 1.3 PQC の研究及び標準化等に関する動向 1.4 本調査で対象とした PQC の種類 1.5 耐量子計算機暗号調査報告書執筆者リスト
2	PQC の活用方法	2.1 公開鍵暗号の利用形態 2.2 PQC の導入における課題 2.3 PQC 導入へのアプローチ 2.4 PQC の活用にむけて
3	格子に基づく暗号技術	3.1 格子に基づく暗号技術の安全性の根拠となる問題 3.2 格子に基づく代表的な暗号方式 3.3 格子に基づく主要な暗号方式 3.4 格子に基づく暗号技術に関するまとめ
4	符号に基づく暗号技術	4.1 符号に基づく暗号技術の安全性の根拠となる問題 4.2 符号に基づく代表的な暗号方式 4.3 符号に基づく主要な暗号方式 4.4 符号に基づく暗号技術に関するまとめ
5	多変数多項式に基づく暗号技術	5.1 多変数多項式に基づく暗号技術の安全性の根拠となる問題

		5.2 多変数多項式に基づく代表的な暗号方式 5.3 多変数多項式に基づく主要な暗号方式 5.4 多変数多項式に基づく暗号技術に関するまとめ
6	同種写像に基づく暗号技術	6.1 同種写像に基づく暗号技術の安全性の根拠となる問題 6.2 同種写像に基づく代表的な暗号方式 6.3 同種写像に基づく主要な暗号方式 6.4 同種写像に基づく暗号技術に関するまとめ
7	ハッシュ関数に基づく署名技術	7.1 ハッシュ関数に基づく署名技術の安全性の根拠となる問題 7.2 ハッシュ関数に基づく代表的な署名方式 7.3 ハッシュ関数に基づく主要な署名方式 7.4 ハッシュ関数に基づく署名技術に関するまとめ

（2）ガイドライン・調査報告書の概要

1 章の冒頭でも述べたように、ガイドラインと調査報告書の概要は同じである。以下、ガイドラインの概要として両者の概要を示す。2022 年度ガイドラインから加筆・修正した箇所を青字、新たに追加された箇所を赤字で示す。

■1 章 はじめに の内容

- ガイドライン内での耐量子計算機暗号の範囲を明確にする。古典アルゴリズムの組み合わせにより定式化され、耐量子計算機性を持つことが技術的に判断可能な暗号方式とする。
- ガイドラインの背景として、量子コンピュータの開発状況の概観と将来予測、および RSA 等の現代暗号に対する量子コンピュータの解読状況、将来予測を説明する。これらを踏まえ、直近の数年間で現代暗号に影響を及ぼす量子コンピュータが開発される可能性は低いが、数十年単位の長期的には短時間で解読可能となると想定されると説明。
- 世界各国の PQC の研究および標準化の動向について説明する。アメリカ NIST が 2016 年から PQC の、2022 年から追加の耐量子署名方式の選定プロジェクトを行っている。欧州の推奨暗号は NIST プロジェクトの上位ラウンドに残った方式が選ばれている。

■2 章 PQC の活用方法 の内容

- 耐量子計算機性を持たない暗号システムに対して耐量子計算機性を持たせるいくつかのアプローチがあることを踏まえた上で、スケーラビリティの観点から耐量子計算機性を持つ公開鍵暗号を利用することが最も汎用的な対応であると考えられることを説明する。

- 現行の公開鍵暗号の利用形態を署名、守秘、鍵共有の用途ごとに整理し、耐量子計算機暗号への移行時の課題についてまとめている。
- 耐量子計算機暗号導入のアプローチを整理し、プライオリティ設定、クリプトグラフィックアジリティ、ハイブリッド構成についてまとめている。

■3 章～7 章の暗号技術に関する章では安全性の根拠となる計算問題（格子、符号、多変数多項式、同種写像、ハッシュ関数）それぞれについて以下の要素を述べる。

- 各章の 1 節では安全性の根拠となる計算問題とその解法アルゴリズムの説明、近年の動向を述べる。また、安全性の根拠となる計算問題の公開解読チャレンジ等がある場合にはその結果を紹介する。
- 各章の 2 節では代表的な暗号方式として、次節の主要な暗号方式の理解を助けるために、構成のひな形となる教科書的な暗号の構成を紹介する。
- 各章の 3 節では主要な暗号方式として、標準化の候補となる暗号の構成および暗号パラメータについて紹介する。
- 各章の 4 節では 1～3 節のまとめを行う

■3 章～7 章の個別の内容

- 3 章（格子）では NIST の標準方式 FIPS-203(ML-KEM),FIPS-204(ML-DSA)についての記述を追記した。また、FrodoKEM が ISO 標準への予備提案を行っていることから記述を更新した。
- 4 章（符号）では安全性の根拠となる計算問題を整理し、攻撃アルゴリズムの漸近計算量に関する動向をアップデートした。また、Classic McEliece, BIKE, HQC の 3 方式が NIST PQC 標準化第 4 ラウンドに進んでいることから情報を更新した。
- 5 章（多変数多項式）ではグレブナー基底計算のアルゴリズムについて追記した。最近の署名方式の進展に従い、MPC-in-the-Head フレームワークについて追記、また署名方式 MAYO, QR-UOV, MiRitH について追記した。
- 6 章（同種写像）では近年の研究の進展を踏まえ内容の追記を行った。レベル構造付き同種写像問題およびそれに基づく鍵共有方式について追記した。また、最近の SQISign 署名方式とその変種の進展について記述した。
- 7 章（ハッシュ）では FIPS-205(SPHINCS+)についての記述を更新、追記した。

2. **【審議】**CRYPTREC 文書としての公開可否について

暗号技術調査 WG（耐量子計算機暗号）の 2023－2024 年度の活動により作成された

- 「CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）2024 年度版」（案）（資料 3-4_別紙 1）
- 「CRYPTREC 耐量子計算機暗号の研究動向調査報告書 2024 年度版」（案）

（資料 3-4_別紙 2）

につきまして、本内容で CRYPTREC のホームページ上で公開してよいかどうかご審議ください。

CRYPTREC
暗号技術ガイドライン（耐量子計算機暗号）

2025 年 2 月

CRYPTREC
暗号技術調査ワーキンググループ（耐量子計算機暗号）

目次

第 1 章

はじめに

暗号は情報を保護するための基礎的な手段である。基本的な暗号の分類として共通鍵暗号と公開鍵暗号があり，さらに公開鍵暗号の下位分類として通信相手の認証などを目的とした署名方式，情報の守秘を目的とした公開鍵暗号方式[*1]，秘密鍵の共有を目的とした鍵共有が存在する[*2] 。これらを含めた基本的な暗号方式を部品（プリミティブ）とした高機能暗号 [6] が数多く提案されている。2025 年現在，署名目的で DSA，ECDSA 等，情報の守秘目的で RSA-OAEP 等，秘密鍵共有の目的では DH，ECDH 等[*3] が国際的な標準暗号方式 [72] として用いられており，日本においても電子政府推奨暗号 [109] とされている。

これらの暗号方式の安全性と深く関わる計算問題として，素因数分解問題や楕円曲線上の離散対数問題があり，古典コンピュータ[*4]を用いる限りでは効率的に解くことが困難であると信じられている。このことから，RSA 暗号や ECDSA 署名はある程度の大きさの鍵長 [108] を用いることで安全性が保てると考えられている。一方で，Shor の量子アルゴリズム [91, 92] は上記計算問題を効率的に解くため，量子コンピュータの高性能化が情報セキュリティに影響を及ぼすとされている。古典コンピュータ上での効率的な実装が可能であり，かつ古典・量子双方のコンピュータを用いた攻撃に対しても安全性を確保できる暗号方式が必要とされている。

量子コンピュータによる攻撃への耐性は耐量子計算機性と呼ばれ，耐量子計算機性を持つ一連の暗号が耐量子計算機暗号（Post-Quantum Cryptography: PQC）と呼ばれる。耐量子計算機性の定式化はそれぞれの暗号技術の定式化を踏まえて行われており，一義的な意味で用いられる単語ではないことに注意が必要である。共通するのは古典計算機による実装が可能であるという点であり，これを以て量子暗号・量子鍵共有と分別される（例えば [62, p.3] を参照）。本ガイドライン内では特に断りのない場合，耐量子計算機暗号（PQC）の言葉を，古典アルゴリズムの組み合わせにより定式化され，かつ耐量子計算機性を持つことが安全性証明や計算量評価のように技術的に判断可能な暗号方式とする。

なお，公開鍵暗号以外の方式でも Grover の量子検索アルゴリズム [44] を用いることで共通鍵暗号方式の安全性の低下 [21] や暗号学的ハッシュ関数の衝突発見の高速化 [22] が知られており，これらのトピックに関しても多くの調査報告書が出版されている。例えば近年の量子コンピュータによる共通鍵暗号方式の安全性への影響を調査した報告書として，CRYPTREC による [113]，日本銀行金融研究所による [120] が存在する。Grover アルゴリズムの最適性 [106] から，量子コンピュータによる共通鍵暗号方式，暗号学的ハッシュ関数の攻撃計算量は鍵長の指数関数であり，公開鍵暗号と比べて影響は限定的と考えられている [113, 37]。

[*1] 本報告書の中では公開鍵暗号を Public-Key Cryptography の意味で用い，その下位分類としての Public-Key Encryption を公開鍵暗号方式と表記する。

[*2] 基本的な暗号方式の定義と性質に関しては，例えば教科書 [114, 1.3 節] などを参照。

[*3] これらの方式には多くの解説記事があるが，DH，DSA に関しては例えば [129] を，ECDH，ECDSA に関しては [112] がある。

[*4] 理論的には決定的チューリングマシンを物理的に実装した計算機で，狭義においては CMOS 半導体を用いた論理回路による計算機を指す。現在普及しているコンピュータとほぼ同義である。

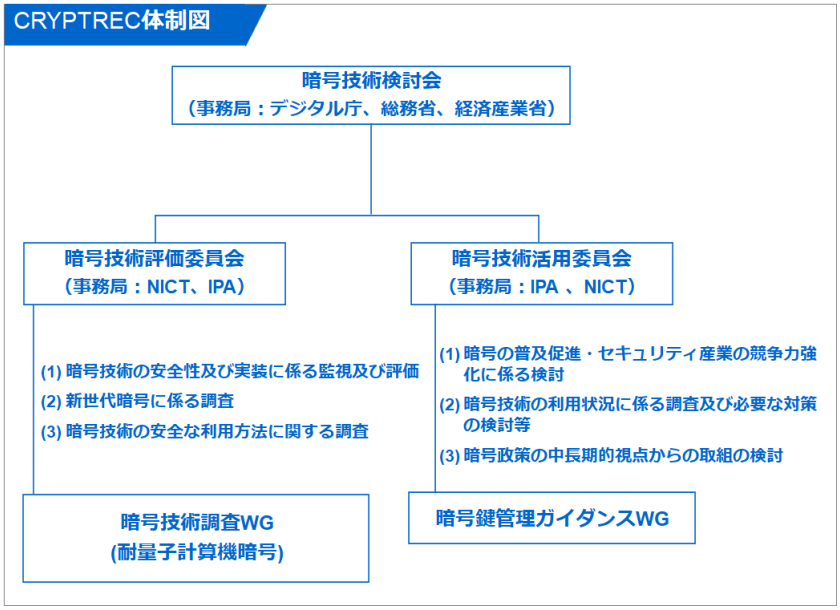


図 1.1: 2024 年度 CRYPTREC 体制図

本報告書の背景および調査内容 近年の世界的な量子コンピュータの開発と商用マシンの普及と並行して，PQC に関する研究及びその標準化に向けた活動も世界各国の組織で進んでおり，国内でも PQC の研究動向を把握する必要性がある。2020 年度第 2 回暗号技術検討会において，2021 年度から暗号技術評価委員会の活動計画として 2 年をかけて PQC の研究動向を調査し，ガイドラインを作成することが決定された。暗号技術評価委員会は暗号技術調査ワーキンググループ（耐量子計算機暗号）を設置し，ワーキンググループにおいて 2022 年 9 月 30 日までの調査結果をガイドライン [5] と調査報告書 [3] としてまとめ，出版した。その後，2022 年度第 2 回暗号技術検討会において，調査活動を継続しさらに，2 年間の研究動向調査を行い新たなガイドラインと調査報告書を作成することが決定され，暗号技術評価委員会は暗号技術調査ワーキンググループ（耐量子計算機暗号）を設置した（図 1.1）。

本ワーキンググループでは PQC の代表的な候補である 5 種類の分類（格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術，同種写像に基づく暗号技術，ハッシュ関数に基づく署名技術）について調査し，原則 2024 年 9 月 30 日までの調査結果をガイドラインと調査報告書にまとめた。本ガイドラインの中で「現在」と表記する場合，特に断りがなければ上記 2024 年 9 月 30 日時点での情報を指すものとする。

ガイドラインは暗号初学者を対象としており，調査報告書は暗号についての知見のある技術者や専門家を対象としている。第 1 章ではガイドラインと調査報告書の概要，PQC を必要とする背景，研究及び標準化に関する動向，調査対象とした PQC の種類についてまとめている。第 2 章では PQC の活用方法と移行に関する内容，特に守秘・鍵共有・署名のための PQC の利用などについて記載している。第 3 章以降では暗号技術に携わる研究者及び技術者を読者として想定し，PQC の代表的な候補である 5 種類の分類をまとめた。ただし，これらの章ではガイドラインの記載内容は調査報告書の簡略版となっており，ガイドラインでは専門的な内容を省略し，暗号初学者が代表的な PQC 方式を把握するために最小限の内容のみを記載した。

1.1 暗号の安全性に影響のある量子コンピュータの開発状況

1.1.1 量子コンピュータの分類

量子コンピュータは重ね合わせ・エンタングルメント等の量子的な物理現象を用いて計算を行うコンピュータの総称である（[116, 第 2 章]，[115] 等を参照）。基本的な計算操作と物理的操作の対応関係を表すモデルにより，量子回路型計算，測定型量子計算，断熱型量子計算，アナログ量子シミュレーション，トポロジカル量子計算，ホロノミック量子計算等に分類できる[*5]。

量子回路型および測定型量子計算モデルで計算を行う量子コンピュータは超伝導量子ビット，冷却原子（中性原子），イオントラップ，シリコン量子，光量子，カラーセンター等多くの種類の物理的実装で開発が進められている。Shor のアルゴリズムをはじめとする暗号に影響のある主要なアルゴリズムは量子回路を用いて記述されていることから，このタイプのコンピュータの大規模化が現代暗号に大きな影響を与えると考えられる。1.2.2 節に述べるように，多くの素因数分解実験が量子回路型計算のフレームワークで行われている。

断熱型量子計算と量子アニーリング　断熱型量子計算の下位分類である量子アニーリング（Quantum Annealing: QA）[*6] はクラウドサービスを通じた商用コンピュータが提供されていることから注目を集めている。量子アニーリングを用いた素因数分解実験も数多く行われている。

量子回路型計算を超伝導量子ビット，イオントラップにより実現したコンピュータ，断熱型量子計算の中でも量子アニーリングを超伝導磁束量子ビットにより実現したコンピュータは物理的なハードウェアの進化とプログラミング環境の進化により商用利用が進んでいる。これらは量子ハードウェアを専門としない技術者でもクラウドを通じて容易に利用可能であることから注目を集めていることを踏まえ，上記の量子回路型コンピュータと量子アニーリング型コンピュータを指してそれぞれ量子ゲート型と量子アニーリング型という名称で分類し対比することもある [116, 第 2 章, p.11]。

アナログ量子シミュレータ　近年，中性原子や光格子を用いた様々な実装が急速に進展している計算フレームワークである。人工的な量子系を用いて別の量子系をシミュレーションするコンピュータの総称であり [126]，古典コンピュータを用いて量子回路や量子アニーリングの出力をシミュレーションする技術とは異なる。リュードベリ原子を用いたアナログ量子シミュレータによる素因数分解実験が報告されている。

規模と性能による分類　NISQ（Noisy Intermediate-Scale Quantum）デバイス [51] は，搭載される物理量子ビットが数十から数百程度で，実行時のノイズが大きい量子デバイスを指す。量子誤り訂正や大規模な計算を行うには不十分な性能とされる。2025 年現在，全ての量子コンピュータは NISQ デバイスであると考えられる。反対に，FTQC（Fault-Tolerant Quantum Computation）デバイスは，ノイズやデコヒーレンスの影響を量子誤り訂正等を用いて低減し，大規模かつ長時間の計算を可能としたデバイスを指す。実際の暗号に用いられる大きさの素因数分解問題，離散対数問題の計算を行うためにはこの規模のコンピュータが必要と考えられている。

実際には NISQ から FTQC への発展途上でも有用な計算が可能となると期待されており，中間的な性能を指す様々な概念が提案されている。特に暗号に関係する概念として CRQC（Cryptographically Relevant Quantum Computer）

[*5] 分類に関しては [115, 102, 39] および [52, Sect 1.6] を参照。

[*6] 断熱型量子計算は基底状態が簡単に用意できる初期ハミルトニアンから，組み合わせ最適化問題の解が基底状態に対応するようなハミルトニアンへとゆっくりと変化させることで解を得る計算フレームワーク [13, Def. 1] である。量子アニーリングはこの条件を開放系，有限時間に緩め，ハミルトニアンをイジングモデルに制限した計算フレームワークを指すものと見なされている [122，§ 3]。

があり，古典コンピュータでは解くことが困難な暗号学的問題を解くことのできる量子コンピュータとして定義されている [9]。

1.1.2 ハードウェアの進展とロードマップ

前節の量子コンピュータの分類を踏まえ現在の量子コンピュータの開発状況と各組織のロードマップを概観する。より詳細な記述は調査報告書 [4] を参照。以下では，物理量子 bit は搭載されている物理的な量子ビット数を表し，論理量子 bit は量子誤り訂正などを行った後の論理レベルでの量子 bit 数を表すものとする。

量子回路型コンピュータの開発は米国の民間企業を中心に 2010 年代以降急速に発展しており，特に超伝導量子ビットによる実装 [40] と中性原子による実装 [30] が 1000 物理量子 bit を超えるプロセッサを実現している。日本国内では 2023 年 3 月に富士通と理研を中心としたチームが超伝導量子ビットを用いた 64 物理量子 bit の量子コンピュータ叡を開発，現在までに 3 台がリリースされクラウドを通じて利用されている [121, 127]。

量子アニーリングマシンの開発も超伝導磁束量子ビット型を中心に進んでおり，2020 年 9 月にリリースされた D-Wave Advantage は約 5000 物理量子 bit を搭載していた [59]。2024 年 6 月に発表された D-Wave Advantage 2 プロトタイプは約 1200 物理量子 bit を搭載 [33]，将来的に 7000 物理量子 bit 規模のアニーリングマシンを提供する予定であるとしている [32]。

しかしながら，調査の範囲で確認された量子コンピュータは総じて NISQ デバイスに留まっており，現在の暗号に対して影響を及ぼす CRQC レベルのものは確認されていない。一方で，量子回路型計算において数年前までは誤り訂正処理を行うことで逆にノイズが蓄積しエラーレートが悪化する状態であったものが，2023 年には誤り訂正後のエラーレートの方が下がるという結果が報告されており [93, 71, 107, 29]，FTQC に向け安定な論理量子ビットの構築が進んでいる。また，近年では多くのコンピュータが実験室レベルではなく，商用として開発されクラウドサービスを通じて公開されている [47, 14, 98]。

世界的に FTQC の開発を目指して研究が進められており，大きな枠組みでは例えば以下の目標が掲げられている。2020 年 1 月に決定された日本のムーンショット目標 6 では，様々な実装による量子コンピュータの開発を行い，2050 年までの FTQC 実現を目指している [117]。欧州の European Quantum Flagship が 2024 年 2 月に公表したロードマップでは，2020 年代後半に様々な実装での 1000 物理量子 bit，2030 年までに 99% 以上の忠実度[*7]をもつ 1000 論理量子 bit デバイスの実現を目標として掲げている [39]。

IBM は 2023 年 12 月にロードマップを発表し，2029 年までに実行可能ゲート数を 1 億に増やし，2033 年には実行可能ゲート数 10 億，論理量子ビットを 1000 まで上げるとしている [48]。富士通では 2024 年 5 月にロードマップ [118] を公開し，2025 年中に 256 物理量子ビットを実現し，2026 年度以降に 1000 物理量子ビットを達成するとしている。

なお，量子コンピュータの性能を十分に引き出す強力なアルゴリズムを実現するためには量子ビット数の増加のみではなく，ゲート操作の忠実度の向上，コヒーレント時間の向上などの課題を克服し，量子誤り訂正，量子ランダムアクセスメモリ等の 2025 年現在では完全には実用化されていない技術を用いる必要がある。それらの開発スピードの予測困難性が，量子コンピュータが暗号に与える影響の将来予測を困難なものとしている。

[*7] ここでは量子ゲート操作の忠実度（gate fidelity）を指す。厳密な定義は決まっていないものの，大まかに量子デバイスの出力が理想的な計算結果とどの程度一致しているかを測る指標である。

1.2 耐量子計算機暗号（PQC） の必要性について

本節では，量子コンピュータによる現代暗号への影響と PQC の必要性についてまとめる。調査の範囲では既存の量子コンピュータの性能が古典コンピュータの暗号解読性能を超えたという報告，および実社会で用いられている大きさのパラメータを持つ暗号方式が解かれたという報告は知られておらず，現代暗号に対する量子コンピュータの直接的な脅威は現時点では生じていないと考えられる。

一方で，各機関が発表しているロードマップが予定通りに達成されると仮定すると，今後数十年で RSA，ECDSA をはじめとする素因数分解問題や離散対数問題の計算困難性に基づいた暗号の解読を可能とする規模の量子計算を実行可能な量子コンピュータが開発される。暗号方式の提案から社会的な普及までは RSA 暗号・楕円曲線暗号で 20 年ほどの期間が必要とされたことから，PQC の場合でも同程度の期間が必要と想定されるため，長期間の移行スケジュールを策定し，準備を行う必要がある。

なお，2048bits の合成数を公開鍵に用いた RSA 暗号（以下，RSA-2048 と表記）は古典で 112-bit 安全性を持つとされており [108]，暗号に影響のある量子コンピュータの開発が仮に実現しなかった場合でも，古典コンピュータの性能の伸びにより長期的には危殆化すると考えられている。このことから，将来的な鍵長の変更もしくは新たな暗号方式への移行は量子コンピュータの大規模化とは独立した課題として準備を進める必要があることは長年議論されてきた [108, 18, 90] ことを指摘しておく。

1.2.1 量子コンピュータの影響による現代暗号の危殆化予測

Shor による素因数分解問題と離散対数問題に対する量子多項式時間アルゴリズム [92] が発表されて以降，RSA-2048 を危殆化させる量子コンピュータの規模の見積もり [41, 43, 125, 42] と実現時期の予測 [23, 16, 56, 66, 67] に関する研究が進められている。

量子コンピュータによる RSA-2048 の危殆化時期に関して，様々な予測が存在する。定量的な予測に基づいたものでは 2039 年以降 [23]，2050 年前後 [16] と少なくとも 20 年程度は実現に時間がかかるとされている。

セキュリティ・量子分野の専門家の予測では，Mariantoni が PQCrypto2014 の招待講演 [56] で調査に 5 年，開発に 10 年程度で 15 年後（2029 年前後）としたもの，Mosca が Workshop on Cybersecurity in a Post-Quantum World（2015 年）で 2026 から 2031 年 [66] と予測したものが有名である。近年では国際会議 RSA conference 2023 内で開かれた暗号専門家によるパネルディスカッションの中で，Shamir が RSA，DH，ECDH に影響を及ぼす 30 年か 40 年で開発される可能性があると発言している [97]。

個人ではなく，多くの専門家へのアンケートを集計した結果が 2019 年から毎年 Global Risk Institute により Quantum Threat Timeline として発行されている。2023 年に行われたアンケートを基にした予測レポート [67] では 24 時間で RSA-2048 を解読可能な量子コンピュータが 15 年以内に出現する可能性が 33% から 54% 程度であると分析している。日本国内の専門家へのアンケート調査では，2019 年に行われた文部科学省科学技術・学術政策研究所（NISTEP）による科学技術予測調査 [123, p. （II-4）48, 52] がある。この中ではある程度コヒーレンス時間の長い数百物理量子 bit 規模の量子回路コンピュータの登場を 2033 年頃と予測しているため，現代暗号に対して脅威となる量子コンピュータが出現するのはそれ以降と解釈できる。

ムーンショット目標 6 では 2050 年頃までに FTQC を実現するとしている [117] ことから，予測が実現されるのであれば現代暗号の量子コンピュータによる危殆化もその付近で起こると考えられる。

1.2.2 量子コンピュータによる素因数分解・離散対数問題計算の現状

Shor のアルゴリズムの実機実験 量子回路型コンピュータ実機を用いた実験は，CRYPTREC 外部調査報告書「Shor のアルゴリズム実装動向調査」[119] に挙げられているもの及びその後の [96, 94, 103] を含めて 15，21，35 の素因数分解実験および離散対数問題 $2^z \equiv 1 \pmod 3$ の離散対数の計算実験を行ったもののみしか知られていない。[100, 101] をはじめとする Shor のアルゴリズムを用いた初期の報告は $N = 15$ の素因数分解回路の量子フーリエ変換部分を除いた部分回路を実装する予備実験的なもの，位数や N の情報を用いて過度な簡略化を行ったものが多かった。しかし，その後 2020 年前後 IBM Quantum を用いて教科書的な簡略回路ではあるがほぼ完全な実装による実験報告 [60] や離散対数問題の実装実験報告 [16] が出版されるなど，実際に問題のインスタンスサイズには表れない量子回路規模の拡大は着実に続いていると考えられる。

Shor のアルゴリズムに関する理論の進展 1.1.2 節に紹介した量子コンピュータの性能進化がターゲットとなる数の目に見える伸びに繋がらない理由が量子コンピュータ実機の性能と Shor のアルゴリズムの性質双方の観点から検証され，明らかになりつつある。

Ichikawa らによる量子コンピュータ実機実験に関するサーベイ論文 [49] によると，実験に用いられた量子ビット数の中央値が 2016 年から 2022 年の間に 5 から 6 に増えたのみでありほぼ横ばいとなっている。量子ノイズ，デコヒーレンス等の影響により，デバイスに搭載されている物理量子ビット数と，実際に安定して動作し測定可能な物理量子ビット数の間には大きな差があり，実際に動作する回路サイズが伸び悩んでいることがわかる。また，2024 年には Cai により Shor のアルゴリズムが量子ノイズに弱い事の理論的な証明 [24] が与えられている。より大きな規模で Shor のアルゴリズムを実行するためには量子ビット数の増加だけではなく，量子ノイズの影響を下げる必要があることが理論的に示された形となる。

以上をまとめると，より大きな数の素因数分解を Shor のアルゴリズムを用いて行うためには十分にノイズが小さく，安定に動作する量子ビットを搭載した量子コンピュータが必要であると考えられる。また，Shor のアルゴリズムを用いて現在より大きな数の素因数分解を行うためには，これまでの実験のように入力インスタンスに合わせ簡略化した量子回路ではなく，汎用の剰余加算・乗算回路による構成を行う必要があるが最低でも数万ゲートの操作を必要とするという山口らの評価 [125] から，現在の量子コンピュータでは実行不可能であると考えられる。

一方，2023 年 8 月に Regev[85] により Shor のアルゴリズムよりも量子ビット数が多い代わりに量子ゲート数の少ないアルゴリズムが提案された。多くのフォロー論文 [84, 36] が発表されているものの，量子コンピュータ実機を用いた実験は確認されていない。

量子アニーリングによる実験 Shor のアルゴリズム以外の素因数分解の計算手法のうち代表的なものとして，2 進数乗算の筆算形式で式展開したものを，組み合わせ最適化問題（Quadratic Unconstrained Binary Optimization: QUBO）として定式化したものがある。QUBO とイジングモデルは自明な変換が知られていることから，量子アニーリングを中心とした断熱量子計算を用いた実験が多数報告されている。

2000 年代後半の初期の実験 [82] ではハミルトニアンに合わせて有機化合物を合成し，最適化問題の変数に対応する原子のスピンを核磁気共鳴（Nuclear Magnetic Resonance: NMR）を用いた分析により結果を取り出すという手法で計算を行っていたためスケーリングが困難であったが，D-Wave 社の量子アニーリングマシンがオンライン上で比較的手軽に利用可能になって以降は実験報告が相次いでいる [124, 104]。素因数分解のターゲットとなる数は着実に大型化しており，現時点での最大は 2023 年に D-wave Advantage 4.1 を用いた 23 ビットの 8219999=32749×251 [35] である。

その他の素因数分解手法 Shor のアルゴリズム，量子アニーリング以外の手法でも様々な素因数分解の実験が行われている。アニーリングと同様の QUBO を Quantum Approximate Optimization Algorithm （QAOA）を用いて解く実験 [83]（143，291311 を分解），Variational Quantum Eigensolver （VQE）を用いて解く実験 [95]（251 を分解），Digitized adiabatic quantum computation を用いて解く実験 [45]（2479 を分解）の報告がある。これらの実験はいずれも IBM Quantum を用いて行われている。

量子回路型コンピュータ上で QAOA を用いた素因数分解問題へのアプローチとして，Schnorr アルゴリズム [89] の部分的な量子化の研究が存在する。Schnorr アルゴリズムは数体篩法の関係探索を係数制限付きの近似最近ベクトル問題に変換して行うが，[105] ではこれをさらに最適化問題に落とし込み，QAOA を 10 量子ビット回路上で実行することで 48bits の数の素因数分解実験結果を報告している。

また，中性原子による実装の一種として，リュードベリ原子によるアナログ量子シミュレータを用いたグラフの最大独立集合問題を解くための枠組みが整理されており，これを用いた素因数分解の実験も行われている。[81] では 6,15,35 の素因数分解のインスタンスを SAT を経由して最大独立集合問題に変換して実験を行っている。

1.3 PQC の研究及び標準化等に関する動向

現在 PQC として扱われている暗号のほとんどは 1994 年に Shor のアルゴリズム [91] が発表される以前から効率性および理論的側面から研究が行われており [58, 54, 57]，2000 年代以降に耐量子計算機性が注目されたものである。

現在，PQC に関する研究成果は暗号の国際会議で主に発表されている。特に Crypto，Eurocrypt，Asiacrypt 等の暗号全般を扱う会議で取り扱われることも多いが，その他 PQC を専門に扱う国際会議として PQCrypto が 2006 年から開催され，2024 年までに 15 回が開催されている。

以下，各国における標準化の動向を述べる。米国 NIST は PQC の標準化活動を初期から大規模に行っており世界への影響力が大きいため，まず米国の状況について述べてその後に各国の状況について述べる。

1.3.1 米国 NIST における標準化の動向

2015 年 8 月国家安全保障局（NSA）が PQC への移行計画 [8] を発表したことを受け，標準化活動が国立標準技術研究所（NIST）により開始された。2016 年 2 月には福岡で開催された国際会議 PQCrypto 2016 において NIST の Moody により NIST PQC　標準化プロジェクトに関する講演 [63] が行われ，選定基準に関する意見募集を経て 12 月に Call for Proposals [77] が正式公開された。

2017 年 11 月 30 日の公募締め切りまでに世界中から耐量子計算機暗号の候補 82 方式が提案され，公募条件を満たした 69 方式が標準化プロジェクト第 1 ラウンド候補として公開されたが，5 方式は公開後に取り下げられている。2019 年 1 月 30 日には，第 2 ラウンドへ進む 26 方式が発表され，その後 2020 年 7 月 22 日には，第 3 ラウンドへ進む Finalists の 7 方式と Alternate Candidates の 8 方式が発表された [65]。

2022 年 7 月 5 日に NIST から標準化方式として公開鍵暗号 1 方式と電子署名 3 方式が発表された [12]。上記 4 方式のうち，格子に基づく公開鍵暗号方式 CRYSTALS-Kyber は FIPS 203 （ML-KEM）[74] として，格子に基づく署名方式 CRYSTALS-Dilithium は FIPS 204 （ML-DSA）[73] として，ハッシュ関数に基づく署名方式 SPHINCS$^+$ は FIPS 205（SLH-DSA）[76] として 2024 年 8 月にそれぞれ標準化されている。また，格子に基づく署名方式 FALCON についてもアルゴリズムの微修正を経た後に FIPS 206 （FN-DSA）として標準化される予定である [20]。

標準化の 4 方式が決定されると同時に，第 3 ラウンド候補の中から第 4 ラウンドへと進む公開鍵暗号方式の 4 方式が発表され，さらに追加の署名方式が再公募されることが周知された [88]。第 4 ラウンドに進んだ 4 方式のうち，BIKE,

Classic McEliece，HQC の 3 方式が符号に基づく公開鍵暗号方式，SIKE が同種写像に基づく公開鍵暗号方式であった。その後，2022 年 8 月に SIKE に対する古典多項式時間による鍵復元攻撃が発表され [25]，致命的であることが確認されたことから提案チームにより候補から取り下げられた。

署名方式の追加公募　上記第 4 ラウンドの発表と並行して，NIST は 2022 年 9 月から正式に追加の NIST PQC 標準化プロジェクト追加署名（Additional Digital Signature Schemes）の募集を開始した。締切の 2023 年 6 月 1 日までに 50 方式の応募があり，翌 7 月に公募条件を満たした 40 方式が発表された。公募の事前情報として，2022 年 7 月に pqc-forum に投稿された文書 [61] では NIST が署名長と検証時間の小さい方式を求めているとし，一例として多変数多項式に基づく署名方式の一種である UOV 方式が挙げられている。また，Module 格子のような構造化格子に基づく署名方式は既に CRYSTALS-Dilithium と FALCON が標準化に決まっていることから，構造化格子に基づく手法以外が望ましいとしており，後半の内容は募集要項にも明記された [75, p.2]。結果として格子に基づく署名は 7 方式，UOV 型の多変数多項式に基づく署名では 7 方式の応募があった。

2022 年の署名方式公募後，NIST は選考の第 1 ラウンド候補を分類ごとに発表したが，その中に 2016 年の標準化には存在しないカテゴリ MPC-in-the-Head が新たに登場している。これは，マルチパーティ計算から構成したゼロ知識証明プロトコルに Fiat-Shamir 変換を適用することで署名方式を得る構成フレームワークであり，格子，符号のように安全性の根拠となる計算問題の種類を表すものではない。MPC-in-the-Head に分類されているそれぞれの方式の安全性は実際には符号問題，多変数方程式問題，共通鍵暗号方式の平文復元問題の困難性などに帰着されている。

2024 年 10 月には第 2 ラウンドの候補となる 14 方式 [87] が発表された。格子に基づく署名方式は格子同型性判定問題を安全性の根拠とした HAWK の 1 方式，多変数多項式に基づく UOV 型署名が 4 方式，MPC-in-the-Head 型の構成を行った署名が 5 方式であった。選定に関わるレポートは [11] で公開されている。

PQC への移行　2015 年に Mosca の提案した暗号の危殆化に関わる不等式 [66] では，X（情報を保護する期間）$+Y$（システム移行期間）と Z（CRQC 開発までの期間）の大小関係によりシステム移行の準備期間を設定する必要があるとしている。一方で，暗号化データを保存し，将来的にコンピュータの性能が上がってから解読するハーベスト攻撃（2.2.2 節も参照）を想定すると，CRQC 開発までの年数によらず，現在の暗号利用にはリスクがあるとも考えられている（例えば [64, Sec. 1] を参照。）以上の背景のもと，2022 年 5 月公表された国家安全保障覚書 NSM-10[68] では 2035 年を目処に暗号システムを PQC に移行することを目標としている。同様に，2022 年 9 月に発表された商用国家安全保障アルゴリズムのリスト 2.0[7] では 2035 年までにシステムに耐量子計算機性をもたせることを目標としたタイムラインを掲載している。

現在使われている暗号から PQC への移行を推進するため，NIST 内の NCCoE（National Cybersecurity Center of Excellence）を中心にコンソーシアムが設立された [38]。組織における暗号のユースケース，相互運用性やリスク評価を含めた移行計画の策定に関する包括的な技術文書が NIST SP 1800-38A から 38C として発行される予定であり，現在は Initial Preliminary Draft[70] が公開されている。

安全性レベル　NIST PQC 標準化プロジェクトにおいて，暗号方式の安全性はレベル 1 から 5 で定義されており，提案者は応募時にパラメータと達成される安全性レベルを示す必要があった。レベル 1, 3, 5 はそれぞれ AES128, AES192, AES256 などの 128, 192, 256bits の秘密鍵を持つブロック暗号の鍵復元の困難性と同等かそれ以上の計算量であり，レベル 2 と 4 はそれぞれ SHA256/SHA3-256 と SHA384/SHA3-384 などの 256bits と 384bits の暗号学的ハッシュ関数の衝突探索の困難性と同等かそれ以上の古典もしくは量子計算量とされている。レベル 1 から 5 の具体的な計算量は表 1.1 で与えられる [75]。古典コンピュータによる攻撃者に対しては古典論理回路のゲート数が，量子コンピュータを利用可能な攻撃者に対しては量子回路のゲート数と最大深さの積が与えられている。計算量評価において，公開鍵暗号方式では，IND-CCA2 安全性を考える際には 2^{64} 個以下の選択暗号文を復号オラクルに古典的にクエリできると

し，署名方式では，EUF-CMA 安全性を考える際には 2^{64} 個以下のメッセージを署名オラクルに古典的にクエリできるとしている。

また，レベル 1,3,5 の量子回路計算量は 2022 年度版調査報告表の中で $2^{157}, 2^{221}, 2^{285}$ とされている部分は 2016 年版では $2^{170}, 2^{233}, 2^{298}$ であった。つまり，2016 年の PQC 候補でレベル 1,3,5 とされているものは 2022 年の定義でもレベル 1,3,5 の基準を満たすことになる。この更新は AES を解読する量子回路の改良により，量子計算量が改善されたことによる。

表 1.1: 2022 年に公表された NIST PQC 標準化プロジェクト追加署名 Call for proposals [77] における安全性レベルと計算量の対応表。各レベルは古典，量子のどちらか一方の基準を満たすものとして定義されている。

レベル	量子回路の（最大深さ）×（ゲート数） *8	古典論理ゲート数
レベル 1	2^{157}	2^{143}
レベル 2	–	2^{146}
レベル 3	2^{221}	2^{207}
レベル 4	–	2^{210}
レベル 5	2^{285}	2^{272}

1.3.2 米国以外での動向

米国以外でも世界各国の機関が調査活動を行い，調査レポートの出版 [17, 69]，各国における PQC 標準方式，推奨暗号の選定 [7, 27, 15, 90, 26, 28, 78, 99] を進めている。国際的な機関では ISO/IEC[50]，IETF[46] 等で移行，標準化の議論が進められている。

各国の政府機関から PQC の標準暗号リスト，推奨暗号リストが公表されている。代表的なものを表 1.2 にまとめる。多くの国が NIST PQC の標準化方式を用いているが，FrodoKEM のように NIST PQC 標準化プロジェクトの第 3 ラウンド以降の選考に漏れた方式，Classic McEliece のように第 4 ラウンド選考中の状態で選ばれた例も存在する。また，多くの機関が NIST 標準方式の単独利用ではなく古典的安全性がよく知られている RSA や ECDSA とのハイブリッドを推奨していること，レベル 3 以上のパラメータ利用を推奨していることも特徴的である。国家による標準暗号以外でも Streamlined NTRU Prime[19] のように OpenSSH の実装 [80] を通じて実用化されている方式も存在する。

韓国では量子耐性暗号研究団の主催する KpqC プロジェクト [128] が耐量子計算機性を持つ公開鍵暗号方式と署名方式の公募を 2022 年に開始している。2022 年 10 月の締切までに公開鍵暗号・鍵共有が 8 方式，署名が 9 方式応募された。2022 年 11 月に第 1 ラウンドを開始，2023 年 12 月に第 2 ラウンドの選考が開始され，2025 年 1 月に共通鍵暗号に基づく MPC-in-the-Head パラダイムの署名方式 AIMer，および格子に基づく公開鍵暗号方式 NTRU+，署名方式 HAETAE，鍵交換方式 SMAUG-T の 4 方式が最終方式として選ばれたことがアナウンスされた。

中国では中国暗号学会（CACR）が中心となり PQC の公募を行っている [110]。2018 年 6 月の募集要項に従い 2019 年 2 月の締切までに公開鍵暗号 38 方式と共通鍵暗号 22 方式が応募されている。2019 年 9 月の第 2 ラウンドの時点で公開鍵暗号 14 方式と共通鍵暗号 10 方式に絞られ，最終的には 2020 年 1 月に一等，二等，三等としてランク付けが発表された [111]。一等として公開鍵暗号方式 LAC.PKE，Aigis-enc，署名 Aigis-sig，共通鍵暗号方式 uBlock，Ballet が挙げられている。

日本では CRYPTREC の暗号技術調査ワーキンググループにおいて 2014 年度に PQC の代表的な候補である格子

表 1.2: 世界各国の標準暗号，推奨暗号リストの状況。表中の勧告，推奨，許容等はそれぞれのレポートからの翻訳であるため，厳密に同じ意味ではない。許容されているバージョン，安全性レベルなど，詳細は引用先のレポートを参照のこと。

方式の名称	NIST PQC（米）	CNSA 2.0（米）[7]	NCSC（英）[27]	ANSSI（仏）[15]	BSI（独）[90]	NCSC（蘭）[26, 28]	NÚKIB（チェコ）[78]	TRAFICOM（フィンランド）[99]
ML-KEM（CRYSTALS-Kyber）	標準化（FIPS 203 [74]）	勧告 ^{a}	推奨 ^{c}	許容 ^{d}	推奨 ^{ef}	推奨	推奨 ^{h}	暗号要件 ^{k}
FrodoKEM	Round 3	–	–	許容 ^{d}	推奨 ^{e}	許容	許容 ^{i}	–
Classic McEliece	Round 4	–	–	–	推奨 ^{e}	許容	許容 ^{i}	–
ML-DSA（CRYSTALS-Dilithium）	標準化（FIPS 204 [73]）	勧告 ^{a}	推奨 ^{c}	許容 ^{d}	推奨 ^{ef}	推奨/許容 ^{g}	推奨 ^{h}	暗号要件 ^{k}
FN-DSA（FALCON）	標準化中	–	–	許容 ^{d}	–	推奨/許容 ^{g}	推奨 ^{i}	–
XMSS/LMS	標準化（NIST SP 800-208）	勧告 ^{b}	推奨	許容 ^{d}	推奨	推奨/許容 ^{g}	推奨 ^{j}	–
SLH-DSA SPHINCS^{+}	標準化（FIPS 205 [76]）	–	推奨 ^{c}	許容 ^{d}	推奨 ^{ef}	推奨/許容 ^{g}	推奨 ^{i}	暗号要件 ^{k}

注釈一覧	
a	汎用量子アルゴリズムとしてレベル 5 パラメータの使用を勧告
b	ソフトウェア・ファームウェアに対する署名のための使用を勧告
c	標準化の最終文書を元に堅牢な実装がされたものの利用を推奨
d	メインストリームの PQC として適切だが，可能な限りパラメータを大きく取ること
e	古典的な安全性が確保された方式とのハイブリッドのみを推奨
f	NIST 標準化の安全性レベル 3,5 を推奨パラメータとする意向
g	緊急性のシナリオによって推奨と許容の方式が異なる
h	単独利用は安全性レベル 5 のみ，他は古典の推奨暗号とのハイブリッド利用とする
i	古典の推奨暗号とのハイブリッド利用とする
j	ファームウェア・ソフトウェアの保護目的での単独利用を推奨
k	古典の推奨暗号とのハイブリッド利用を推奨

に基づく暗号技術について調査を行い，報告書「格子問題等の困難性に関する調査」を公開している [1]。さらに 2017 年度から 2018 年度にかけて，PQC の代表的な候補である 4 種類の分類（格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術，同種写像に基づく暗号技術）について調査し，報告書にまとめた [2]。また，2021 年度から 2022 年度にかけて，上記 4 種類に加えてハッシュ関数に基づく署名技術を加えた 5 種類の技術について調査を行い，報告書 [3] およびガイドライン [5] としてまとめている。

1.4 本調査で対象とした PQC の種類

本調査報告書では PQC の調査を格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術，同種写像に基づく暗号技術，ハッシュ関数に基づく署名技術の 5 分類で行った。この分類は，安全性の根拠となる数学的な計算問題の種類に基づいて行われている。

例えば，教科書的な RSA 暗号では 2 つの異なる大きな素数 p, q と指数 d を秘密鍵，積 $N = pq$ と指数 e を公開鍵と

している。鍵復元の困難性と素因数分解問題の困難性は確率的多項式時間等価であるため [86]，RSA 暗号の鍵復元の安全性は素因数分解問題の困難性に基づくものと考えることができ，素因数分解に基づく暗号に分類できる。同様に，楕円曲線暗号の場合も例えば楕円曲線上の ElGamal 暗号のように安全性が楕円曲線上の離散対数問題の困難性に基づくため，離散対数問題に基づく暗号に分類できる。

本ガイドライン・報告書で扱う代表的な 5 種類の PQC （格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術, 同種写像に基づく暗号技術，ハッシュ関数に基づく署名技術） も RSA 暗号等と同様に，暗号の安全性がそれぞれ格子問題の困難性，符号復号問題の困難性，多変数代数方程式の求解困難性，同種写像の計算困難性，ハッシュ関数の衝突発見困難性に基づいている。そして，これらの問題を量子コンピュータを利用して効率よく解くアルゴリズムはまだ発見されていないことから，上記の暗号は PQC であると期待されている。暗号方式と数学的な計算問題の具体的な関係は各章の第 1 節に記載されている。

上記 5 種類を選んだ理由は主に研究期間の長さ，研究コミュニティの大きさによる。より細かい歴史的な背景は各章の第 4 節に記載されている。

格子に基づく暗号技術は 1997 年の Ajtai と Dwork による論文 [10] から 25 年以上の歴史を持ち，解読技術である格子アルゴリズムに関しても 50 年の歴史を持つ [34, 53, 55]。符号に基づく暗号技術は McEliece による 1978 年の論文 [58] から 40 年以上の歴史を持ち，解読技術は通信における符号の復号技術であり符号理論として 70 年以上研究が行われている。多変数多項式に基づく暗号技術は Ong と Schnorr による 1983 年の論文 [79] を源流[*9]とし，1988 年の松本-今井暗号 [57] を経て 40 年以上の研究が続けられている。同種写像の計算問題に基づく暗号技術も Couveignes による 1997 年の提案 [31] から 25 年以上研究が続けられている。ハッシュ関数に基づく署名方式は Lamport による 1979 年の論文 [54] から 40 年以上の研究が行われている。

[*9] ただし Ong と Schnorr による方式の安全性は素因数分解問題に基づくため耐量子計算機性を持たないことに注意。

1.5　耐量子計算機暗号調査報告書執筆者リスト

主査	國廣 昇	筑波大学
委員	青木 和麻呂	文教大学
委員	伊藤 忠彦	セコム株式会社
委員	下山 武司	国立情報学研究所
委員	高木 剛	東京大学
委員	高島 克幸	早稲田大学
委員	成定　真太郎	KDDI 総合研究所
委員	廣瀬 勝一	福井大学
委員	安田 貴徳	岡山理科大学
委員	安田 雅哉	立教大学
事務局	青野 良範	情報通信研究機構
事務局	五十部 孝典	情報通信研究機構
事務局	伊藤 竜馬	情報通信研究機構
事務局	大久保 美也子	情報通信研究機構
事務局	大東 俊博	情報通信研究機構
事務局	小川 一人	情報通信研究機構
事務局	金森 祥子	情報通信研究機構
事務局	黒川 貴司	情報通信研究機構
事務局	高安 敦	情報通信研究機構
事務局	横山 和弘	情報通信研究機構
事務局	吉田 真紀	情報通信研究機構
事務局	篠原 直行	情報通信研究機構

第 1 章の参照文献

[1] CRYPTREC 暗号技術調査 WG (暗号解析評価). 格子問題等の困難性に関する調査. CRYPTREC EX-2404-2014, https://www.cryptrec.go.jp/exreport/cryptrec-ex-2404-2014.pdf. 2015-03.

[2] CRYPTREC 暗号技術調査 WG (暗号解析評価). 耐量子計算機暗号の研究動向調査報告書. CRYPTREC TR-2001-2018, https://www.cryptrec.go.jp/report/cryptrec-tr-2001-2018.pdf. 2019-04.

[3] CRYPTREC 暗号技術調査 WG (耐量子計算機暗号). CRYPTREC 耐量子計算機暗号の研究動向調査報告書. CRYPTREC GL-2004-2022, https://www.cryptrec.go.jp/report/cryptrec-tr-2001-2022.pdf. 2023-03.

[4] CRYPTREC 暗号技術調査 WG (耐量子計算機暗号). CRYPTREC 耐量子計算機暗号の研究動向調査報告書. CRYPTREC GL-2004-****, https://www.cryptrec.go.jp/report/cryptrec-tr-2001-****.pdf. 2025-03.

[5] CRYPTREC 暗号技術調査 WG (耐量子計算機暗号). CRYPTREC 暗号技術ガイドライン (耐量子計算機暗号). CRYPTREC GL-2004-2022, https://www.cryptrec.go.jp/report/cryptrec-gl-2004-2022.pdf. 2023-03.

[6] CRYPTREC 暗号技術調査 WG (高機能暗号). CRYPTREC 暗号技術ガイドライン (高機能暗号). CRYPTREC GL-2005-2022, https://www.cryptrec.go.jp/report/cryptrec-gl-2005-2022.pdf. 2023-03.

[7] National Security Agency. Announcing the Commercial National Security Algorithm Suite 2.0. https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF. 2022-09. (2024-12-06 閲覧).

[8] National Security Agency. Cryptography Today. https://web.archive.org/web/20150815072948/https://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml. 2015-08. (2024-12-05 Internet Archive 版を確認).

[9] National Security Agency. Frequently Asked Questions, Quantum Computing and Post-Quantum Cryptography. https://media.defense.gov/2021/Aug/04/2002821837/-1/-1/1/Quantum_FAQs_20210804.PDF. 2021-08. (2024-12-01 閲覧).

[10] M. Ajtai, Cynthia Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. STOC. ACM, 1997, pp. 284–293.

[11] G. Alagic et al. Status Report on the First Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8528, https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8528.pdf. 2024-10.

[12] G. Alagic et al. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8413, https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413-upd1.pdf. 2022-07.

[13] T. Albash, D. A. Lidar. Adiabatic quantum computation. Rev. Mod. Phys. Vol. 90, Iss. 1 (2018), p. 015002.

[14] Amazon Braket 量子コンピュータ. https://aws.amazon.com/jp/braket/quantum-computers/.

[15] ANSSI. ANSSI views on the Post-Quantum Cryptography transition (2023 follow up). https://cyber.gouv.fr/sites/default/files/document/follow_up_position_paper_on_post_quantum_cryptography.pdf. 2023-12. (2024-12-06 閲覧).

[16] Y. Aono, S. Liu, T. Tanaka, S. Uno, R. Van Meter, N. Shinohara, R. Nojima. The Present and Future of Discrete Logarithm Problems on Noisy Quantum Computers. IEEE Transactions on Quantum Engineering. Vol. 3 (2022), pp. 1–21.

[17] GSM Association. Post Quantum Cryptography – Guidelines for Telecom Use Cases. https://www.gsma.com/newsroom/wp-content/uploads/PQ.03-Post-Quantum-Cryptography-Guidelines-for-Telecom-Use-v1.0.pdf. 2024-02. (2024-12-06 閲覧).

[18] E. Barker. Recommendation for Key Management: Part 1 – General. NIST SP 800-57 Part 1 Rev. 5, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf. 2020-05.

[19] D. J. Bernstein et al. NTRU Prime. https://ntruprime.cr.yp.to/. (2024-12-06 閲覧).

[20] C. Boutin. NIST Releases First 3 Finalized Post-Quantum Encryption Standards. https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards. 2023-08. (2024-12-06 閲覧).

[21] G. Brassard. Searching a Quantum Phone Book. Science. Vol. 275, Num. 5300 (1997), pp. 627–628.

[22] G. Brassard, P. Høyer and A. Tapp. Quantum Cryptanalysis of Hash and Claw-Free Functions. LATIN. Vol. 1380. Lecture Notes in Computer Science. Springer, 1998, pp. 163–169.

[23] J. Sevilla and C. J. Riedel. Forecasting timelines of quantum computing. (2020). arXiv: 2009.05045.

[24] J.-Y. Cai. Shor's algorithm does not factor large integers in the presence of noise. Science China Information Sciences. Vol. 67, Num. 7 (2024).

[25] W. Castryck, T. Decru. An Efficient Key Recovery Attack on SIDH. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 423–447.

[26] National Cyber Security Centre. Guidelines for quantum-safe transport-layer encryption. https://www.ncsc.nl/documenten/publicaties/2022/juli/guidelines-for-quantum-safe-transport-layer-encryption/guidelines-for-quantum-safe-transport-layer-encryption. 2022-07. (2024-12-06 閲覧).

[27] National Cyber Security Centre. Next steps in preparing for post-quantum cryptography. https://www.ncsc.gov.uk/pdfs/whitepaper/next-steps-preparing-for-post-quantum-cryptography.pdf. 2024-08. (2024-12-06 閲覧).

[28] National Cyber Security Centre. The PQC Migration Handbook, Guidelines for migrating to post-quantum cryptography (Version 2). https://publications.tno.nl/publication/34641918/oicFLj/attema-2023-pqc.pdf. 2023-12. (2024-12-06 閲覧).

[29] Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. Nature. Vol. 616, Num. 7955 (2024).

[30] Atom Computing. Quantum startup Atom Computing first to exceed 1,000 qubits. https://atom-computing.com/quantum-startup-atom-computing-first-to-exceed-1000-qubits/. 2023-10. (2024-12-01 閲覧).

[31] J.-M. Couveignes. Hard Homogeneous Spaces. Cryptology ePrint Archive, Paper 2006/291. 2006. https://eprint.iacr.org/2006/291.

[32] D-Wave. Ahead of the Game: D-Wave Delivers Prototype of Next-Generation Advantage2 Annealing Quantum Computer. https://www.dwavesys.com/company/newsroom/press-release/ahead-of-the-game-d-wave-delivers-prototype-of-next-generation-advantage2-annealing-quantum-computer/. 2022-06. (2024-12-01 閲覧).

[33] D-Wave. The Most Connected and Powerful Quantum Computer Built for Business. https://www.dwavesys.com/solutions-and-products/systems/. (2024-12-01 閲覧).

[34] U. Dieter. How to calculate shortest vectors in a lattice. Mathematics of Computation. Vol. 29 (1975), pp. 827–833.

[35] J. Ding, G. Spallitta, R. Sebastiani. Experimenting with D-Wave quantum annealers on prime factorization problems. Frontiers Comput. Sci. Vol. 6 (2024).

[36] M. Ekerå, J. Gärtner. Extending Regev's Factoring Algorithm to Compute Discrete Logarithms. PQCrypto (2). Vol. 14772. Lecture Notes in Computer Science. Springer, 2024, pp. 211–242.

[37] ETSI. Quantum-Safe Cryptography (QSC); Limits to quantum computing applied to symmetric key sizes. https://www.etsi.org/deliver/etsi_gr/QSC/001_099/006/01.01.01_60/gr_QSC006v010101p.pdf. 2017-02. (2024-12-01 閲覧).

[38] National Cybersecurity Center of Excellence. Migration to Post-Quantum Cryptography. https://www.nccoe.nist.gov/crypto-agility-considerations-migrating-post-quantum-cryptographic-algorithms.

[39] European Quantum Flagship. Strategic Research and Industry Agenda. https://qt.eu/media/pdf/Strategic-Reseach-and-Industry-Agenda-2030.pdf. 2024-02.

[40] J. Gambetta. The hardware and software for the era of quantum utility is here. https://jila.colorado.edu/qip2019/qip2019_posters_monday.pdf. 2023-12. (2024-12-01 閲覧).

[41] C. Gidney, M. Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum. Vol. 5 (2021), p. 433.

[42] É. Gouzien, D. Ruiz, F.-M. Le Régent, J. Guillaud, N. Sangouard. Performance Analysis of a Repetition Cat Code Architecture: Computing 256-bit Elliptic Curve Logarithm in 9 Hours with 126 133 Cat Qubits. Phys. Rev. Lett. Vol. 131, Iss. 4 (2023), p. 040602.

[43] É. Gouzien, N. Sangouard. Factoring 2048-bit RSA Integers in 177 Days with 13 436 Qubits and a Multimode Memory. Phys. Rev. Lett. Vol. 127, Iss. 14 (2021), p. 140503.

[44] L. K. Grover. A fast quantum mechanical algorithm for database search. STOC. ACM, 1996, pp. 212–219.

[45] N. N. Hegade, K. Paul, F. Albarrán-Arriagada, X. Chen, E. Solano. Digitized adiabatic quantum factorization. Phys. Rev. A. Vol. 104, Iss. 5 (2021), p. L050403.

[46] P. E. Hoffman, S. Celi. Post-Quantum Use In Protocols (pquip). https://datatracker.ietf.org/wg/pquip/about/. (2024-12-06 閲覧).

[47] IBM Quantum Platform. https://quantum.ibm.com/.

[48] IBM、次世代量子プロセッサーおよび IBM Quantum System Two を発表するとともに、 実用的な量子コンピューティングの時代の前進に向けロードマップを拡張. https://jp.newsroom.ibm.com/2023-12-05-IBM-Debuts-Next-Generation-Quantum-Processor-IBM-Quantum-System-Two,-Extends-Roadmap-to-Advance-Era-of-Quantum-Utility. 2023-12. (2024-12-01 閲覧).

[49] T. Ichikawa et al. Current numbers of qubits and their uses. Nature Reviews Physics. Vol. 6, Num. 6 (2024), pp. 345–347.

[50] ISO. PQCRYPTO Post-quantum cryptography for long-term security. https://www.iso.org/organization/5984715.html. (2024-12-06 閲覧).

[51] P. John. Quantum Computing in the NISQ era and beyond. Quantum. Vol. 2 (2018), p. 79.

[52] S. P. Jordan. Quantum Computation Beyond the Circuit Model. 2008. arXiv: 0809.2307.

[53] R. Kannan. Improved Algorithms for Integer Programming and Related Lattice Problems. STOC. ACM, 1983, pp. 193–206.

[54] L. Lamport. Constructing digital signatures from a one-way function. SRI International Technical Report, CSL-98. 1979-10.

[55] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen. Vol. 261, Num. 4 (1982), pp. 515–534.

[56] M. Mariantoni. Building a superconducting quantum computer (Invited Talk). PQCrypto 2014. 2024-10. (2024-12-01 閲覧) 暗号危殆化の予測については動画 https://www.youtube.com/watch?v=wWHAs--HA1c の 49:30 で述べられている.

[57] T. Matsumoto, H. Imai. Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. EUROCRYPT. Vol. 330. Lecture Notes in Computer Science. Springer, 1988, pp. 419–453.

[58] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report. Vol. 44 (1978), pp. 114–116.

[59] C. McGeoch, P. Farre. D-Wave Advantage システム: 概要. https://dwavejapan.com/app/uploads/2020/12/14-1049A-A_J-The_D-Wave_Advantage_System_An_Overview_0-.pdf. 2020-12. (2024-12-01 閲覧).

[60] A. Mirko, Z. H. Saleem, K. Muir. Experimental study of Shor's factoring algorithm using the IBM Q Experience. Physical Review A. Vol. 100, Iss. 1 (2019), p. 012305.

[61] D. Moody. Announcement: The End of the 3rd Round – the First PQC Algorithms to be Standardized. https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/G0DoD7lkGPk/m/f3Hl0sh3AgAJ. 2022-07. (2024-12-06 閲覧).

[62] D. Moody. Are we there yet? An Update on the NIST PQC Standardization Project. https://csrc.nist.gov/csrc/media/Presentations/2024/update-on-the-nist-pqc-standardization-project/images-media/moody-are-we-there-yet-pqc-pqc2024.pdf. 2024-04. (2024-12-01 閲覧).

[63] D. Moody. Post-Quantum Cryptography: NIST's Plan for the Future. https://pqcrypto2016.jp/data/pqc2016_nist_announcement.pdf. 2016-02. (2024-12-06 閲覧).

[64] D. Moody, R. Perlner, A. Regenscheid, A. Robinson, D. Cooper. Transition to Post-Quantum Cryptography Standards. NIST IR 8547 (initial public draft), https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf. 2024-11. (2025-02-17 閲覧).

[65] D. Moody et al. Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8309, https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf. 2020-07.

[66] M. Mosca. Cybersecurity in a quantum world: will we be ready? Workshop on Cybersecurity in a Post-Quantum World. Session 8. 2015-04. (2024-02-29 閲覧).

[67] M. Mosca, M. Piani. 2023 Quantum Threat Timeline Report. https://globalriskinstitute.org/publication/2023-quantum-threat-timeline-report/. 2023-12. (2024-12-02 閲覧).

[68] National Security Memorandum on Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems. https://www.whitehouse.gov/briefing-room/statements-releases/2022/05/04/national-security-memorandum-on-promoting-united-states-leadership-in-quantum-computing-while-mitigating-risks-to-vulnerable-cryptographic-systems/. 2022-05. (2025-01-11 閲覧).

[69] NCSA. Guidelines for Post – Quantum Readiness. https://www.navy.mi.th/storage/frontend/article/23852/file/th/Quantum%20Readiness.pdf. 2023-12. (2024-12-06 閲覧).

[70] W. Newhouse et al. Migration to Post-Quantum Cryptography: Preparation for Considering the Implementation and Adoption of Quantum Safe Cryptography. NIST SP 1800-38 (initial preliminary draft), https://csrc.nist.gov/pubs/sp/1800/38/iprd-(1). 2023-12. (2025-02-17 閲覧).

[71] Z. Ni et al. Beating the break-even point with a discrete-variable-encoded logical qubit. Nature. Vol. 616, Num. 7955 (2023), pp. 56–60.

[72] NIST. Digital Signature Standard (DSS). NIST FIPS 186-5, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf. 2023-02.

[73] NIST. Module-Lattice-Based Digital Signature Standard. NIST FIPS 204, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf. 2024-08.

[74] NIST. Module-Lattice-Based Key-Encapsulation Mechanism Standard. NIST FIPS 203, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf. 2024-08.

[75] NIST. Standardization of additional digital signature schemes, call for proposals. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf. 2022-10. (2024-03-05 閲覧).

[76] NIST. Stateless Hash-Based Digital Signature Standard. NIST FIPS 205, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf. 2024-08.

[77] NIST. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf. 2016-12. (2024-03-05 閲覧).

[78] NÚKIB. Minimum requirements for cryptographic algorithms – Cryptographic security recommendations. https://nukib.gov.cz/download/publications_en/Minimum_Requirements_for_Cryptographic_Algorithms_final.pdf. 2023-11. (2024-12-06 閲覧).

[79] H. Ong, C. P. Schnorr. Signatures through Approximate Representation by Quadratic Forms. CRYPTO. Plenum Press, New York, 1983, pp. 117–131.

[80] OpenSSH 9.0 was released. https://www.openssh.com/txt/release-9.0. 2022-04. (2024-12-06 閲覧) Streamlined NTRU Prime と X25519 を組み合わせたハイブリッド鍵交換は 8.5 で試験的に実装され，9.0 からはデフォルトで利用される仕様となっている.

[81] J. Park et al. Rydberg-atom experiment for the integer factorization problem. Physical Review Research. Vol. 6, Iss. 2 (2024), p. 023241.

[82] X. Peng, Z. Liao, N. Xu, G. Qin, X. Zhou, D. Suter, J. Du. Quantum Adiabatic Algorithm for Factorization and Its Experimental Implementation. Physical Review Letters. Vol. 101, Iss. 22 (2008), p. 220405.

[83] L. Qiu, M. Alam, A. Ash-Saki, S. Ghosh. Resiliency analysis and improvement of variational quantum factoring in superconducting qubit. ISLPED. ACM, 2020, pp. 229–234.

[84] S. Ragavan, V. Vaikuntanathan. Space-Efficient and Noise-Robust Quantum Factoring. CRYPTO (6). Vol. 14925. Lecture Notes in Computer Science. Springer, 2024, pp. 107–140.

[85] O. Regev. An Efficient Quantum Factoring Algorithm. arXiv: 2308.06572.

[86] R. L. Rivest, A. Shamir, L. M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM. Vol. 21, Num. 2 (1978), pp. 120–126.

[87] Round 2 Additional Signatures. https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures. 2024-10. (2024-12-06 閲覧).

[88] Round 4 Submissions. https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions. 2022-07. (2024-12-06 閲覧).

[89] C. P. Schnorr. Fast Factoring Integers by SVP Algorithms, corrected. Cryptology ePrint Archive, Paper 2021/933. 2021. https://eprint.iacr.org/2021/933.

[90] Federal office for information security. Cryptographic mechanisms: recommendations and key lengths version: 2024-1. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.html. 2024-02. (2024-12-05 閲覧).

[91] P. W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. FOCS. IEEE Computer Society, 1994, pp. 124–134.

[92] P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. Vol. 26, Num. 5 (1997), pp. 1484–1509.

[93] V. V. Sivak et al. Real-time quantum error correction beyond break-even. Nature. Vol. 616, Num. 7955 (2023), pp. 50–55.

[94] U. Skosana, M. Tame. Demonstration of Shor's factoring algorithm for $N = 21$ on IBM quantum processors. Scientific Reports. Vol. 11, Num. 16599 (2021).

[95] M. Sobhani, Y. Chai, T. Hartung, K. Jansen. Variational Quantum Eigensolver Approach to Prime Factorization on IBM's Noisy Intermediate Scale Quantum Computer. arXiv: 2410.01935.

[96] E. G. Johansen and T. Simula. Prime number factorization using a spinor Bose-Einstein condensate-inspired topological quantum computer. Quantum Inf. Process. Vol. 21, Num. 1 (2022), p. 31.

[97] The Cryptographers' Panel. https://www.rsaconference.com/library/presentation/usa/2023/the%20cryptographers%20panel. 2023-04. RSA Conference 2023 (2024-12-05 閲覧).

[98] The Leap quantum cloud service. https://www.dwavesys.com/solutions-and-products/cloud-platform/.

[99] TRAFICOM. Kryptografiset vahvuusvaatimukset luottamuksellisuuden suojaamiseen – kansalliset turvallisuusluokat. https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/file/Kryptografiset_vahvuusvaatimukset_-_kansalliset_turvallisuusluokat_0.pdf. 2024-09. (2024-12-06 閲覧).

[100] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, R. Cleve, I. L. Chuang. Experimental Realization of an Order-Finding Algorithm with an NMR Quantum Computer. Phys. Rev. Lett. Vol. 85, Iss. 25 (2000), pp. 5452–5455.

[101] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature. Vol. 414, Num. 6866 (2001), pp. 883–887.

[102] D.-S. Wang. A comparative study of universal quantum computing models: Toward a physical unification. Quantum Eng. Vol. 3, Num. 4 (2021).

[103] W. Wang, Z. You, S. Wang, Z. Tang, H. Ian. Computing Shor's algorithmic steps with classical light beams. Scientific Reports. Vol. 12, Num. 21157 (2022).

[104] D. Willsch, P. Hanussek, G. Hoever, M. Willsch, F. Jin, H. De Raedt, K. Michielsen. The State of Factoring on Quantum Computers. 2024. arXiv: 2410.14397.

[105] B. Yan et al. Factoring integers with sublinear resources on a superconducting quantum processor. arXiv: 2212.12372.

[106] C. Zalka. Grover's quantum searching algorithm is optimal. Phys. Rev. A. Vol. 60, Iss. 4 (1999), pp. 2746–2751.

[107] J. Zander. Advancing science: Microsoft and Quantinuum demonstrate the most reliable logical qubits on record with an error rate 800x better than physical qubits. https://blogs.microsoft.com/blog/2024/04/03/advancing-science-microsoft-and-quantinuum-demonstrate-the-most-reliable-logical-qubits-on-record-with-an-error-rate-800x-better-than-physical-qubits/. 2024-04. (2024-12-01 閲覧).

[108] デジタル庁, 総務省, 経済産業省. 暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準. CRYPTREC LS-0003-2022r1, https://www.cryptrec.go.jp/list/cryptrec-ls-0003-2022r1.pdf. 2022-03.

[109] デジタル庁, 総務省, 経済産業省. 電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）. CRYPTREC LS-0001-2022r1, https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2022r1.pdf. 2024-05.

[110] 中国密码学会. 全国密码算法设计竞赛通知. https://sfjs.cacrnet.org.cn/site/content/309.html. 2018-06. (2025-01-11 閲覧).

[111] 中国密码学会. 关于全国密码算法设计竞赛算法评选结果的公示. https://sfjs.cacrnet.org.cn/site/content/854.html. 2020-01. (2025-01-11 閲覧).

[112] 宮地 充子. 楕円曲線の理論的及び実用的可能性. IEICE FUNDAMENTALS REVIEW. Vol. 14, Num. 4 (2021), pp. 329–336.

[113] 細山田 光倫. 量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価. CRYPTREC EX-2901-2019, https://www.cryptrec.go.jp/exreport/cryptrec-ex-2901-2019.pdf. 2020-01.

[114] 縫田 光司. 耐量子計算機暗号. 森北出版, 2020.

[115] 伊藤 公平. 量子計算. 2010-02. https://www.ieice-hbkb.org/files/ad_base/view_pdf.html?p=/files/S2/S2gun_05hen_03.pdf. 電子情報通信学会 知識ベース 知識の森 S2 群（ナノ・量子・バイオ）5 編（量子通信と量子計算）3 章.

[116] 国立国会図書館調査及び立法考査局. 量子情報技術：科学技術に関する調査プロジェクト報告書. 2022-03. https://www.ndl.go.jp/jp/diet/publication/document/2022/index.html.

[117] 国立研究開発法人科学技術振興機構. 目標 6 2050 年までに、経済・産業・安全保障を飛躍的に発展させる誤り耐性型汎用量子コンピュータを実現. https://www.jst.go.jp/moonshot/program/goal6/index.html. (2024-12-01 閲覧).

[118] 富士通. 量子コンピュータの誤り耐性量子計算を解説！エラー訂正とエラー緩和の最新トレンドを紐解く. https://activate.fujitsu/ja/key-technologies-article/ta-fault-tolerant-quantum-computation-20240515. 2024-05. (2024-12-01 閲覧).

[119] 高安 敦. Shor のアルゴリズム実装動向調査. CRYPTREC EX-2005-2020, https://www.cryptrec.go.jp/exreport/cryptrec-ex-3005-2020.pdf. 2021-06.

[120] 清藤 武暢, 四方 順司. 量子コンピュータが共通鍵暗号の安全性に与える影響. 金融研究. Vol. 38, Num. 1 (2019), pp. 45–72. https://cir.nii.ac.jp/crid/1523106604811659392.

[121] 理化学研究所. 量子コンピュータを利用できる「量子計算クラウドサービス」開始 ―国産超伝導量子コンピュータ初号機の公開―. https://www.riken.jp/pr/news/2023/20230324_1/. 2023-03. (2024-12-01 閲覧).

[122] 大関 真之. 量子アニーリングが拓く機械学習と計算技術の新時代 (量子システム推定の数理). 数理解析研究所講究録. Vol. 2059 (2017), pp. 13–23. https://cir.nii.ac.jp/crid/1050564288163922560.

[123] 文部科学省 科学技術・学術政策研究所科学技術予測センター. 第 11 回科学技術予測調査 デルファイ調査. https://nistep.repo.nii.ac.jp/?action=repository_uri&item_id=6692&file_id=13&file_no=3. 2020-06. (2024-12-05 閲覧).

[124] 山口 純平, 伊豆 哲也. イジング計算を用いた暗号解析について. オペレーションズ・リサーチ：経営の科学. Vol. 67, Num. 6 (2022), pp. 290–296. https://cir.nii.ac.jp/crid/1520011030559130112.

[125] 山口 純平, 伊豆 哲也, 國廣 昇. 素因数分解問題に対する Shor アルゴリズムの実装と量子計算機シミュレータを用いた実験. 暗号と情報セキュリティシンポジウム (SCIS 2023). 2023-01, 4A2–3.

[126] 大塩 耕平. アナログ量子シミュレータの開発動向と応用. https://www.mizuho-rt.co.jp/publication/others/pdf/mhrt003_01.pdf. 2024-03. (2024-12-06 閲覧).

[127] 大阪大学 量子情報・量子生命研究センター. 【プレスリリース】大阪大学に設置した超伝導量子コンピュータ国産３号機の クラウドサービスを開始. https://qiqb.osaka-u.ac.jp/20231220pr/. 2023-12. (2024-12-01 閲覧).

[128] 量子耐性暗号研究団. KpqC. https://kpqc.or.kr/. (2024-12-06 閲覧).

[129] 満保 雅浩. 公開鍵暗号. 映像情報メディア学会誌. Vol. 69, Num. 9 (2015), pp. 714–720.

第 2 章

PQC の活用方法

将来，一定以上の能力を持つ量子コンピュータが登場した場合には，既存の公開鍵暗号が解読される（破られる）という脅威が指摘されている [1, 20]。本章では，現在，標準的に用いられている公開鍵暗号の解読が可能となる水準の量子コンピュータを Cryptographically Relevant Quantum Computer（CRQC）と記載し，CRQC を用いた攻撃に対しても安全な性質を「耐量子計算機性」と記載する。また，耐量子計算機暗号（Post-Quantum Cryptography: PQC）とは，耐量子計算機性を持つ暗号アルゴリズムを意味し，本稿の対象である公開鍵暗号アルゴリズム以外にも，共通鍵暗号やハッシュ関数も含まれるものとする [1]。加えて，「耐量子計算機性を持つ情報システム」とは，CRQC を用いた攻撃に対しても安全な情報システムを示すものとする。

ある情報システムが，既存の公開鍵暗号を利用していた場合，その情報システムは，将来における CRQC を用いた攻撃の脅威に晒されることになる。そのような脅威への対応方法としては，情報システム内の（耐量子計算機性を持たない）既存の公開鍵暗号方式部分を，耐量子計算機性を持つ公開鍵暗号方式に置き換えることで，その情報システムに耐量子計算機性を持たせることが考えられる。

なお，耐量子計算機性を持たせるためには異なるアプローチも考えられる。例えば，今まで公開鍵暗号を利用していた情報システムを，公開鍵暗号を利用しない仕組みに置き換えるアプローチである。具体的には，信頼できる特使等の別の情報共有手段を利用し，通信相手と共通鍵の事前共有を行う方法である。しかし，このアプローチでは，情報システムの「スケーラビリティ」[*1]が損なわれることが予想され，場合によっては実現不可能なコストが発生する。

現在，普及している情報システムの中には，公開鍵暗号を利用することにより，そのサービスのスケーラビリティを維持しているものも多い。インターネットはその代表例であり，通信相手を認証する用途等で公開鍵暗号を利用することにより，大規模な通信ネットワークの構築及び維持を実現している [4, 11, 14, 18]。このような大規模情報システムにおいて，仮に，耐量子計算機性を持たせるために公開鍵暗号の利用を取りやめた場合，スケーラビリティが損なわれ，その結果，維持・運用コストが大きく上昇してシステムの維持も困難となる。このため，公開鍵暗号を利用した情報システムの現在及び将来においてスケーラビリティ上の懸念が発生しないという見通しがない限り，耐量子計算機性の実現のためのアプローチとしては，耐量子計算機性を持つ公開鍵暗号を利用することが望ましい。

以下では，より具体的に，耐量子計算機性を持たせるためのアプローチについて紹介する。公開鍵暗号によって暗号化（守秘・鍵共有）を行う情報システムに対して，耐量子計算機性を持たせるアプローチには，表 2.1 に示す手法及びその組み合わせが存在するが，一般に下段のアプローチになるほどスケーラビリティが低下する。ここで，最もスケー

[*1] スケーラビリティとは，要求される処理量等の変化に応じてそのシステムの対処能力を柔軟に増減させることができる能力である。
https://www.gartner.com/en/information-technology/glossary/scalability
本章では，情報システムの規模（ステークホルダ数，利用者数，処理量等）が増減した場合でも，その情報システムが消費するリソース（計算量，通信量，人の手間等）が極端に増加しない，又は，減少させることができる能力の意味で利用する。

表 2.1: 公開鍵暗号による暗号化（守秘・鍵共有）を行う情報システムに対して耐量子計算機性を持たせるためのアプローチ

	アプローチ	概要
1.	削除・匿名化	情報システムが，漏洩しても問題ない情報以外は保管しない/扱わないようにする。又は，保管する/扱う情報を加工することによって，漏洩しても問題ないように変形する。この方式は，スケーラビリティが最も高いが，可用性が大きく低下することが考えられ，選択できないことも多い。
2.	耐量子計算機性を持つ公開鍵暗号の採用	最も一般的な解決策であり，スケーラビリティを確保できる。現代暗号の利点を維持するアプローチである。
3.	公開鍵暗号を用いない鍵共有手段の導入	公開鍵暗号を利用している情報システムを，公開鍵暗号を利用しない仕組み（例えば，物理的に通信相手全員に IC カードを配布することで，共通鍵の事前共有を行うなど）に置き換えることで，耐量子計算機性を持たせる。暗号技術の観点からは，公開鍵暗号が登場する以前の思想で再設計することになる。スケーラビリティが低く，不特定多数が利用するシステムでは採用が困難と考えられる。また，通信当事者の捕捉が容易となることも考えられ，匿名性の確保やプライバシ保護に関する再設計も併せてが必要になる可能性がある。
4.	物理アクセス制御	上記 1〜3 のアプローチが採用できない場合にも採用可能である。暗号技術の観点からは，暗号技術が発展する以前の思想で再設計することになる。実装コスト及び運用コストが非常に高くなることが予想される。

ラビリティが期待できるデータ削除や匿名化といった手法は，そのデータが削除や匿名化が可能であるか否かを検討した後に実施する必要があり，運用上のスケーラビリティは高いものの，導入前の検討のために時間を必要とし，情報システムの可用性が低下するおそれもある。また，法令やポリシー等で削除・匿名化が許容されていない場合には，実施できないおそれもある。

これらの事情より，耐量子計算機性を持たせるための最も汎用的かつ根本的な対応は，既存の公開鍵暗号方式を耐量子計算機性を持つ公開鍵暗号方式に置き換えることであると考えられる。

ただし，情報システムで利用されている公開鍵暗号方式を，耐量子計算機性を持つ公開鍵暗号方式に置き換えることは容易ではない。公開鍵暗号方式を，耐量子計算機性を持つものへとすることは，実装をシンプルに切り替えただけでは完了せず，公開鍵暗号がどのように利用されているのかについて認識した上で，運用やデータ管理に係る様々な処理も併せて実施することが要求される（以降，暗号方式の置き換えに加えて，これらの処理を行うことを「暗号移行」と呼ぶ)。そこで本章では，公開鍵暗号のいくつかの利用形態を念頭に，耐量子計算機性を持つ公開鍵暗号方式への暗号移行について紹介する。まず，現行の公開鍵暗号の利用形態を紹介した上で，各利用形態における CRQC による脅威及びその対策について，システム運用やデータ管理処理の観点を踏まえて概説する。また，脅威を評価する上で重要となる，保護対象となるデータの保護期間について記載した上で，利用形態や保護対象を踏まえた対応についても概説する。

2.1 公開鍵暗号の利用形態

既存の公開鍵暗号方式を，耐量子計算機性を持つものへと暗号移行するに際しては，その公開鍵暗号方式の利用形態ごとに，暗号移行のプロセスが大きく異なることが予想される。そこで，本節で公開鍵暗号の利用形態について概説した上で，次節以降で各利用形態における暗号移行のプロセスについて述べる。公開鍵暗号にはいくつかの利用形態が存在するが，本章では「電子政府における調達のために参照すべき暗号のリスト」[24] （以下「CRYPTREC 暗号リスト」と呼ぶ。）に合わせて，公開鍵暗号を署名・守秘・鍵共有に分類し，以降その分類に沿って概説する。また，本節では，署名用途／守秘用途／鍵共有用途の耐量子計算機性を持つ公開鍵暗号方式を，それぞれ署名用途／守秘用途／鍵共有用途の PQC と表記する。

2.1.1 署名用途での公開鍵暗号の利用

本節では，署名を付与する行為を「デジタル署名処理」と呼び，付与される署名データを「デジタル署名」と呼ぶ。デジタル署名が付与されたコンテンツを改竄すると，その改竄を検知することができる。このため，署名用途の公開鍵暗号を用い，コンテンツにデジタル署名を付与することで，コンテンツの改竄によりもたらされる被害を防止することができる。コンテンツは，人が読む文章（ドキュメントデータ），動画等の情報であることもあれば，暗号鍵の鍵情報[*2]であることもある。また，デジタル署名処理に用いられる秘密鍵が，対応する公開鍵を含む電子証明書によって所定の人物／組織／装置等と紐づいている場合では，コンテンツの生成人物／組織／装置を確認（認証）することもできる。このように署名用途の公開鍵暗号は，コンテンツの改竄防止，署名者の認証，データ元の認証等に利用される。

具体的な署名用途の公開鍵暗号の利用例としては，TLS 通信 [18] におけるクライアント認証（利用者の認証）やサーバ認証（サービス提供者の認証），OS のコードサイン（バイナリデータが改竄されていないことの確認）等に広く利用されている。また，公開鍵の配布手段の一種である公開鍵暗号基盤（PKI）の構成においても，公開鍵暗号は広く利用されており [4]，コンテンツに対して署名が付与された時刻を確認可能なタイムスタンプ署名方式 [23] 等も存在する。CRYPTREC 暗号リストには，DSA，ECDSA，EdDSA，RSA-PSS，及び RSASSA-PKCS1-v1_5 が署名用途の公開鍵暗号として記載されている。

2.1.2 守秘用途での公開鍵暗号の利用

守秘用途の公開鍵暗号によって暗号化された暗号文は，対応する秘密鍵なしに復号することは困難となる。このため，守秘用途の公開鍵暗号は，意図した相手だけにデータを提示するために利用することができる。暗号化処理による保護は，ドキュメントデータ，動画等の情報に対して行われることもあれば，暗号鍵の鍵情報[*3]に対して行われることもある。保護が鍵情報に対して行われるユースケースとしては，鍵情報を通信当事者間で共有する場合や，暗号鍵所有者がその鍵情報をバックアップする場合等が該当する。

守秘用途及び鍵共有用途の公開鍵暗号の一般的な実装形態として，公開鍵暗号方式により別の暗号鍵を保護し，その暗号鍵を利用した共通鍵暗号方式によりコンテンツの秘匿性や完全性を保護するというアプローチが存在する。このアプローチでは，共通鍵暗号方式の暗号鍵（以下，単に共通鍵と呼ぶ）は送信者により作成され，配送される。したがって，ある時点で共通鍵が漏洩した場合には，過去にその秘密鍵を持つ利用者に対して配送された共通鍵が漏洩するお

[*2] 鍵情報には暗号鍵やメタデータが含まれ [5]，公開鍵暗号の鍵のみではなく共通鍵暗号の鍵に関する情報も含む概念となる。
[*3] 秘密鍵，共通鍵，鍵導出鍵及びそれらの鍵のメタデータを含む。

それがある。また，受信者は共通鍵の生成に関わることがないため，送信者が別の通信相手と共通鍵を使い回していても察知することができない。このため，昨今の TLS 通信等における共通鍵の共有においては，守秘用途の公開鍵暗号でなく，次節で概説する鍵共有用途での公開鍵暗号を一時的な鍵と組み合わせて利用することが望ましいと考えられている [5, 19]。なお，「TLS 暗号設定ガイドライン」[5] においても，鍵交換（鍵共有・守秘）においては，Perfect Forward Security（PFS）[*4]の特性を持つ DHE（又は ECDHE）を選択することがセキュリティ上望ましいと記載されている。CRYPTREC 暗号リストには，RSA-OAEP 及び RSAES-PKCS1-v1_5[*5]が守秘用途の公開鍵暗号として記載されている。また，RFC7525 においても [19]，4.1 節において守秘用途で使用される RSA 暗号方式による鍵の転送（RSA key transport）は利用すべきでないと記載されており，4.2 節において一時的（Ephemeral）な鍵を用いる暗号スイート[*6]が推奨されている。

2.1.3 鍵共有用途での公開鍵暗号の利用

鍵共有用途での公開鍵暗号は，鍵共有に参加する二者が，同一の鍵情報[*7]を共有するために使用される。近年利用されている二者間鍵共有を目的とした多くの公開鍵暗号プロトコルにおいては，鍵共有に参加する双方が何らかの値を生成し，その値に対して秘密鍵を使用した計算を行う。結果として，共有される鍵には双方の生成した値が影響することとなり，一方のみの計算で暗号鍵を導出することはできない。このため，守秘用途でのデータ送付と異なり，送信者があらかじめ意図した特定の鍵を，共有鍵として利用することはできない。CRYPTREC 暗号リストには，DH，ECDH，及び PSEC-KEM が鍵共有用途の公開鍵暗号として記載されている。

2.2 PQC の導入における課題

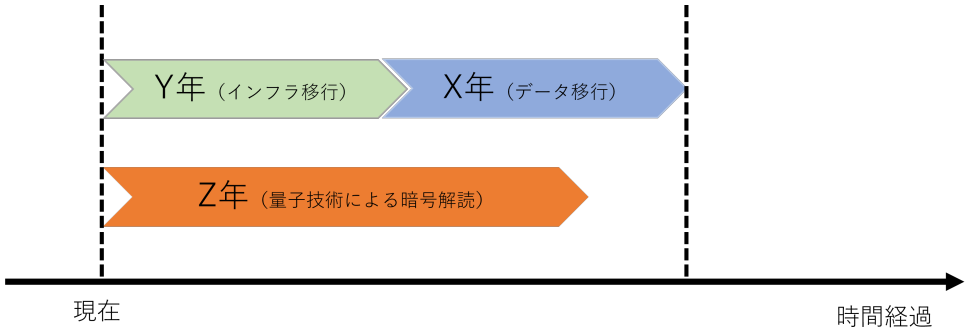


図 2.1: Mosca の発表 [15] より

現在広く利用されている公開鍵暗号が，量子コンピュータを利用した攻撃に起因して，“近い将来”に危殆化する可能性は低い [7] と考えられている。他方で，Michel Mosca [15] が指摘するように，その情報システムで生成されるデータに対して暗号方式による保護が期待される期間（図 2.1 における X）に，暗号処理の実装の置き換えに要する期間（同図における Y）を加えたものが，CRQC による攻撃が実現するまでの期間（同図における Z）よりも長い場合（X + Y > Z の場合）は，当該情報システムで生成されるデータは CRQC による攻撃の脅威にさらされることになる。

[*4] ある時点における鍵が漏洩した場合でも，漏洩した鍵とは異なる鍵を使用していた過去の暗号文の復号はできない性質。
[*5] 守秘用途の RSAES-PKCS1-v1_5 は，運用監視暗号リストに記載されており，互換性維持以外での利用は推奨されていない。
[*6] 複数の暗号アルゴリズムの組合せ
[*7] 共通鍵暗号の共通鍵，鍵導出機能の鍵やパラメータ等

すなわち，CRQC 実現までの期間（Z）が非常に長く，遠い将来であったとしても，その情報システムの X や Y の値が大きければ，何らかの対応が求められる。なお本章において，特記しない限り以降では，X，Y，Z は図 2.1 における X，Y，Z を示す。

もっとも，CRQC の実現時期は未だ不透明であり，Z を予想することは困難である。また，X は，暗号方式のみならず，保護対象となるデータの性質等によっても大きく異なる。特に，保護対象となるデータに対して，保護期間が設定されていない場合などは，X を導出すること自体が新たな課題となる。同様に，Y も，暗号方式の実装形態によって大きく変化する。さらに，X 及び Y は，情報システムの運用を通して，将来において変動することもありうる。

このように，ある公開鍵暗号アプリケーションが利用されている際に，CRQC による脅威について備える必要性があるか否かを判断しようとした場合，Z は不確定であり，X や Y も変動しうるため，判断が難しいという課題がある [25]。ここで，保護対象となるデータに保護期間が設定されていない場合においては，判断に先駆けて（X 導出のために）データの保護期間を決定することとなり，場合によってはその判断を行うための情報収集に相当の期間を必要とする。

PQC の導入においては，その情報システムに耐量子計算機性を持たせることが必要なのか，また，いつまでにそれを行う必要があるのか，を判断すること自体が課題となる。

2.2.1 署名用途での課題

署名用途の公開鍵暗号は，コンテンツの改竄防止，認証等に利用されるが，ユースケースによって脅威の性質は大きく異なる。例えば，TLS 通信 [18] におけるクライアント認証やサーバ認証においては，認証用に付与されたデジタル署名の検証を行うのは基本的にその場限りとなるため，X の値は小さくなる。また，Web ブラウザが信用するサーバ認証用の証明書の有効期間は，ごく一部の例外を除いて 1 年程度であり，それほど長い期間利用されることはない。そのため，X の値は，守秘用途や他の認証用途に比べて非常に小さくなる [25]。さらに，ブラウザのアップデートやルート認証局の入れ替えを，より迅速に実施できる体制を整備しており，Y の値も守秘用途や他の認証用途に比べて小さい。

他方で，電子データに対するドキュメント署名や，バイナリデータに対するコードサインであれば，署名対象のデータを利用する人が存在する限り（数十年に渡り）デジタル署名が検証されることもある。特に，コードサインにおいては，仮に電子証明書に有効期間が記載されていたとしても，その有効期間満了後にも検証されることが十分に考えられる。そのため，X の値は，守秘用途や他の認証用途に比べて非常に大きくなる。

このように，署名用途においては，X の値は大きく異なりうるものであり，個々のアプリケーションごとに判断する必要がある。また，公開鍵の配布のために PKI を利用した場合，トラストアンカーの置き換え等に時間を要するため，Y が 10 年以上となることも珍しくない。

2.2.2 守秘用途での課題

守秘用途の公開鍵暗号においては，攻撃者が事前に暗号技術で保護されたデータを収集して保存しておき，後からそのデータに対して攻撃を行う攻撃である，Harvest Now Decrypt Later 攻撃（以下，「ハーベスト攻撃」と呼ぶ）[*8] の脅威が指摘されている。

ハーベスト攻撃においては，保護対象となるデータの保護期間，すなわち X の値が大きくなるほど，攻撃者が攻撃可能な期間が長くなる。これは，攻撃者が CRQC の開発を待たずに攻撃（保護された情報の収集）を開始できるためである。一方，防御側は，攻撃者に情報が収集される前に，情報システムに耐量子計算機性を付与することが求められ

[*8] Record Now Decrypt Later 攻撃，Store Now Decrypt Later 攻撃等とも呼ばれる。

る。保護対象となる情報の保護期間が長くなるほど，この不均衡は大きくなり，攻撃者の攻撃可能期間が長くなる。

守秘用途の公開鍵暗号では，保護対象となるコンテンツや鍵情報の保護期間が非常に長期となることが想定されている場合や，無期限で保護することが想定されている場合も存在する。例えば，患者を特定又は推測可能な形態で保管された遺伝性疾患に関する医療情報や，外交関係の機微な情報，さらには，それらの情報の暗号化に利用される鍵などは長い保護期間を持つ傾向にある。また，ドキュメントの生成時において，無期限に守秘することを前提としており，公開することを想定していない情報も存在する。

これらの情報においては，X の値は非常に大きくなるため，おそらく X + Y > Z が成立することになる。そのため，速やかに CRQC の脅威に対する何らかの対応を行うことで，被害を軽減することが望ましい [25]。

2.2.3 鍵共有用途での課題

鍵共有用途での課題は，守秘用途における課題と同種の課題を含んでいる。例えば，鍵共有で共有された共通鍵が，非常に長い保管期間を持つデータの暗号化に利用されていた場合，X の値は非常に大きくなり，X + Y > Z が成立すると考えられ，速やかに CRQC の脅威に対する何らかの対策が必要となる。

さらに，守秘用途では存在しない新たな懸念も存在する。例えば，一時的（Ephemeral）な鍵情報を用いた DH 鍵共有方式を採用することにより PFS を達成している情報システムが存在し，その情報システムは，PFS であることを前提とした運用ポリシーを策定していたとする。この情報システムの DH 鍵共有処理部分を，耐量子計算機性を持つ標準化された公開鍵暗号方式に置き換える場合，以下の 2 つの方針が考えられる。

1）鍵共有用途の PQC に置き換える
2）守秘用途の PQC に置き換える

標準化された鍵共有用途の PQC が存在するのであれば，1）が選択可能であり，比較的容易に実現可能だと考えられる。しかし，そのような鍵共有用途の PQC が存在せず，守秘用途の PQC しか標準化されていない場合には，2）を選択することとなり，守秘用途の PQC を用いて鍵共有部分を構成することとなる。

2）の選択において，守秘用途の PQC を単純に導入した場合，PFS の性質を持たなくなるおそれがあり，それによりデータ保護及び運用ポリシー策定時に想定していなかった経路からの情報漏洩等が発生する懸念が生じる。

他方で，2）の選択において，既存の鍵交換及び守秘用途の PQC の両方のハイブリッド構成を用いることによって対応するアプローチも存在する[*9]。ハイブリッド構成を用いることで，既存のアルゴリズムでしか防げない攻撃に対しても，新たなアルゴリズムでしか防げない攻撃に対しても，安全な構成とすることができる [22]。

もっとも，鍵共有処理を複数回行うことに起因し，処理量及びデータ転送量が増加するため，その増加に対応できるように情報システムや通信プロトコルの修正が必要となりうることには注意が必要である。

2.3 PQC 導入へのアプローチ

2.2 節でも記載したように，CRQC の実現時期（又は実現までの期間 Z）は不透明ながら，X や Y の値が大きな情報システムにおいては，何らかの対応を取ることが望ましい。また，本章冒頭で記載したように，情報システムに耐量子計算機性を持たせる手段は，耐量子計算機性を持つ公開鍵暗号方式の導入だけではないものの，スケーラビリティを考慮すると耐量子計算機性を持つ公開鍵暗号方式の利用が有望である。本節では，耐量子計算機性を持つ公開鍵暗号方式への暗号移行を念頭に，その暗号移行を円滑に行う上での考慮事項について概説する。

[*9] TLS における [13, 22]，CMS における [16] 等が当該アプローチとして挙げられる。

2.3.1 プライオリティ設定の重要性

公開鍵暗号は様々な用途において普及している。それらの全ての公開鍵暗号方式を耐量子計算機性を持つものへ暗号移行するためには，長い期間及び労力を要する。また，情報システムの中には，そのシステムの利用期間及び生成されるデータの保護期間が短い等の理由により，耐量子計算機性を持たせる必要がないものも存在するかもしれない。

そこで，暗号移行を検討する上では，X,Y,Z を意識して対応することが重要と考えられる。もっとも，X や Y は暗号方式の利用局面ごとに異なることも想定され，またそれらの値は将来において変動する可能性がある。さらに，Z は不確定であり，予想すること自体も困難である。このような状況の下で，全ての暗号モジュールに対して X,Y,Z を分析するアプローチを取ることは，作業量の観点で大きな困難が伴うことが予想され，結果として本当に保護が必要なデータに対する対応に手が回らないおそれがある。そこで，暗号移行を行う担当者は，優先度の高いものを洗い出し，その優先度に応じて対応を行うことが適切である [9, 26, 25]。

PQC への暗号移行を検討するにあたり，あらかじめ優先順位付けを行うことの重要性は，金融庁の報告書 [27] でも触れられており，基本事項は以下のように整理されている。

- 暗号解読可能な量子コンピュータによる既存の暗号危殆化に関連するリスクに基づいて，移行対象の優先順位付けを行う。
- 移行対象の詳細な把握のため，クリプト・インベントリを構築する。
- 暗号危殆化状況に応じて安全かつ迅速に対応できるアーキテクチャを検討する。
- 優先順位の高いものを中心に移行期限を設定し，期限超過の可能性も踏まえたリスク低減策も検討する。

ここで，クリプト・インベントリとは利用している暗号モジュールや暗号方式のリストのことであり，その作成においては，既に管理簿や仕様書等が存在する場合はそれを利用することができる。また，管理簿や仕様書等が存在しない場合は，何らかの自動化ツールを使うことが，効率の観点からもミスを減らす面からも望ましい。そのような自動化ツールの利用を検討する上では，NIST NCCoE の検討 [9] が参考になる。

CRQC による攻撃リスクの評価においては，CRQC による攻撃が成功した場合の影響，暗号方式によって保護される情報の保護期間（X の把握のために必要），情報システムで利用する各暗号モジュールの移行に要する時間（Y），CRQC を利用する攻撃を行うための前提条件の難易度（攻撃対象である暗号化データ取得の難易度や，そのデータを利用した攻撃の難易度）等の把握が有用である。

この優先順位付けに先駆けて，過剰な保護期間が設定されている情報の保管期間短縮，不要な情報の消去，公開可能な情報の公開等を併せて実施する事も望ましい。このような処理により，X の短縮が期待でき，暗号移行の対象となるシステムを削減する効果が期待される。

暗号移行に際しては，速やかに PQC に暗号移行するというアプローチと，あらかじめクリプトグラフィック・アジリティ [12][*10]を向上させつつ，ある程度以上のクリプトグラフィック・アジリティを達成した上で暗号移行するというアプローチが存在する。

クリプトグラフィック・アジリティが向上すると，Y や X の値が小さくなる。このため，例えば，PQC の評価が十分にされておらず，暗号移行開始の妨げとなっている期間においては，当面の間はクリプトグラフィック・アジリティ向上に努めるというアプローチも一定の合理性があるものと考えられる [26]。

[*10] 暗号方式を変更可能とする性質。2.3.2 節参照。

2.3.2 クリプトグラフィック・アジリティの重要性

クリプトグラフィック・アジリティは，文脈によって捕捉範囲が異なり，それに伴って異なる意味合いを持つことがある [2]。しかしながら，それらに通底しているのは，暗号アルゴリズムや暗号プロトコルをより迅速に変更できる性質が挙げられる。

暗号移行においては，暗号移行の対象となる情報システムの暗号部分が，情報システムにハードコードされている場合には，暗号アルゴリズムの変更が困難である。このような状態は「クリプトグラフィック・アジリティを持たない」と表現することができる。

他方で，標準プロトコルを採用する情報システム，暗号モジュールにも標準プロトコルを利用している情報システム，その API が適切に定義されている情報システム，相互運用性が確保されている情報システム，及び暗号回路を含むファームウェアアップデートをオンラインで実施できるように設計されている情報システム等では，その暗号移行に要する時間は比較的短くなり，$X + Y > Z$ となる可能性も低くなる。X 及び Y の値が十分に低く，所定の目標期間以内に暗号移行が可能なシステムは，「クリプトグラフィック・アジリティを持つ」と表現することができる [2, 25]。

クリプトグラフィック・アジリティを持たせるための対応は，PQC の実装とは独立して実施することが可能である [3]。また，より短い期間での暗号移行を行うことが可能となれば，移行プロセスを開始するまでの猶予期間（$Z-X-Y$）をより長くすることが期待される。

PQC への暗号移行を実施するにあたっては，まずは暗号移行を長期化する要素を排除することを試み，情報システムにおける暗号プロトコルの変更をより迅速にできるようにシフトさせていく対策，すなわちクリプトグラフィック・アジリティを確保する対策を併せて実施することが効果的である [2, 3, 25]。

2.3.3 既存暗号方式とのハイブリッド構成

暗号移行においては，ハイブリッド構成を採用することができる。PQC への暗号移行の文脈におけるハイブリッド構成とは，既存の公開鍵暗号と，PQC の両方を利用することによって何らかの目標の達成を目指すものであるが，厳密な定義は見当たらない [8]。ハイブリッド構成の目標は，暗号アルゴリズムの切替期間中における相互運用性の確保や，既存暗号方式しか利用できない機器に対する後方互換性の確保であることもあれば，両方のアルゴリズムのうち片方が危殆化した場合の安全性の維持であることもある。

また，ハイブリッドという用語は，単一の暗号モジュールを構成するコンポジット方式 [13, 16] の文脈で使用されることもあれば，複数の暗号モジュールの出力を入力として受け取り，新たな出力を生成するコンバイナー構造に対して使用されることもある [8]。

なお，IETF の標準化活動において，ハイブリッド構成による鍵共有方式に関しては一定の合意が見受けられるが [13, 16]，ハイブリッド構成によるデジタル署名方式 [17] に関しては合意に時間を要している。

2.3.4 署名用途固有の対策

署名用途の公開鍵暗号は様々なユースケースで利用されるが，PKI 等のインフラの移行に要する時間（Y）やコードサイン証明書が利用される期間（X）が比較的長いことから，速やかな PQC への暗号移行が困難である。この場合においても，以下の対応を取ることが望ましい。

PKI においては，一般に Y が長くなる傾向にあるが，電子証明書の有効期間の短縮や，1 枚の電子証明書に対して（既存暗号方式と署名用途の PQC の）2 つの公開鍵及びデジタル署名を付与する方式などを採用することで，Y の短縮

が期待できる [21]。なお，後者の 2 つの公開鍵暗及びデジタル署名を利用する方式においては，実装やポリシー管理の複雑さが大きく増加することから，注意を必要とする。

X を実質的に短縮する技術として，タイムスタンプ更新技術が存在する。例えば，ERS[10] 等を利用することで，タイムスタンプの更新や，暗号方式の更新が可能となる。Z が経過する前に，既存の公開鍵暗号を PQC に更新することが可能であれば，X,Y,Z の関係によらず，データは保護される。ただし，このアプローチでは，データ構造の複雑さが増加する傾向があり，（PQC への即時の暗号移行に比べては小さいものの）情報システムの運用費用が増加する点には注意を必要とする。

2.3.5 守秘及び鍵共有用途固有の対策

既に述べたように，耐量子計算機性を持たせるための一般的な対策は，既存の暗号方式を耐量子計算機性を持つ公開鍵暗号方式に移行することである。ここで，2.2.2 節及び 2.2.3 節で述べたとおり，守秘及び鍵共有用途で保護されたコンテンツや鍵情報は，保護期間が非常に長いことや，場合によっては無期限で保護されることも考えられる。このような情報に対するハーベスト攻撃の脅威を考慮すると，当該情報は，将来における CRQC による解読リスクに既に晒されていることから，一刻も早く耐量子計算機性を持たせる対応を始めることが望ましい。ただし，全ての守秘及び鍵共有用途の公開鍵暗号を移行するためには非常に大きなリソースが要求され，現実的なコストでは実現が困難であるおそれがある。

このような状況においても，守秘及び鍵共有用途固有の対策を効率的に行う方法 [26] として，以下のアプローチがある。

Z に対して X + Y の値が非常に小さく，X + Y $\ll$ Z と予測される暗号文に対しては，CRYPTREC による注意喚起情報 [6] に注意を払いつつ，現在用いている暗号の使用を継続する。また，X + Y $>$ Z となることが十分予想される暗号文に対しては，2.3 節前段で述べた，PQC への暗号移行や，暗号文の保護期間である X の短縮，情報システムの暗号処理の実装の置き換えに要する期間 Y の短縮を行う。その結果，X や Y の値を十分に小さくすることができるのであれば，現在用いている暗号方式の使用を継続する。

一方で，X + Y $>$ Z と予想される，又は，X + Y $>$ Z となることが避けられない暗号文に対してしては，暗号システムの PQC への暗号移行を進めつつも，既存の公開鍵暗号によって保護されている暗号文は公開ネットワーク等に保管せず，適切にアクセスコントロールを行う。

なお，現在 DH を利用している場合は，2.2.3 節で述べたような検討を行い，DH 固有の性質が必要か否かをあらかじめ検討することが望ましい。

2.4 PQC の活用にむけて

PQC への暗号移行においては，どのようなデータに対して，どのような暗号技術を利用しているのかを把握することが第一歩となる。また，保護対象となるデータの保護期間等をあらかじめ把握しておくことで，より効率的な対応ができる [26]。その上で，公開できるデータは公開し，破棄可能なデータは破棄することも検討すべきである。この検討を進めることで，クリプトグラフィック・アジリティ [12] の確保も見込まれ，より効果的な PQC への移行が期待できる。CRQC の脅威への対策を検討するにあたっては，保護されている情報の価値，CRQC による攻撃が成功した場合の影響，図 2.1 における X,Y,Z の関係等を踏まえ，プライオリティを付けて，そのプライオリティ順に対策を実施することが望ましい [27, 25]。

第 2 章の参照文献

[1] National Security Agency. The Commercial National Security Algorithm Suite 2.0 and Quantum Computing FAQ. 2024-04. https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/1/CSI_CNSA_2.0_FAQ_.PDF. (2025-01-06 閲覧).

[2] N. Alnahawi, N. Schmitt, A. Wiesmaier, A. Heinemann, T. Grasmeyer. On the State of Crypto-Agility. Cryptology ePrint Archive, Paper 2023/487. 2023. https://eprint.iacr.org/2023/487.

[3] A. Amadori et al. The PQC Migration Handbook. https://publications.tno.nl/publication/34643386/fXcPVHsX/TNO-2024-pqc-en.pdf. 2024-12. (2025-01-06 閲覧).

[4] S. Boeyen, S. Santesson, T. Polk, R. Housley, S. Farrell, D. Cooper. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, https://www.rfc-editor.org/info/rfc5280. 2008-05. (2023-04-12 閲覧).

[5] CRYPTREC. TLS 暗号設定ガイドライン. CRYPTREC GL-3001-3.0.1, https://www.cryptrec.go.jp/report/cryptrec-gl-3001-3.0.1.pdf. 2020-07.

[6] CRYPTREC. 注意喚起一覧. https://www.cryptrec.go.jp/er.html. (2024-03-05 閲覧).

[7] CRYPTREC 暗号技術評価委員会. 注意喚起情報 “現在の量子コンピュータによる暗号技術の安全性への影響”. https://www.cryptrec.go.jp/topics/cryptrec-er-0001-2019.html.

[8] F. Driscoll, M. Parsons, B. Hale. Terminology for Post-Quantum Traditional Hybrid Schemes. Internet-Draft. 2024-12. https://datatracker.ietf.org/doc/draft-ietf-pquip-pqt-hybrid-terminology/05/. (2025-02-20 閲覧).

[9] NIST National Cyersecurity Center of Excellence. Migration to Post-Quantum Cryptography Quantum Readiness: Cryptographic Discovery. NIST SP 1800-38B (initial preliminary draft), https://www.nccoe.nist.gov/sites/default/files/2023-12/pqc-migration-nist-sp-1800-38b-preliminary-draft.pdf. 2023-12. (2025-02-17 閲覧).

[10] T. Gondrom, R. Brandner, U. Pordesch. Evidence Record Syntax (ERS). RFC 4998, https://www.rfc-editor.org/info/rfc4998. 2007-08. (2023-04-12 閲覧).

[11] P. E. Hoffman. DNS Security Extensions (DNSSEC). RFC 9364, https://www.rfc-editor.org/info/rfc9364. 2023-02.

[12] R. Housley. Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms. RFC 7696, https://www.rfc-editor.org/info/rfc7696. 2015-11. (2023-04-12 閲覧).

[13] K. Kwiatkowski, P. Kampanakis, B. Westerbaan, D. Stebila. Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLSv1.3. Internet-Draft. 2024-12. https://datatracker.ietf.org/doc/draft-kwiatkowski-tls-ecdhe-mlkem/03/. (2025-02-20 閲覧).

[14] M. Lepinski, S. Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480, https://www.rfc-editor.org/info/rfc6480. 2012-02. (2025-01-15 閲覧).

[15] M. Mosca. Cybersecurity in a quantum world: will we be ready? Workshop on Cybersecurity in a Post-Quantum World. Session 8. 2015-04. (2024-02-29 閲覧).

[16] M. Ounsworth, J. Gray. Composite KEM For Use In Internet PKI. Internet-Draft. 2024-10. https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-kem/. (2025-01-06 閲覧).

[17] M. Ounsworth, J. Gray, M. Pala, J. Klaußner, S. Fluhrer. Composite ML-DSA For use in X.509 Public Key Infrastructure and CMS. Internet-Draft. 2024-10. https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-sigs/03/. (2025-01-15 閲覧).

[18] E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, https://www.rfc-editor.org/info/rfc8446. 2018-08. (2023-04-12 閲覧).

[19] Y. Sheffer, R. Holz, P. Saint-Andre. Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). RFC 7525, https://www.rfc-editor.org/info/rfc7525. 2015-05. (2023-04-12 閲覧).

[20] P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. Vol. 26, Num. 5 (1997), pp. 1484–1509.

[21] D. Stebila, S. Fluhrer, S. Gueron. Hybrid key exchange in TLS 1.3. Internet-Draft. 2024-10. https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/11/. (2025-02-20 閲覧).

[22] D. Stebila, S. Fluhrer, S. Gueron. Hybrid key exchange in TLS 1.3. Internet-Draft. 2025-01. https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/12/. (2025-02-20 閲覧).

[23] R. Zuccherato, P. Cain, Dr. C. Adams, D. Pinkas. Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161, https://www.rfc-editor.org/info/rfc3161. 2001-08. (2023-04-12 閲覧).

[24] デジタル庁, 総務省, 経済産業省. 電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）. CRYPTREC LS-0001-2022r1, https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2022r1.pdf. 2024-05.

[25] 伊藤 忠彦. 耐量子計算機暗号への移行へ向けた課題と社会実装への論点整理. 電子情報通信学会誌. Vol. 106, Num. 11 (2023), pp. 1026–1030.

[26] 伊藤 忠彦, 宇根 正志, 清藤 武暢. 量子コンピュータによる脅威を見据えた暗号の移行対応. 2019-08. https://www.imes.boj.or.jp/research/papers/japanese/19-J-15.pdf. (2025-01-06 閲覧).

[27] 預金取扱金融機関の耐量子計算機暗号への対応に関する検討会. 預金取扱金融機関の耐量子計算機暗号への対応に関する検討会報告書. 2024-11. https://www.fsa.go.jp/singi/pqc/houkokusyo.pdf. (2025-01-06 閲覧).

3

第 3 章

格子に基づく暗号技術

本章では格子に基づく暗号技術についてまとめる。格子に基づく暗号技術の安全性は，LWE (Learning with Errors) 問題，LWR (Learning with Rounding) 問題，NTRU 問題，およびそれらの変種等を含む，格子理論に関係する問題を解く計算の困難性に依存している。

3.1 格子に基づく暗号技術の安全性の根拠となる問題

3.1.1 LWE 問題の紹介

LWE 問題は機械学習理論から派生した求解困難な問題で，整数剰余類環 $\mathbb{Z}_q$ 上の秘密ベクトル $\mathbf{s} \in \mathbb{Z}_q^n$ に関するランダムな連立線形「近似」方程式が与えられたとき，その秘密ベクトルを復元する問題である。具体的な数値例として $n = 4,\ q = 17$ に対して，秘密ベクトル $\mathbf{s} = (s_1, s_2, s_3, s_4) \in \mathbb{Z}_{17}^4$ に関する連立線形近似方程式

$$
\begin{cases}
14s_1 + 15s_2 + 5s_3 + 2s_4 \approx 8 \pmod{17} \\
13s_1 + 14s_2 + 14s_3 + 6s_4 \approx 16 \pmod{17} \\
6s_1 + 10s_2 + 13s_3 + s_4 \approx 12 \pmod{17} \\
\qquad\qquad\qquad\quad \vdots \\
6s_1 + 7s_2 + 16s_3 + 2s_4 \approx 3 \pmod{17}
\end{cases}
$$

が与えられたとする。(この数値例は [88] から引用した。) ただし，各線形方程式の値は近似値であり，その誤差はこの例では ± 1 以内と仮定する。このとき，この連立線形近似方程式の解 $\mathbf{s}$ を求めるのが LWE 問題である。ここに示した数値例では $\mathbf{s} = (0, 13, 9, 11) \in \mathbb{Z}_{17}^4$ が解となる。LWE 問題で注意すべきことは, 連立線形近似方程式に誤差がない場合は，Gauss の消去法により効率的に解を求めることができる点である。逆に言うと，連立線形近似方程式で与えられる誤差の大きさが LWE 問題の求解を困難にする。

3.1.2 NTRU 問題の紹介

ここでは，NTRU 問題を紹介する。

定義 3.1 (NTRU 問題 [55]) 2 つの正の整数 n と q に対し，$\phi \in \mathbb{Z}[x]$ を次数 n の多項式とし，$R_q = \mathbb{Z}_q[x]/(\phi)$ とする。係数が小さい 2 つの多項式 $f \in R_q^{\times}, g \in R_q$ に対して，$h = g \cdot f^{-1} \in R_q$ とする。(特に，f は環 R_q の可逆元に注意) このとき，与えられた多項式 h から，f または g の多項式を復元する問題を (探索) NTRU 問題という。

NTRU 問題における多項式 ϕ の選び方として，$\phi = x^n \pm 1, x^n - x - 1, x^n - x^{n/2} + 1, \sum_{i=0}^{n-1} x^i$ などがある [7, Table 1]。(最後の ϕ のみ，次数は $n-1$ である。) また，多項式 f (または g) の選び方として，$\{-1, 0, 1\}$ などの小さい係数を持つ多項式や，小さい素数 p と係数が小さい多項式 F に対し $f = pF$ または $f = pF + 1$ と選ぶことが多い。

3.1.3 格子問題の公開チャレンジの求解状況

SVP や LWE に対する求解アルゴリズムをテストする目的で，ドイツ・ダルムシュタット大学によって「SVP チャレンジ」・「LWE チャレンジ」と呼ばれる求解コンテストがインターネット上で開催されている [31]。2018 年に，篩(ふるい)をベースとした高速な格子アルゴリズム群である General Sieve Kernel (G6K)[9] が提案され，SVP チャレンジ・LWE チャレンジの求解記録が飛躍的に更新された。具体的には，SVP チャレンジにおいては，G6K 内の篩アルゴリズムを GPU 実装することで，180 次元の SVP インスタンスが 4 NVIDIA Turing GPUs の計算機（1.5TB RAM）を用いて 51.6 日で求解されたと 2021 年 2 月に報告されている [40]。（ただし，本報告では Gaussian Heuristic で期待される最短ベクトル長に対する近似因子が 1.04002 なので，今回見つかった格子ベクトルは 180 次元 SVP インスタンスの厳密解ではなく近似解である。）また 2023 年 7 月に，186 次元の SVP インスタンスに対して，次のスペックを持つ計算機システムで約 50 日程度で近似因子が 1.01405 の非常に短い格子ベクトルを見つけることに成功している（計算時間の内訳は，Progressive pnj-BKZ による基底簡約に 12.3 日，Sieving に短い格子ベクトルの探索に 38 日かかったと報告されている）[32]。

- CPU: 1 ∗ Intel Xeon Gold 6330, 56 threads @ 2.00GHz
- GPU: 4 ∗ NVIDIA A100 80GB PCIe
- Max RAM used: 1441.6685 GB

さらに，2024 年 7 月に，190 次元の SVP インスタンスに対し 1.5TBytes（4 NVIDIA 4090）と 2.0TBytes（3 NVIDIA 4090 D）の RAM を持つ 2 つの計算機で G6K ライブラリの $\beta = 155$〜158 の篩次元を用いて，約 4 か月で近似因子が 1.04237 の短い格子ベクトルを見つけたと報告されている。

LWE チャレンジにおいては，$(n, \alpha) = (40, 0.040), (45, 0.030), (50, 0.025), (55, 0.020), (90, 0.005)$ の数多くの LWE インスタンスが G6K 内の progressive-BKZ の改良により求解されたと 2022 年 6〜10 月に報告されている。（ただし，n は LWE の秘密ベクトル長で，α はノイズの大きさに関するパラメータで，組 (n, α) のバランスで LWE インスタンスの難しさが大きく変化する。）例えば，$(n, \alpha) = (50, 0.025)$ と $(40, 0.040)$ の 2 つの LWE インスタンスに対して，次のスペックを持つ計算システムでそれぞれ約 592 時間と約 683 時間で求解されている [102]：

- HardwareCPU：AMD EPYC 7002 Series 128@2.6GHz
- RAM：1.5TB
- GPU：8 ∗ NVIDIA GeForce RTX 3090
- VRAM：8 ∗ 24GB（936.2 GB/s）

2024 年 9 月には，$(n, \alpha) = (95, 0.005)$ の LWE インスタンスに対して，最大 144 の篩次元を用いて約 46 日で求解したと報告されている（計算機は 8∗Nvidia RTX 4090, 2∗ Intel Xeon Platinum 8480+ Processor, 32∗ 64GB DDR5-4800MHz）。

3.2 格子に基づく代表的な暗号方式

3.2.1 Hash-and-Sign に基づく署名方式の格子問題への拡張

Hash-and-Sign に基づく署名方式は，Diffie,Hellman らによってその基本形が示されており，落とし戸つき一方向性関数 $f(x)$ ならびに $f^{-1}(x)$ を用いて署名・検証が行われる。

- M：メッセージ
- $h = hash(M)$: 暗号学的ハッシュ関数
- $\sigma = f^{-1}(h)$：署名
- $h = f(\sigma)$ が成り立つかを確認：署名検証

Diffie,Hellman らによる方式では，一方向性関数 $f(x)$ として，素数 p を法とした離散対数問題に基づく関数 $f(x) = a^x \mod p$ が提示されている。

この署名方式は，さまざまな改良が提案されているが，格子問題の困難性に基づく落とし戸つき関数を用いた Hash-and-Sign 署名方式が，Gentry らによって提案されている [52]。以下にその方式を示す。次のパラメータを準備する。

- m, n：正の整数 (セキュリティパラメータ)
- $hash(M)$: 暗号学的ハッシュ関数
- q：素数
- $L = m^{1+\epsilon}, (\epsilon > 0)$：秘密鍵の大きさの上限

以下に具体的な署名方式を示す。

鍵生成 $A \in \mathbb{Z}_q^{n \times m}$ をランダムな行列，$S \in \Lambda_q^{\perp}(\mathbf{A}, \mathbf{q}), ||S|| < L$ を短いベクトルとし，$SA = 0 \mod q$ を満たす行列の組 (A, S) を生成する (具体的な手法は [3] 参照)。秘密鍵を S，公開鍵を A とする。

署名生成 メッセージ M に対しハッシュ関数を作用させた値 $H = hash(M)$ を $D_{\mathbb{Z}^m, s}$ にマッピングし，その値を u とする。$tA = u \mod q$ を満たす t を任意に求める。秘密鍵 S を用いて，$-t$ に近い格子 $\Lambda_q^{\perp}(\mathbf{A}, \mathbf{q})$ 上の点 v を求め，$\sigma = v + t$ とする。σ を署名として出力する。

署名検証 メッセージ m にハッシュ関数を作用させた値 $h = hash(m)$ を $D_{\mathbb{Z}^m, s}$ にマッピングし，値を u とする。σ が短いベクトルでありかつ $(\sigma - u)A = 0$ である場合に正当な署名として受理する。

署名の正当性については, 次のように示される。構成の仕方から，$\sigma - u = v$ であり，v は格子 $\Lambda_q^{\perp}(\mathbf{A}, \mathbf{q})$ 上の点であるから，$(\sigma - u)A \mod q = vA \mod q = 0$ が成り立つ。また 秘密鍵 S の特徴から，$\sigma \in D_{\mathbb{Z}^m, s}$ であることから，σ は短いベクトルとなる。本署名方式は LWE 仮定の元で SUF-CMA(Strong Existential Unforgeability under Chosen Message Attack) 安全であることが示されている。

3.2.2 Fiat-Shamir 署名方式の格子問題への拡張

Fiat, Shamir らによって提示された Fiat-Shamir 変換 [48] に基づく署名方式を総称して Fiat-Shamir 署名と呼ばれており，現在までさまざまな方式が提案されている。以下に基本となる方式の一つである素因数分解問題をベースとする方式を記す。合成数 $n = pq$ (p, q は素数) を法とするべき乗剰余演算 $g(x) = g^x \mod n$ を一方向性関数として利

用し，秘密鍵 s，公開鍵 $a = g(s)$ を準備する。

- M：メッセージ m
- $h = hash(M)$: 暗号学的ハッシュ関数
- r：ランダムな値
- $(z, y) = (g(r)h + s, g(r))$：署名
- $g(z) = a^r y$ が成り立つかを確認：署名検証

Lyubashevsky によって，Fiat-Shamir with Aborts 型の格子ベースの署名方式が提案されている [48]。以下にその具体的な署名方式について述べる。次のパラメータを準備する。

- $hash(M)$: 暗号学的ハッシュ関数
- m：正の整数 (セキュリティパラメータ)
- n：2 のべき乗 (セキュリティパラメータ)
- σ：正の整数 (セキュリティパラメータ)
- κ：${}^{2^\kappa}{}_n C_\kappa > 160$ を満たす整数
- p：$(2\sigma + 1)^m 2^{-128/n}$ 程度の素数
- $R = \mathbb{Z}_p[x]/(x^n + 1)$：多項式剰余環
- $D = \{z \in R \mid ||g||_\infty \leq mn\sigma\kappa\}$：内積に基づくハッシュ関数向け空間
- $G = \{g \in R \mid ||g||_\infty \leq mn\sigma\kappa - \sigma\kappa\}$：署名空間

ただし $||z||_\infty$ は z の最大値ノルムとする。以下に具体的な署名方式を示す。

R に属する m 個の多項式の集合 R^m の要素 $\hat{a}$ に対し，D^m 上のハッシュ関数 $h_{\hat{a}}(\hat{z}), (\hat{z} \in D^m)$ を以下のように定める。$h_{\hat{a}}(\hat{z}) = \hat{a} \cdot \hat{z} = a_1 z_1 + \cdots + a_m z_m \in R$.

鍵生成 短い多項式を成分とするランダムなベクトル $\hat{s}$，ならびに D^m のランダムなベクトル $\hat{a}$ によるハッシュ関数 $h_{\hat{a}}()$ を作用させた値 $S = h_{\hat{a}}(\hat{s})$ を求め，$\hat{s}$ を秘密鍵，S を公開鍵とする。

署名生成 メッセージを M とする。

多項式を成分とするベクトル $\hat{y} \in D^m$ をランダムに選択し，$c = hash(h_{\hat{a}}(\hat{y})||M), \hat{z} = \hat{y} + c\hat{s}$ を求める。$\hat{z} \in G^m$ となるまで，ベクトル $\hat{y}$ の選択をくりかえす。$\sigma = (\hat{z}, c)$ を署名として出力する。

署名検証 $\hat{z} \in G^m$ ならびに $c = hash(h_{\hat{a}}(\hat{z}) - Sc, M)$ が成り立つ場合に署名を受理する。

この署名方式の正当性は，$h_{\hat{a}}(\hat{z}) - Sc = h_{\hat{a}}(\hat{y} + c\hat{s}) - h_{\hat{a}}(\hat{s})c = h_{\hat{a}}(\hat{y})$ が成り立つことから保証される。安全性については，環 R 上のイデアルに対する γ-SVP 問題の困難性と等価であることが示されている。

3.3 格子に基づく主要な暗号方式

本節では，格子に基づく主要な暗号方式として，表 3.1 に挙げる公開鍵暗号と 2 つの署名を取り上げ，その概要と設計原理を説明する。

格子を用いた主な公開鍵暗号の構成として，最初期の Ajtai-Dwork 型 [4]，GGH 型 [53] から近年の [87] による LWE 型 (Regev 型)，[52, 64] に代表される dual-LWE 型，[55] に代表される NTRU 型が存在する。格子を用いた署名の構成としては主に GGH/NTRUSign 型 [53, 55]，Fiat-Shamir with abort 型 [67, 68]，Hash-and-Sign 型 [52, Sect.6],

Plantard-Susilo-Win 型 [83] 等が知られている[*1]。

また，安全性の根拠となる計算問題に関しても，最短ベクトル問題に直接還元するもの，LWE 問題，SIS 問題，LWR 問題およびそれらの Module 版，Ring 版へと還元するもの，NTRU 問題に還元するものへと分類できる。

表 3.1: 格子に基づく暗号の分類

文献	暗号化	鍵交換	署名
ML-KEM (FIPS 203) [79]	○	○	
ML-DSA (FIPS 204) [78]			○
FALCON [51]			○

- ML-KEM は CRYSTALS-Kyber に基づく dual-LWE 型の公開鍵暗号であり，安全性の根拠に $x^n+1, n=2^k$ の形の多項式により定義される環上の Module-LWE 問題の困難性を置いている。NIST により FIPS 標準アルゴリズムとして制定されたことから，取り上げる。
- ML-DSA は CRYSTALS-Dilithium に基づく Fiat-Shamir 型の署名方式であり，$x^{256}+1$ を定義多項式とする環上の Module-LWE 問題の計算困難性を安全性の根拠としている。環の性質を用いた数論変換による高速処理とサイズの圧縮が可能であり，公開鍵サイズと署名サイズの和を最小化することを目的としてパラメータ設計を行っている。NIST により FIPS 標準アルゴリズムとして制定されたことから，取り上げる。
- FALCON は Hash-and-Sign 型の署名方式であり，x^n+1 を定義多項式とする NTRU 格子上の SIS 問題の困難性を安全性の根拠としている。格子上の高速フーリエサンプリングを用いた高速な署名生成を特徴とし，方式提案後も数多くの改良が提案されていることから取り上げる。

3.3.1 FIPS 203：Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)

KEM とは公開されたチャンネル上で 2 者が秘密を共有するアルゴリズム群である。KEM で安全に生成された共有の秘密は共通鍵暗号で用いられ，暗号や認証などの安全なやり取りの中で重要な役割を果たす。ML-KEM [79] は CRYSTALS-Kyber に基づく KEM で，その安全性は加群上の LWE 問題の計算量困難性に基づく。具体的には，2 のべき数 $n=256$ に対し $R:=\mathbb{Z}[X]/(X^n+1)$ を基本環とし，素数 $q=3329$ に対し $R_q:=R/qR=\mathbb{Z}_q[X]/(X^n+1)$ をその剰余環とする。環 R_q の元は $\mathbb{Z}_q$ を係数とする $n-1$ 以下の次数の多項式 $f=f_0+f_1X+\cdots+f_{n-1}X^{n-1}$ と表せ，その係数ベクトル $(f_0,f_1,\ldots,f_{n-1})$ を対応させることで，$\mathbb{Z}_q$ 加群として R_q は $\mathbb{Z}_q^n$ と同型である。ML-KEM は，階数パラメータ $k\in\{2,3,4\}$ に対し，$\mathbb{Z}_q$ 加群 $R_q^k\simeq(\mathbb{Z}_q^n)^k$ 上の LWE 問題を安全性の根拠とした KEM である。特に，R_q における乗算を高速化するために，Number-Theoretic Transform（NTT）とよばれる変換を利用する。ここでは，ML-KEM の最も基本となる構成要素である NTT を説明したのちに，ML-KEM の基本構成について説明する。

3.3.1.1 数論変換：Number-Theoretic Transform (NTT)

NTT は，環 R_q の元 f を R_q と同型な環 T_q の元 $\widehat{f}$ に写し，T_q における乗算を利用して効率的に R_q の 2 つの元の乗算を行う手法である。これは $\mathbb{C}$ 上の高速フーリエ変換による多項式乗算と同じアイデアで，NTT はその $\mathbb{Z}_q$ 上版とみなせる。ML-KEM では，2 のべき数 $n=2^8=256$ と素数 $q=3329$ で定まる剰余環 $R_q=\mathbb{Z}_q[X]/(X^n+1)$ を用

[*1] この分類に関しては例えば [57, Sect. 3]，[46, Sect. 5.5] 等を参照。

いる（ML-KEM の暗号パラメータについては，後述の 3.3.1.3 節を参照）。これらの暗号パラメータ (n, q) において，$\mathbb{Z}_q^* := \mathbb{Z}_q \setminus \{0\}$ は位数 $q-1 = 3328 = 2^8 \cdot 13$ の巡回群で，$\mathbb{Z}_q^*$ は位数 $2^8 = 256 = n$ の巡回部分群 $\langle\zeta\rangle$ を唯一つ含む。具体的には，$\mathbb{Z}_q$ において $\zeta := 17 \bmod q$ が 1 の原始 n 乗根で，$\{\zeta, \zeta^3, \ldots, \zeta^{n-1}\}$ が $\mathbb{Z}_q$ に含まれる 1 の原始 n 乗根のすべてである。ここで，$N = \frac{n}{2} = 128$ とおくと，各 $i = 0, 1, \ldots, N-1$ に対して，$\zeta^{(2i+1)N} \equiv -1 \pmod q$ である。ゆえに，多項式環 $\mathbb{Z}_q[X]$ において，$X^n + 1$ は次のように N 個の 2 次式の積に分解できる。

$$X^n + 1 = \prod_{i=0}^{N-1} \left(X^2 - \zeta^{2i+1}\right) = \prod_{i=0}^{N-1} \left(X^2 - \zeta^{2\mathsf{BitRev}_7(i)+1}\right) \in \mathbb{Z}_q[X].$$

ただし，$\mathsf{BitRev}_7(i)$ は符号なし 7 ビット整数 i のビット逆順整数を表し，実装上の都合のため ML-KEM ではこの順序を利用する。以下では，数論変換の原理を説明するために，$i = 0, 1, \cdots, N-1$ の単純な順序を用いる。上記の X^n+1 の分解により，次の（$\mathbb{Z}_q$ 加群としての）同型を得る。

$$R_q = \mathbb{Z}_q[X]/(X^n+1) \simeq \bigoplus_{i=0}^{N-1} \mathbb{Z}_q[X]/\left(X^2 - \zeta^{2i+1}\right) =: T_q.$$

具体的には，この同型は

$$\mathsf{NTT} : R_q \longrightarrow T_q, \quad f \longmapsto \widehat{f} := \left(f \bmod (X^2 - \zeta^{2i+1})\right)_{i=0}^{N-1} \tag{3.1}$$

で定まる。特に，T_q を **NTT 空間**，$\widehat{f} = \mathsf{NTT}(f) \in T_q$ を $f \in R_q$ の **NTT 表現**とよぶ。

■NTT 表現について　$f = f_0 + f_1 X + \cdots + X^{n-1} \in R_q$ の偶数と奇数の次数に関する多項式をそれぞれ

$$f_e := f_0 + f_2 Y + f_4 Y^2 + \cdots + f_{2N-2} Y^{N-1}, \quad f_o := f_1 + f_3 Y + f_5 Y^2 + \cdots + f_{2N-1} Y^{N-1} \in \mathbb{Z}_q[Y]$$

とおく。構成から $f = f_e(X^2) + f_o(X^2)X$ なので，各 $i = 0, 1, \ldots, N-1$ に対して，

$$\widehat{f}_{2i} := f_e(\zeta^{2i+1}) = \sum_{j=0}^{N-1} f_{2j} \zeta^{(2i+1)j}, \quad \widehat{f}_{2i+1} := f_o(\zeta^{2i+1}) = \sum_{j=0}^{N-1} f_{2j+1} \zeta^{(2i+1)j}$$

とおくと，

$$f \equiv \widehat{f}_{2i} + \widehat{f}_{2i+1} X \pmod{\left(X^2 - \zeta^{2i+1}\right)} \tag{3.2}$$

が成り立つ。これより，f の NTT 表現について，$\widehat{f} = \left(\widehat{f}_{2i} + \widehat{f}_{2i+1} X\right)_{i=0}^{N-1} \in T_q$ とかける。

■NTT 表現の行列表示　$\mathbb{Z}_q$ の元を成分とする $N \times N$ 行列を

$$\mathbf{B} = A(\zeta) := \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \zeta^3 & \zeta^6 & \cdots & \zeta^{3(N-1)} \\ 1 & \zeta^5 & \zeta^{10} & \cdots & \zeta^{5(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{2N-1} & \zeta^{2(2N-1)} & \cdots & \zeta^{(N-1)(2N-1)} \end{pmatrix} \in (\mathbb{Z}_q)^{N \times N}$$

とおく。R_q の元 $f = f_0 + f_1 X + \cdots + X^{n-1}$ の偶数と奇数の次数に関するそれぞれの係数ベクトル $(f_0, f_2, \ldots, f_{2N-2}), (f_1, f_3, \ldots, f_{2N-1}) \in \mathbb{Z}_q^N$ に対して

$$\begin{pmatrix} \widehat{f}_0 \\ \widehat{f}_2 \\ \widehat{f}_4 \\ \vdots \\ \widehat{f}_{2N-2} \end{pmatrix} = \begin{pmatrix} f_e(1) \\ f_e(\zeta^3) \\ f_e(\zeta^5) \\ \vdots \\ f_e(\zeta^{2N-1}) \end{pmatrix} = \mathbf{B} \begin{pmatrix} f_0 \\ f_2 \\ f_4 \\ \vdots \\ f_{2N-2} \end{pmatrix}, \quad \begin{pmatrix} \widehat{f}_1 \\ \widehat{f}_3 \\ \widehat{f}_5 \\ \vdots \\ \widehat{f}_{2N-1} \end{pmatrix} = \begin{pmatrix} f_o(1) \\ f_o(\zeta^3) \\ f_o(\zeta^5) \\ \vdots \\ f_o(\zeta^{2N-1}) \end{pmatrix} = \mathbf{B} \begin{pmatrix} f_1 \\ f_3 \\ f_5 \\ \vdots \\ f_{2N-1} \end{pmatrix}$$

が成り立つ。つまり，$f \in R_q$ の偶数と奇数の次数の係数ベクトルはそれぞれ $\mathbf{B}$ による線形変換（つまり，離散フーリエ変換）で $\widehat{f} \in T_q$ の偶数と奇数の添え字番号のベクトルに写る。$\mathbf{B}$ の逆行列は $\mathbf{C} := \frac{1}{N} A(\zeta^{-1})$ なので，式 (3.1) の NTT 写像の逆写像 NTT^{-1} は行列 $\mathbf{C}$ を用いて計算可能である（つまり，逆離散フーリエ変換から計算可能）。

■NTT 空間における乗算　R_q の 2 つの元 f, g に対して，その積を $h := f \cdot g \in R_q$ とおく。h の NTT 表現 $\widehat{h} \in T_q$ について，式 (3.2) から，各 $i = 0, 1, \ldots, N-1$ に対して

$$\widehat{h}_{2i} + \widehat{h}_{2i+1} X \equiv h = f \cdot g \equiv (\widehat{f}_{2i} + \widehat{f}_{2i+1} X)\,(\widehat{g}_{2i} + \widehat{g}_{2i+1} X) \bmod \left(X^2 - \zeta^{2i+1}\right)$$

が成り立つ。ここで，2 つの NTT 表現 $\widehat{f} = \left(\widehat{f}_{2i} + \widehat{f}_{2i+1} X\right)_{i=0}^{N-1}, \widehat{g} = (\widehat{g}_{2i} + \widehat{g}_{2i+1} X)_{i=0}^{N-1} \in T_q$ の積を

$$\begin{aligned}\widehat{f} \circ \widehat{g} &:= \left(\left(\widehat{f}_{2i} + \widehat{f}_{2i+1} X\right) \cdot (\widehat{g}_{2i} + \widehat{g}_{2i+1} X) \bmod \left(X^2 - \zeta^{2i+1}\right)\right)_{i=0}^{N-1} \\ &= \left(\widehat{f}_{2i}\widehat{g}_{2i} + \widehat{f}_{2i+1}\widehat{g}_{2i+1}\zeta^{2i+1} + \left(\widehat{f}_{2i}\widehat{g}_{2i+1} + \widehat{f}_{2i+1}\widehat{g}_{2i}\right) X\right)_{i=0}^{N-1} \in T_q\end{aligned}$$

と定めると，

$$\mathsf{NTT}(f \cdot g) = \mathsf{NTT}(f) \circ \mathsf{NTT}(g) \iff f \cdot g = \mathsf{NTT}^{-1}\left(\widehat{f} \circ \widehat{g}\right) \in R_q$$

が成り立つ（つまり，式 (3.1) の NTT 写像は環の同型写像である）。特に，NTT 空間 T_q における乗算は，成分ごとの演算であるため，（R_q における乗算に比べて）効率的に計算可能である。

3.3.1.2　ML-KEM の基本構成と処理概要

加群 $R_q^k \simeq (\mathbb{Z}_q^n)^k$ 上の LWE 問題に基づく ML-KEM は 2 つのステップで構成される。1 つ目は R_q^k 上の LWE 問題から公開鍵暗号（K-PKE）を構成し，2 つ目は藤崎-岡本変換により KEM に変換する。藤崎-岡本変換の性質により，公開鍵暗号方式から構成される KEM はより一般的な攻撃モデルにおいて安全であり，IND-CCA2 安全性を満たす。

■K-PKE の処理概要　ここでは，K-PKE の処理概要とその原理が分かるように，簡略化した形で各アルゴリズムの処理を説明する。特に，処理の高速化のために，NTT 変換を適宜利用する。

K-PKE 鍵生成　鍵生成アルゴリズム（FIPS 203 の Algorithm 13, $\mathsf{K\text{-}PKE.KeyGen}(d)$）では，乱数 d を入力として，暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ と復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ を次のように出力する。

- 入力：乱数 d
- 出力：暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ と復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$
  1. $(\rho, \sigma) \leftarrow \mathsf{G}(d \| k)$：ハッシュ関数 G を用いて擬似ランダムな乱数の組 (ρ, σ) を生成
  2. $\widehat{\mathbf{A}} = \left(\widehat{\mathbf{A}}[i,j]\right)_{0 \le i,j < k} \in (T_q)^{k \times k}$: 乱数 ρ を用いて，NTT 表現の公開鍵行列を生成
  3. $\mathbf{s} = (\mathbf{s}[i])_{0 \le i < k} \in R_q^k$：各 $\mathbf{s}[i] \in R_q$ のすべて $\mathbb{Z}_q$ 係数は中心二項分布 CBD からサンプルする（十分小さい）
  4. $\mathbf{e} = (\mathbf{e}[i])_{0 \le i < k} \in R_q^k$：各 $\mathbf{e}[i] \in R_q$ のすべての $\mathbb{Z}_q$ 係数は CBD からサンプルする（十分小さい）
  5. $\widehat{\mathbf{s}} = (\mathsf{NTT}(\mathbf{s}[i]))_{0 \le i < k} \in T_q^k$：各 $\mathbf{s}[i]$ を NTT 変換
  6. $\widehat{\mathbf{e}} = (\mathsf{NTT}(\mathbf{e}[i]))_{0 \le i < k} \in T_q^k$：前のステップ同様，各 $\mathbf{e}[i]$ を NTT 変換
  7. $\widehat{\mathbf{t}} = \widehat{\mathbf{A}} \circ \widehat{\mathbf{s}} + \widehat{\mathbf{e}} = \left(\sum_{j=0}^{k-1} \widehat{\mathbf{A}}[i,j] \circ \widehat{\mathbf{s}}[j] + \widehat{\mathbf{e}}[i]\right)_{0 \le i < k} \in T_q^k$：NTT 空間上で LWE 関係式を生成
  8. $\mathtt{ek}_{\mathtt{PKE}} = \left(\widehat{\mathbf{t}}, \rho\right), \mathtt{dk}_{\mathtt{PKE}} = \widehat{\mathbf{s}}$（公開鍵行列 $\widehat{\mathbf{A}}$ は ρ から復元可能であることに注意）
  9. $(\mathtt{ek}_{\mathtt{PKE}}, \mathtt{dk}_{\mathtt{PKE}})$ を出力

ステップ 2 において，NTT 表現の公開鍵行列の各成分 $\widehat{\mathbf{A}}[i,j]$ は，入力する乱数から擬似ランダムな T_q の元を出力する SampleNTT 関数（FIPS 203 の Algorithm 7）を用いて生成する（具体的には，$\widehat{\mathbf{A}}[i,j] \leftarrow \mathsf{SampleNTT}(\rho \| i \| j)$ と生成)。ステップ 3, 4 において，各 $\mathbf{s}[i]$ または $\mathbf{e}[i]$ の多項式のすべての $\mathbb{Z}_q$ 係数は，SamplePolyCBD 関数（FIPS 203 の Algorithm 8）を用いて生成する。具体的には，$\eta \in \{2,3\}$ に対する $\mathbb{Z}_q$ 上の二項分布 CBD_η を

(i) $(x_1, \ldots, x_\eta, y_1, \ldots, y_\eta) \in \{0,1\}^{2\eta}$ を一様ランダムにサンプルする
(ii) $\sum_{i=1}^{\eta}(x_i - y_i) \bmod q \in \mathbb{Z}_q$ を出力

と定め，$\mathbf{s}[i]$ と $\mathbf{e}[i]$ の各 $\mathbb{Z}_q$ 係数は CBD_η からサンプルする（CBD_η の引数の一つとして，ステップ 1 で生成した σ を用いる)。ステップ 8 において，FIPS 203 では $\left(\widehat{\mathbf{t}}, \rho\right)$ と $\widehat{\mathbf{s}}$ をそれぞれ符号化関数 ByteEncode（FIPS 203 の Algorithm 5）で符号化したものを暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ と復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ とする。

鍵生成アルゴリズムにおいて，ρ から NTT 表現の公開鍵行列 $\widehat{\mathbf{A}}$ が復元可能なので，暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ は NTT 表現の LWE インスタンスの組 $\left(\widehat{\mathbf{A}}, \widehat{\mathbf{t}}\right)$ に対応する。特に，$\mathbf{t} := \mathsf{NTT}^{-1}\left(\widehat{\mathbf{t}}\right) \in R_q^k, \mathbf{A} := \mathsf{NTT}^{-1}\left(\widehat{\mathbf{A}}\right) \in (R_q)^{k \times k}$ とおくと，R_q^k 上の LWE 関係式 $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$ が成り立つ。一方，復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ は NTT 表現の LWE の秘密 $\widehat{\mathbf{s}}$ であるので，暗号鍵から復号鍵を見つけるのは $T_q^k \simeq R_q^k$ 上の探索 LWE 問題である。特に，適切な暗号パラメータ（後述の 3.3.1.3 節を参照）を利用した場合，その LWE 問題を解くのは計算量的に非常に困難である。また，鍵生成アルゴリズムにおいて，NTT 空間上で公開鍵行列 $\widehat{\mathbf{A}}$ を直接生成すると共に，ステップ 7 で NTT 空間上で LWE 関係式を生成することで，計算の高速化を図る。

K-PKE 暗号化　暗号化アルゴリズム（FIPS 203 の Algorithm 14, K-PKE.Encrypt）では，暗号化鍵 $\mathtt{ek}_{\mathtt{PKE}}$ と平文 m を入力として，次のように暗号文 c を出力する。

- 入力：暗号化鍵 $\mathtt{ek}_{\mathtt{PKE}} = \left(\widehat{\mathbf{t}}, \rho\right)$，平文 m と乱数 r
- 出力：暗号文 c
  1. ρ から NTT 表現の公開鍵行列 $\widehat{\mathbf{A}} \in (T_q)^{k \times k}$ を復元
  2. $\mathbf{y} = (\mathbf{y}[i])_{0 \le i < k} \in R_q^k$：各 $\mathbf{y}[i] \in R_q$ のすべての $\mathbb{Z}_q$ 係数は二項分布 CBD からサンプルする（十分小さい）
  3. $\mathbf{e}_1 = (\mathbf{e}_1[i])_{0 \le i < k} \in R_q^k$：各 $\mathbf{e}[i] \in R_q$ のすべての $\mathbb{Z}_q$ 係数は CBD からサンプルする（十分小さい）
  4. $e_2 \in R_q$：すべての $\mathbb{Z}_q$ 係数は CBD からサンプルする（十分小さい）
  5. $\widehat{\mathbf{y}} = (\mathsf{NTT}(\mathbf{y}[i]))_{0 \le i < k} \in T_q^k$
  6. $\mathbf{u} = \mathsf{NTT}^{-1}\left(\widehat{\mathbf{A}}^\top \circ \widehat{\mathbf{y}}\right) + \mathbf{e}_1 = \mathbf{A}^\top \mathbf{y} + \mathbf{e}_1 = \left(\sum_{j=0}^{k-1} \mathbf{A}[j,i]\mathbf{y}[j] + \mathbf{e}_1[i]\right)_{0 \le i < k} \in R_q^k$
     （ただし，$\mathbf{A} = \mathsf{NTT}^{-1}(\widehat{\mathbf{A}}) = (\mathbf{A}[i,j])_{0 \le i,j < k} \in (R_q)^{k \times k}$ とする）
  7. $\mu = \mathsf{Decompress}(\mathsf{ByteDecode}(m)) \in R_q$：平文 m をビット列化した後に R_q の元に変換
  8. $v = \mathsf{NTT}^{-1}\left(\widehat{\mathbf{t}}^\top \circ \widehat{\mathbf{y}}\right) + e_2 + \mu = \mathbf{t}^\top \mathbf{y} + e_2 + \mu \in R_q$
  9. $c = (\mathbf{u}, v) \in R_q^k \times R_q$ を出力

ステップ 2, 3, 4 において，r をシードとした擬似乱数を引数とした SamplePolyCBD 関数で，すべての $\mathbb{Z}_q$ 係数が十分小さい多項式を生成する。ステップ 7 では，バイト列で表現された平文 m を ByteDecode 関数（FIPS 203, Algorithm 6）でビット列 $(m_0, m_1, \ldots, m_{n-1})$ に変換した後に，各ビット $m_i \in \{0,1\}$ を Decompress 関数で $\mu_i := \left\lceil \frac{q}{2} \cdot m_i \right\rfloor \in \mathbb{Z}_q$ に変換する。また，各 μ_i を係数とする多項式を $\mu = \mu_0 + \mu_1 x + \cdots + \mu_{n-1} x^{n-1} \in R_q$ とする。ステップ 9 において，FIPS 203 では $\mathbf{u}$ と v はそれぞれ Compress 関数で圧縮した後，ByteEncode 関数で符号化する。

上記の暗号化アルゴリズムにおいて，暗号文は $c = (\mathbf{u}, v) = \left(\mathbf{A}^\top \mathbf{y} + \mathbf{e}_1, \mathbf{t}^\top \mathbf{y} + e_2 + \mu\right) \in R_q^k \times R_q$ の形で，LWE に基づく Lindner-Peikert による暗号方式と同様，R_q^k 上の LWE 問題が計算困難であれば，暗号文から μ（つまり，平

文 m）の情報が洩れない。また，ステップ 6 と 8 において，NTT 空間上で $\mathbf{A}^\top\mathbf{y}$ と $\mathbf{t}^\top\mathbf{y}$ を計算することで，計算の高速化を図る。

K-PKE 復号　復号アルゴリズム（FIPS 203 の Algorithm 15, K-PKE.Decrypt）では，復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ と暗号文 c を入力とし，次のように復号文 m' を出力する。

- 入力：復号鍵 $\mathtt{dk}_{\mathtt{PKE}} = \widehat{\mathbf{s}}$ と暗号文 $c = (\mathbf{u}, v)$
- 出力：復号文 m'
  1. $w = v - \mathsf{NTT}^{-1}\left(\widehat{\mathbf{s}}^\top \circ \mathsf{NTT}(\mathbf{u})\right) = v - \mathbf{s}^\top\mathbf{u} \in R_q$
  2. $m' = \mathsf{ByteEncode}\left(\mathsf{Compress}(w)\right)$ を出力

ステップ 2 において，多項式表現の R_q の元 $w = w_0 + w_1 x + \cdots + w_{n-1}x^{n-1}$ に対して，各係数 $w_i \in \mathbb{Z}_q$ を Compress 関数で $z_i := \left\lceil \frac{2}{q} \cdot w_i \right\rfloor \bmod 2 \in \{0, 1\}$ に変換する。また，ビット列 $(z_0, z_1, \ldots, z_{n-1})$ を ByteEncode 関数（FIPS 203, Algorithm 5）でバイト列に変換する。特に，ByteEncode 関数と ByteDecode 関数は互いの逆関数である。

上記の復号アルゴリズムにおいて，暗号文 $c = (\mathbf{u}, v) = \left(\mathbf{A}^\top\mathbf{y} + \mathbf{e}_1, \mathbf{t}^\top\mathbf{y} + e_2 + \mu\right) \in R_q^k \times R_q$ に対して，R_q^k 上の LWE 関係式 $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$ から，

$$\begin{aligned} w = v - \mathbf{s}^\top\mathbf{u} &= \mathbf{t}^\top\mathbf{y} + e_2 + \mu - (\mathbf{A}\mathbf{s})^\top\mathbf{y} - \mathbf{s}^\top\mathbf{e}_1 \\ &= \mathbf{t}^\top\mathbf{y} + e_2 + \mu - \left(\mathbf{t}^\top + \mathbf{e}^\top\right)\mathbf{y} - \mathbf{s}^\top\mathbf{e}_1 = \mu + \underbrace{e_2 - \mathbf{e}^\top\mathbf{y} - \mathbf{s}^\top\mathbf{e}_1}_{\text{すべての } \mathbb{Z}_q \text{ 係数が十分小さい}} \in R_q \end{aligned}$$

が成り立つ。ここで，$\mathbf{s}, \mathbf{e}, \mathbf{e}_1, \mathbf{y} \in R_q^k$ の各成分 $\mathbf{s}[i], \mathbf{e}[i], \mathbf{e}_1[i], \mathbf{y}[i] \in R_q$ と $\mu \in R_q$ のすべての $\mathbb{Z}_q$ 係数は十分小さいことに注意する。よって，Compress 関数による各 $\mathbb{Z}_q$ 係数におけるノイズ補正により

$$\mathsf{Compress}(w) = \mathsf{Compress}(\mu) = (m_0, m_1, \ldots, m_{n-1}) \in \{0, 1\}^{n-1}$$

が成り立つ。最後に，ByteEncode 関数により，平文のビット列 $(m_0, m_1, \ldots, m_{n-1})$ をバイト列に変換することで，元の平文 m に復号できる（つまり，復号文 m' は平文 m に一致する）。また，ステップ 1 において，NTT 空間上で $\mathbf{s}^\top\mathbf{u}$ を計算することで，計算の高速化を図る。

■ML-KEM の処理概要　上記で構成した K-PKE 方式を用いて，ML-KEM を下記のように構成する。

ML-KEM 鍵生成　鍵生成アルゴリズム（FIPS 203, Algorithm 16）では，K-PKE 鍵生成アルゴリズムを用いて，2 つの乱数 d, z から鍵カプセル化鍵 $\mathtt{ek}$ とデカプセル化鍵 $\mathtt{dk}$ を次のように出力する。

- 入力：2 つの乱数 d, z
- 出力：鍵カプセル化鍵 $\mathtt{ek}$ とデカプセル化鍵 $\mathtt{dk}$
  1. K-PKE 鍵生成アルゴリズムで，乱数 d から $(\mathtt{ek}_{\mathtt{PKE}}, \mathtt{dk}_{\mathtt{PKE}})$ を生成
  2. $\mathtt{ek} = \mathtt{ek}_{\mathtt{PKE}}$
  3. $\mathtt{dk} = (\mathtt{dk}_{\mathtt{PKE}}, \mathtt{ek}, \mathsf{H}(\mathtt{ek}), z)$：H はハッシュ関数
  4. $(\mathtt{ek}, \mathtt{dk})$ を出力

ML-KEM 鍵カプセル化　鍵カプセル化アルゴリズム（FIPS 203, Algorithm 17）では，K-PKE 暗号化アルゴリズムを用いて，鍵カプセル化鍵 $\mathtt{ek}$ と乱数 m から共有の秘密鍵 K と暗号文 c を次のように出力する。

- 入力：鍵カプセル化鍵 $\mathtt{ek}$ と乱数 m
- 出力：共有の秘密鍵 K と暗号文 c

1. $(K, r) = \mathsf{G}\,(m \| \mathsf{H}(\mathtt{ek}))$：G はハッシュ関数
2. K-PKE 暗号化アルゴリズムで，$(\mathtt{ek}, m, r)$ から暗号文 c を生成
3. (K, c) を出力

ML-KEM デカプセル化　デカプセル化アルゴリズム (FIPS 203, Algorithm 18) では，K-PKE 復号アルゴリズムを用いて，デカプセル化 dk と暗号文 c から，共有の秘密鍵 K を次のように出力する。また，c が改竄されていないことを保証するために，K-PKE 暗号化アルゴリズムで復号文から暗号文 c' を生成し，c と c' が一致するか検証する。

- 入力：デカプセル化 $\mathtt{dk} = (\mathtt{dk}_{\mathrm{PKE}}, \mathtt{ek}, \mathsf{H}(\mathtt{ek}), z)$ と暗号文 c
- 出力：共有の秘密鍵 K
  1. K-PKE 復号アルゴリズムで，復号鍵 $\mathtt{dk}_{\mathrm{PKE}}$ と暗号文 c から，復号文 m' を生成
  2. $(K', r') = \mathsf{G}\,(m' \| \mathsf{H}(\mathtt{ek}))$
  3. $\bar{K} = \mathsf{J}(z \| c)$：J はハッシュ関数
  4. K-PKE 暗号化アルゴリズムで，$(\mathtt{ek}, m', r')$ から暗号文 c' を生成
  5. $c \neq c'$ の場合は，$K' = \bar{K}$ とおく
  6. K' を出力

3.3.1.3　暗号パラメータ

ML-KEM における主な暗号パラメータと対応する鍵や暗号文のサイズと安全性レベルは以下である。具体的には，LWE の次元 $n = 256$ と剰余素数 $q = 3329$ は ML-KEM-512, -768, -1024 の 3 種類の暗号パラメータで共通であるが，主に 3 種類の階数パラメータ $k \in \{2, 3, 4\}$ により安全性レベルが異なる。(ML-KEM のパラメータ名は，$n \times k \in \{512, 768, 1024\}$ の値により名づけられている。)

	暗号パラメータ			サイズ（単位：バイト）				安全性
	n	q	k	カプセル化鍵	デカプセル化鍵	暗号文	共有の秘密鍵	レベル
ML-KEM-512	256	3329	2	800	1632	768	32	レベル 1
ML-KEM-768	256	3329	3	1184	2400	1088	32	レベル 3
ML-KEM-1024	256	3329	4	1568	3168	1568	32	レベル 5

3.3.1.4　CRYSTALS-Kyber との違い

- Kyber の round 3 version では，共有する秘密鍵は長さが可変な値として扱われていた。一方，ML-KEM では，その長さは 256 ビットに固定している。また，その鍵は直接共通鍵として利用できる。
- ML-KEM.Encaps と ML-KEM.Decaps のアルゴリズムでは，第 3 ランド仕様とは異なる藤崎-岡本変換を利用する。具体的には，ML-KEM.Encaps は共有する秘密の導出において暗号文のハッシュ値を含まず，ML-KEM.Decaps ではその変更に合わせている。
- 第 3 ラウンドの仕様では，ML-KEM.Ecaps アルゴリズム内の初期乱数 m は使う前にハッシュ化される。具体的には，アルゴリズム 16 の 1 と 2 行目の間に，$m \leftarrow H(m)$ のステップがあったが，ML-KEM ではその処理は不必要で行わない。
- ML-KEM では, 第 3 ラウンドの仕様にはなかった入力データの検証ステップを含む。例えば, ML-KEM.Encaps

では，カプセル化キーを含むバイト配列が，モジュラー還元なしで q を法とする整数配列に正しくデコードされることを必要とする。

3.3.2 FIPS 204: Module-Lattice-Based Digital Signature Standard（ML-DSA）

ML-DSA [78] は CRYSTALS-Dilithium に基づく署名方式である。ML-KEM と同じように，2 のべき数 $n = 256$ に対し $R := \mathbb{Z}[X]/(X^n + 1)$ を基本環とし，素数 $q = 8380417$ に対し $R_q := R/qR = \mathbb{Z}_q[X]/(X^n + 1)$ をその剰余環とする。階数パラメータ $k \in \{2, 3, 4\}$ に対し，ML-DSA の安全性は $\mathbb{Z}_q$ 加群 $R_q^k \simeq (\mathbb{Z}_q^n)^k$ 上の LWE 問題の計算量困難性に依存する。また，ML-KEM と同様に，R_q における乗算を高速化するために，NTT 変換を利用する。ここでは，主に ML-DSA の構成と処理概要について説明する。

3.3.2.1 ML-DSA における NTT 変換

ML-DSA では, 2 のべき数 $n = 2^8 = 256$ と素数 $q = 2^{23} - 2^{13} + 1 = 8380417$ で定まる剰余環 $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ を用いる（ML-DSA の暗号パラメータについては，後述の 3.3.2.3 節を参照)。これらの暗号パラメータの組 (n, q) において，$\mathbb{Z}_q^*$ は位数 $q - 1 = 2^{13} \cdot 1023$ の巡回群である。ML-DSA では，$\mathbb{Z}_q$ における 1 の原始 512 乗根 $\zeta := 1753 \bmod q$ をとる。このとき，多項式環 $\mathbb{Z}_q[X]$ において，$X^n + 1$ は次のように n 個の 1 次式の積に分解できる。

$$X^n + 1 = \prod_{i=0}^{n-1} \left(X - \zeta^{2i+1}\right) = \prod_{i=0}^{n-1} \left(X - \zeta^{2\mathsf{BitRev}_8(i)+1}\right) \in \mathbb{Z}_q[X].$$

ただし，$\mathsf{BitRev}_8(i)$ は符号なし 8 ビット整数 i のビット逆順整数とし，ML-DSA ではこの順序を利用する。具体的には，各 $i = 0, 1, \ldots, n-1$ に対し $\zeta_i := \zeta^{2\mathsf{BitRev}_8(i)+1}$ とおき，環としての同型

$$\mathsf{NTT} : R_q \simeq \bigoplus_{i=0}^{n-1} \mathbb{Z}_q[X]/(X - \zeta_i) \simeq \bigoplus_{i=0}^{n-1} \mathbb{Z}_q =: T_q, \quad f \mapsto \widehat{f} := (f(\zeta_0), f(\zeta_1), \ldots, f(\zeta_{n-1}))$$

を用いて，R_q における乗算を効率的に行う。

3.3.2.2 ML-DSA の構成と処理概要

加群 $R_q^k \simeq (\mathbb{Z}_q^n)^k$ 上の LWE 問題に基づく ML-DSA は，下記のアルゴリズム群で構成される。ただし，ML-DSA の処理概要とその原理が分かるように，簡略化した形で各アルゴリズムの処理を説明する。

■ML-DSA 鍵生成　鍵生成アルゴリズム（FIPS 204, Algorithm 6）では，乱数 ξ を入力として，公開鍵 $\mathtt{pk}$ と秘密鍵 $\mathtt{sk}$ を次のように出力する（ただし，ℓ は次元パラメータとする)。

- 入力：乱数 ξ
- 出力：公開鍵 $\mathtt{pk}$ と秘密鍵 $\mathtt{sk}$
  1. $(\rho, \rho', K) = \mathsf{H}(\xi)$：ハッシュ関数 H で乱数 ξ から 3 つのデータの組 (ρ, ρ', K) を一意的に生成
  2. $\widehat{\mathbf{A}} = \mathsf{ExpandA}(\rho) \in (T_q)^{k \times \ell}$：擬似ランダムな行列 $\mathbf{A} \in (R_q)^{k \times \ell}$ を生成し，その NTT 表現を $\widehat{\mathbf{A}}$ を計算
  3. $(\mathbf{s}_1, \mathbf{s}_2) = \mathsf{ExpandS}(\rho') \in S_\eta^\ell \times S_\eta^k$：$S_\eta$ はすべての係数が $[-\eta, \eta]$ 内の R の元全体の集合（例：$\eta \in \{2, 4\}$）
  4. $\mathbf{t} = \mathsf{NTT}^{-1}\left(\widehat{\mathbf{A}} \circ \mathsf{NTT}(\mathbf{s}_1)\right) + \mathbf{s}_2 = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \in R_q^k$：NTT 変換を利用して計算
  5. $(\mathbf{t}_1, \mathbf{t}_0) = \mathsf{Power2Round}(\mathbf{t}) \in R_q^k \times R_q^k$：$\mathbf{t} \in R_q^k$ を上位と下位ビットに分割
  6. $\mathtt{pk} = (\rho, \mathbf{t}_1)$ とおき，そのハッシュ値 $tr = \mathsf{H}(\mathtt{pk})$ を計算

7. $\mathtt{pk}$ と $\mathtt{sk} = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$ を出力

ステップ 2 において，ExpandA 関数（FIPS 204, Algorithm 32）は，乱数シード ρ から擬似ランダムな $\mathbf{A}$ を生成し，その NTT 表現 $\widehat{A}$ を計算する。ステップ 3 において，ExpandS 関数（FIPS 204, Algorithm 33）は，棄却サンプリングを用いてある範囲 $[-\eta, \eta]$ 内の係数をもつ R の元の組を生成する（$\eta \in \{2, 4\}$）。ステップ 5 において，Power2Round 関数（FIPS 204, Algorithm 35）を用いて，$\mathbf{t} \in R_q^k$ の各成分のすべての係数を上位と下位のビットに分割する。

鍵生成アルゴリズムにおいて，本質的に公開鍵 $\mathtt{pk}$ は $(\mathbf{A}, \mathbf{t})$ に対応し，その公開鍵に関する付属情報をいくつか含むが秘密鍵 $\mathtt{sk}$ は $(\mathbf{s}_1, \mathbf{s}_2)$ に対応する。公開鍵と秘密鍵の間に，R_q^k 上の LWE 関係式 $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$ が成り立つ。これより，公開鍵から秘密鍵を見つけるのは R_q^k 上の探索 LWE 問題で，適切な暗号パラメータ（後述の 3.3.2.3 節を参照）を利用した場合，その LWE 問題を解くのは計算量的に非常に困難である。

■ML-DSA 署名生成　署名生成アルゴリズム（FIPS 204, Algorithm 7）では，秘密鍵 $\mathtt{sk}$ と平文 M' を入力として，平文に対応する署名 σ を次のように出力する。

- 入力：秘密鍵 $\mathtt{sk} = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$ と平文 M'
- 出力：署名 σ
  1. $\widehat{\mathbf{s}}_1 = \mathsf{NTT}(\mathbf{s}_1) \in T_q^\ell, \widehat{\mathbf{s}}_2 = \mathsf{NTT}(\mathbf{s}_2) \in T_q^k, \widehat{\mathbf{t}}_0 = \mathsf{NTT}(\mathbf{t}_0) \in T_q^k$：NTT 表現を計算
  2. $\widehat{\mathbf{A}} = \mathsf{ExpandA}(\rho) \in (T_q)^{k \times \ell}$：$\rho$ から $\widehat{\mathbf{A}}$ を復元
  3. $\mu = \mathsf{H}(tr \| M')$：秘密鍵の一部 tr と平文 M' から定まるハッシュ値
  4. 次を繰り返す：
     (a) $\mathbf{y} = (\mathbf{y}[i])_{i=0}^{\ell} \in R_q^\ell$：各 $\mathbf{y}[i] \in R_q$ の各 $\mathbb{Z}_q$ 係数をある範囲で擬似ランダムにサンプル（十分小さい）
     (b) $\mathbf{w} = \mathsf{NTT}^{-1}\left(\widehat{\mathbf{A}} \circ \mathsf{NTT}(\mathbf{y})\right) = \mathbf{A}\mathbf{y} \in R_q^k$：NTT 変換を利用
     (c) $\mathbf{w}_1 = \mathsf{HighBits}(\mathbf{w}) \in R_q^k$：$\mathbf{w}$ の各成分の上位ビット
     (d) $\tilde{c} = \mathsf{H}(\mu \| \mathbf{w}_1)$
     (e) $c = \mathsf{SampleInBall}(\tilde{c}) \in R_q$：各係数を $\{-1, 0, 1\}$ からサンプルする（十分小さい）
     (f) $\widehat{c} = \mathsf{NTT}(c) \in T_q$
     (g) $c\mathbf{s}_1 = \mathsf{NTT}^{-1}(\widehat{c} \circ \widehat{\mathbf{s}}_1) \in R_q^\ell, c\mathbf{s}_2 = \mathsf{NTT}^{-1}(\widehat{c} \circ \widehat{\mathbf{s}}_2) \in R_q^k$：NTT 空間の乗算を利用
     (h) $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1 \in R_q^\ell$
     (i) $\mathbf{r}_0 = \mathsf{LowBits}(\mathbf{w} - c\mathbf{s}_2) \in R_q^k$：$\mathbf{w} - c\mathbf{s}_2$ の各成分の下位ビット
     (j) $\mathbf{z}$ と $\mathbf{r}_0$ のすべての $\mathbb{Z}_q$ 係数が十分小さい場合，次の処理を行う：
        i. $c\mathbf{t}_0 = \mathrm{NTT}^{-1}(\widehat{c} \circ \widehat{\mathbf{t}}_0) \in R_q^k$：NTT 空間の乗算を利用
        ii. $\mathbf{h} = \mathsf{MakeHint}\left(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0\right)$：長さ k の不一致真理値ベクトル
        $c\mathbf{t}_0$ のすべての $\mathbb{Z}_q$ 係数が十分小さく，かつ $\mathbf{h}$ 内の 1 の個数が十分少ないとき，ステップ 5 に進む
  5. $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$ を出力

ステップ 4 (e) において，乱数 $\tilde{c}$ を引数とする SampleInBall 関数（FIPS 204, Algorithm 29）で，すべての $\mathbb{Z}_q$ 係数を $\{-1, 0, 1\}$ からサンプルした多項式 $c \in R_q$ を生成する（ただし，係数ベクトルのハミング重みは 64 以下）。ステップ 4 (f) ii において，MakeHint 関数（FIPS 204, Algorithm 39）は，$\mathsf{HighBits}(\mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0)$ と $\mathsf{HighBits}(\mathbf{w} - c\mathbf{s}_2)$ の $\mathbb{Z}_q$ 係数の不一致真理値による長さ k のベクトル $\mathbf{h}$ を計算する。次の署名検証時で $\mathbf{w}_1$ を復元するために $\mathbf{h}$ を用いる。

署名生成アルゴリズムにおいて，ステップ 4 が主処理で，すべての $\mathbb{Z}_q$ 係数が十分小さい $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1 \in R_q^\ell$ を見つけるまで $\mathbf{y} \in R_q^\ell$ を取り直す。具体的には，擬似ランダムにサンプルしたすべての $\mathbb{Z}_q$ 係数が十分小さい $\mathbf{y} \in R_q^\ell$ から，

コミットメント $\mathbf{w}_1$ を生成し，$\mathbf{w}_1$ と μ から定まるハッシュ値であるチャレンジ $\tilde{c}$ を求める。また，レスポンスとして，すべての $\mathbb{Z}_q$ 係数が十分小さい $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$ を生成する。チャレンジ $\tilde{c}$，レスポンス $\mathbf{z}$，コミットメント $\mathbf{w}_1$ のヒント $\mathbf{h}$ の 3 つの組 $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$ を平文に対応する署名とする。

■ML-DSA 署名検証　署名検証アルゴリズム（FIPS 204, Algorithm 8）では，公開鍵 $\mathtt{pk} = (\rho, \mathbf{t}_1)$ と署名 $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$ 付きの平文 M' を入力として，署名検証の結果を次のように真偽値で出力する。

- 入力：公開鍵 $\mathtt{pk} = (\rho, \mathbf{t}_1)$，署名 $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$ 付きの平文 M'
- 出力：真偽値
  1. $\widehat{\mathbf{A}} = \mathsf{ExpandA}(\rho)$：$\rho$ から $\widehat{\mathbf{A}}$ を復元
  2. $tr = \mathsf{H}(\mathtt{pk})$：pk のハッシュ値
  3. $\mu = \mathsf{H}(tr \| M')$：秘密鍵の一部 tr と平文 M' から定まるハッシュ値
  4. $c' = \mathsf{SampleInBall}(\tilde{c}) \in R_q$：各係数を $\{-1, 0, 1\}$ からサンプルする
  5. $\mathbf{w}'_{\mathrm{Approx}} = \mathsf{NTT}^{-1}\left(\widehat{\mathbf{A}} \circ \mathsf{NTT}(\mathbf{z}) - \mathsf{NTT}(c') \circ \mathsf{NTT}(\mathbf{t}_1 \cdot 2^d)\right) = \mathbf{A}\mathbf{z} - c'\mathbf{t}_1 \cdot 2^d \in R_q^\ell$
     （ただし，d は上位と下位ビットを分割する閾値）
  6. $\mathbf{w}'_1 = \mathsf{UseHint}(\mathbf{h}, \mathbf{w}'_{\mathrm{Approx}})$：署名生成時のコミットメントを復元
  7. $\tilde{c}' = \mathsf{H}(\mu \| \mathbf{w}'_1)$：$\mu$ と $\mathbf{w}'_1$ から定まるハッシュ値
  8. $\mathbf{z}$ のすべての $\mathbb{Z}_q$ 係数が十分小さく，かつ $\tilde{c} = \tilde{c}'$ のとき，「真」を出力。それ以外は，「偽」を出力

ステップ 6 において，UseHint 関数（FIPS 204, Algorithm 40）で，$\mathbf{w}'_{\mathrm{Approx}}$ が $\mathbf{w}$ に十分近いとき，ヒント $\mathbf{h}$ を元に署名生成時のコミットメント $\mathbf{w}_1$ を復元する（つまり，$\mathbf{w}'_1 = \mathbf{w}_1$）。具体的には，$\sigma$ が正当な署名であれば，$c' = c$ で $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$ なので，LWE 関係式 $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$ と $\mathbf{t}_1 \cdot 2^d \approx \mathbf{t}$（$\mathbf{t}_1$ は $\mathbf{t}$ の上位ビット）より

$$\begin{aligned}\mathbf{w}'_{\mathrm{Approx}} = \mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d &= \mathbf{A}\mathbf{y} + c\mathbf{A}\mathbf{s}_1 - c\mathbf{t}_1 \cdot 2^d \\ &= \mathbf{w} + c(\mathbf{t} - \mathbf{s}_2) - c\mathbf{t}_1 \cdot 2^d \approx \mathbf{w} - c\mathbf{s}_2 \approx \mathbf{w}\end{aligned}$$

が成り立つ（$c\mathbf{s}_2 \in R_q^k$ のすべての $\mathbb{Z}_q$ 係数は十分小さいことに注意）。このとき，ステップ 7 で $\tilde{c}' = \tilde{c}$ となり検証に成功する。一方，平文 M' が改竄または署名 σ が偽造された場合は，非常に高い確率で $\tilde{c} \neq \tilde{c}'$ となり，検証に失敗する。

3.3.2.3　暗号パラメータ

ML-DSA における主な暗号パラメータと対応する鍵や署名のサイズと安全性レベルは以下である。具体的には，LWE の次元 $n = 256$ と剰余素数 $q = 8380417$ は ML-DSA-44, -65, -87 の 3 種類の暗号パラメータで共通であるが，主に公開鍵行列 $\mathbf{A} \in (R_q)^{k \times \ell}$ のサイズ (k, ℓ) により安全性レベルが異なる（特に，ML-DSA のパラメータ名は，(k, ℓ) により名づけられている）。

	暗号パラメータ			サイズ（単位：バイト）			安全性
	n	q	(k, ℓ)	秘密鍵	公開鍵	署名	レベル
ML-DSA-44	256	8380417	(4, 4)	2560	1312	2420	レベル 2
ML-DSA-65	256	8380417	(6, 5)	4032	1952	3309	レベル 3
ML-DSA-87	256	8380417	(8, 7)	4896	2592	4627	レベル 5

3.3.2.4 CRYSTALS-Dilithium との違い

- CRYSTALS-Dilithum の version 3.1 と round 3 version との違いは，安全性を確保するために，署名アルゴリズム内の秘密ランダムシード ρ' とメッセージ表現 μ の長さが 384 から 512 ビットへの増大である。加えて，公開鍵のハッシュに関する変数 tr のサイズを 384 から 256 ビットに減少させる一方，鍵生成において変数 ζ を ρ' に再ラベル付けし，そのサイズを 256 から 512 ビットに増大させている。
- MD-DSA と CRYSTALS-Dilithum の version 3.1 との違いについて，ML-DSA では tr の長さを 512 ビットに増やし，ML-DSA-65 と ML-DSA-87 のパラメータ設定それぞれで $\tilde{c}$ の長さを 384 と 512 ビットに増大している。CRYSTALS-Dilithum version 3.1 では，デフォルトの署名アルゴリズムは署名者の秘密鍵とメッセージから疑似ランダム生成された ρ' について確定的で，optional version では ρ' は 512 ビットのランダム列としてサンプリングされる。一方，ML-DSA では，ρ' は署名者の秘密鍵，メッセージ，と Approved RBG から生成された 256 ビットの文字列 rnd から生成される。また，ML-DSA 標準では，rnd が 256 ビットの定数文字列である optional deteministic version を許可している。

3.3.3 FALCON

歴史: FALCON は 2017 年 11 月の NIST PQC 標準化プロジェクトの公募に Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang の 10 名を開発者として公表された [50]。その後修正が加えられ，現在の最新版は 2020 年 10 月に公開された v1.2[51] である。以下の記述はこの仕様書に従う。

参照 URL: 開発者による公式ページ `https://falcon-sign.info/` を参照した。

設計原理: FALCON は多項式 $x^n + 1, n = 2^k$ により定義される NTRU 格子上の SIS 問題の困難性を安全性の根拠とした格子ベースの署名方式であり，形式的には Gentry ら [52] の Hash-and-Sign 型の格子ベース署名をひな型としている。高速フーリエサンプリングを用いるため，定義多項式の次数を 2^k の形としていることからパラメータ選択の自由度に制限があり，NIST PQC 標準化プロジェクトの提案方式では安全性レベル 1 および 5 のパラメータセットのみが提案されている。

アルゴリズムの詳細: 表 3.2，3.3，3.4 に，Gentry ら [52] の Hash-and-Sign 型格子ベース署名と FALCON の鍵生成，署名生成，署名検証関数を並置する。

パブリックパラメータは以下で与えられる。

- n, q: 環を定義する多項式 $\phi(x) = x^n + 1$ と法 q で，演算は $\mathbb{Z}_q[x]/(\phi)$ で行われる。
- σ: 離散ガウス分布の大きさを指定する。
- β: 有効な署名のノルムの上限を指定する。

アルゴリズム中で用いられるサブルーチンのうち，主なものを列挙する。

- $\mathsf{FFT}(f), \mathsf{invFFT}(\boldsymbol{s})$: 多項式 $f \in \mathbb{R}[x]/(\phi)$ に対して，そのフーリエ変換 $\mathsf{FFT}(f)$ を n 次元ベクトル $(f(\zeta_k))_{k=0,\ldots,n-1}$ で定義する[*2]。ただし，$\zeta_k := \exp((2k+1)\pi i/n)$。逆演算を $\mathsf{invFFT} : \mathbb{R}^n \to \mathbb{R}[x]/(\phi)$ で

[*2] 数式上は差が無いが高速フーリエ変換による実装を行ったサブルーチンも同じ記号で示すため，Fast Fourier の意味で FFT と名づけられている。

示す。変換，逆変換ともに標準的な高速フーリエ変換の手法が利用可能である。コンピュータ上での計算には浮動小数点演算を用いるため，実行環境ごとに差が出ないように IEEE754 で規定される浮動小数点の表現と演算を用いることが指定されている。

多項式を成分とするベクトル，行列に対しても FFT は成分ごとのフーリエ変換と定義し，invFFT も適切な切り分けにより実数成分の行列，ベクトルから多項式成分の行列，ベクトルへ変換するものとする。

また，演算 $\mathsf{FFT}(f) \odot \mathsf{FFT}(g)$ を成分ごとの積と定義する。FFT 表現での多項式の積 $\mathsf{FFT}(fg)$ の計算に対応する。

- $\mathsf{HashToPoint}(\mathsf{str}, q, n)$: ビット列 str を多項式 $c \in \mathbb{Z}_q[x]/(\phi)$ に SHAKE256 ハッシュ関数を用いて写像する。
- Compress, Decompress: 多項式 $s \in \mathbb{Z}[x]$ を文字列に変換する関数とその逆関数とする。
- $\mathsf{NTRUGen}(\phi, q)$: 計算が行われる環 $\mathbb{Z}_q[x]/(\phi)$ を指定するパラメータを入力とし，秘密鍵 $\hat{B}$ の元となる多項式 f, g, F, G を出力する。このとき，f, g は係数が離散ガウス分布の n 次多項式，F, G は $fG - gF \equiv q \mod \phi$ を満たすように計算される。

表 3.2: Hash-and-Sign 型格子ベース署名および FALCON における鍵生成関数の比較

	Gentry らの格子ベース署名 [52, Sect. 7.1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	FALCON[51, Algorithm 4] $\mathsf{KeyGen}(\phi, q) \to (pk, sk)$
1:	$BA \equiv 0 \pmod q$ を満たす行列の組 (A, B) を生成 B: 成分の小さい行列 A: ランダム行列	$f, g, F, G \leftarrow \mathsf{NTRUGen}(\phi, q)$ $B \leftarrow \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$ $\hat{B} \leftarrow \mathsf{FFT}(B)$ $G \leftarrow \hat{B} \times \hat{B}^*$ $T \leftarrow \mathsf{ffLDL}^*(G)$ **for** each leaf leaf of T **do** $\quad \mathsf{leaf.value} \leftarrow \sigma / \sqrt{\mathsf{leaf.value}}$ $h \leftarrow gf^{-1} \bmod q$
retrun	$pk = A, sk = B$	$pk = h, sk = (\hat{B}, T)$

NTRU 型暗号の秘密鍵 (f, g) のうち，f は環 $\mathbb{Z}_q/(\phi)$ の中で逆元を持つため，適当な $F, G \in \mathbb{Z}[x]$ を用いて

$$fG - gF = q \bmod \phi \tag{3.3}$$

と書くことができる。この関係式と公開鍵 $h = f^{-1}g$ を Hash-and-Sign フレームワーク [52] における行列 A, B と捉えると，

$$A = \begin{bmatrix} 1 \\ h \end{bmatrix},\ B = \begin{bmatrix} g & -f \\ G & -F \end{bmatrix} \tag{3.4}$$

と表現することができる。このとき，行列 A は多項式 h の情報のみで表現可能であるため，$pk = h$ となる。

また，署名の生成には $\boldsymbol{s}A \equiv H(m)$ を満たす短いベクトル $\boldsymbol{s}$ を生成する必要があり，効率化のため Ducas-Prest[39] の高速フーリエサンプリングを用いる。サンプリングアルゴリズムに必要な情報が B の FFT 表現

$$\mathsf{FFT}(B) = \begin{bmatrix} \mathsf{FFT}(g) & \mathsf{FFT}(-f) \\ \mathsf{FFT}(G) & \mathsf{FFT}(-F) \end{bmatrix} \tag{3.5}$$

およびそれを元にした LDL 木と呼ばれる木構造 T である。木の中には $\hat{B}$ のグラム行列 $G = \hat{B} \times \hat{B}^\star$ の[*3] LDL 分解における L の情報が格納され，それを用いて Babai の最近平面アルゴリズムの高速化および離散ガウス分布の高速なサンプリングが可能となる。サンプリングを行うための付加情報として，木の全ての葉にある値を leaf.value から $\sigma/\sqrt{\mathsf{leaf.value}}$ に書き換えることで鍵生成が完了する。

表 3.3: Hash-and-Sign 型格子ベース署名および FALCON における署名生成関数の比較

	Gentry らの格子ベース署名 [52, Sect. 7.1] $\mathsf{Sign}(sk = (\hat{B}, T), m \in \{0,1\}^*) \to \sigma$	FALCON[51, Algorithm 10] $\mathsf{Sign}(sk = (\hat{B}, T), m \in \{0,1\}^*, \lfloor \beta^2 \rfloor) \to \sigma$
1:	$\boldsymbol{c} \leftarrow H(m)$ //平文のハッシュ値をベクトル化	$r \leftarrow \{0,1\}^{320}$ $c \leftarrow \mathsf{HashToPoint}(r \| m, q, n)$ $\hat{\boldsymbol{t}} \leftarrow \left(-\frac{1}{q}\mathsf{FFT}(c) \odot \mathsf{FFT}(F), \frac{1}{q}\mathsf{FFT}(c) \odot \mathsf{FFT}(f)\right)$
2:	T を使い，$\boldsymbol{s}A \equiv \boldsymbol{c} \pmod q$ を 満たすベクトル $\boldsymbol{s}$ をサンプリング	**do** 　**do** 　　$\boldsymbol{z} \leftarrow \mathsf{ffSampling}_n(\hat{\boldsymbol{t}}, T)$ 　　$\hat{\boldsymbol{s}} \leftarrow (\hat{\boldsymbol{t}} - \hat{\boldsymbol{z}})\hat{B}$ 　**while** $\|\boldsymbol{s}\|^2 > \lfloor \beta^2 \rfloor$ 　$(s_1, s_2) \leftarrow \mathsf{invFFT}(\hat{\boldsymbol{s}})$ 　$\mathsf{s} \leftarrow \mathsf{Compress}(s_2, 8 \cdot \mathsf{sbytelen} - 328)$ **while** $(\mathsf{s} = \perp)$
return	$\sigma = \boldsymbol{s}$	$\sigma = (r, \mathsf{s})$

表 3.3 の署名生成関数の説明を記述する。平文にランダムビット r を結合した後，HashToPoint 関数で多項式 $c \in \mathbb{Z}_q/(\phi)$ を出力する。関係式 (3.3), (3.4) より，ベクトル $\hat{\boldsymbol{t}}$ は $(\mathsf{FFT}(c), \mathsf{FFT}(0))\hat{B}^{-1}$ と等しい事がわかる。これらの情報を用いて，署名ベクトルのサンプリングを行う。

関数 $\mathsf{ffSampling}_n$ は，離散ガウス分布のサンプリングを行い，FFT 表現で出力するサブルーチンである。具体的には，整数ベクトル $\boldsymbol{z} \in \mathbb{Z}^{2n}$ を，$\boldsymbol{t} = [c, 0]B^{-1}$ を中心として $\exp(-\|(\boldsymbol{z} - \boldsymbol{t})B\|^2/2\sigma^2)$ に比例した確率でサンプリングを行う。実装の効率化のため，実際には近似を行っている [51, Sect. 3.9.1, 3.9.2]。このとき，$(\boldsymbol{t} - \boldsymbol{z})B$ は原点を中心とした集合

$$\boldsymbol{t} + \Lambda(B) = \{(c, 0) + x \in (\mathbb{Z}[x]/(\phi))^2 : x \in \Lambda(B)\}$$

上の離散ガウス分布となるため，$\boldsymbol{s}$ は短く，かつ

$$\boldsymbol{s}A \equiv ([c, 0]B^{-1} - \boldsymbol{z})BA \equiv [c, 0] \begin{bmatrix} 1 \\ h \end{bmatrix} = c \text{ in } \mathbb{Z}_q[x]/(\phi)$$

が成り立つ。このとき，$\boldsymbol{s}A = c$ の関係から $s_1 + s_2 h = c$ が成り立つ。この関係式が署名の検証時に用いられる。

サンプリングされた $\hat{\boldsymbol{s}}$ が $\|\hat{\boldsymbol{s}}\|^2 \le \lfloor \beta^2 \rfloor$ を満たしていれば invFFT により通常空間の表現に戻し，Compress 関数を用いて圧縮された文字列 s を生成し，ハッシュ関数のシード r とともに署名とする。

表 3.4 の署名検証関数の説明を記述する。平文，ハッシュ関数のシード値，署名文字列から各要素を復元し，$s_1 = c - s_2 h$ を計算する。署名が正しく生成されていれば $\boldsymbol{s}A = c$ の関係から，s_1 は短い元となるはずなので，$\|(s_1, s_2)\|^2 \le \lfloor \beta^2 \rfloor$ が満たされ検証が完了する。

[*3] $B^\star$ は体 $\mathbb{Q}[x]/(\phi)$ におけるエルミート共役。詳細は [51, p.23]

表 3.4: Hash-and-Sign 型格子ベース署名および FALCON における署名検証関数の比較

	Gentry らの格子ベース署名 [52, Sect. 7.1]	FALCON[51, Algorithm 16]
	$\mathsf{Vrfy}(m \in \{0,1\}^*, \sigma = s, pk = A)$	$\mathsf{Vrfy}(m \in \{0,1\}^*, \sigma = (r, \mathsf{s}), pk = h, \lfloor \beta^2 \rfloor)$
1:	$\boldsymbol{t} \leftarrow H(m)$	$c \leftarrow \mathsf{HashToPoint}(r \| m, q, n)$
2:	**if** $\boldsymbol{t} - \boldsymbol{s}A \equiv 0 \pmod q$	$s_2 \leftarrow \mathsf{Decompress}(\mathsf{s}, 8 \cdot \mathsf{sbytelen} - 328)$
	AND $\boldsymbol{s}$ が短い **then return** accept	**if** $(s_2 = \perp)$
		return reject
		$s_1 \leftarrow c - s_2 h \bmod q$
		if $\|(s_1, s_2)\|^2 \leq \lfloor \beta^2 \rfloor$
		return accept
		else
		return reject

安全性とパラメータ: FALCON の安全性は $\phi(x) = x^n + 1, q = 12289$ を定義多項式とする NTRU 格子上の計算問題として表現される。鍵復元の困難性は SIS 問題，署名偽造はターゲットベクトルに近い点を求める計算問題として定式化される。後者は Kannan の埋め込みにより短いベクトルを求める計算問題に変換される。セキュリティに関わるパラメータは n, q, σ, β の 4 個で，n は格子の次元を表し，大きく取ることで安全性が上がるが処理速度が低下する。q は環を定義するための法で，大きくとることでノイズ耐性が上がるが格子が疎になり安全性が低下する。σ はガウス分布の大きさを指定するパラメータで，大きくとることで安全性が上がるがエラー率が上がる。β は署名ベクトルの長さの上限を指定するパラメータで，大きくとることで署名生成時のやり直し回数が下がるが，安全性が低下する。

具体的な困難性の評価およびパラメータ設定は，SIS 問題を BKZ アルゴリズムを用いて解いた場合の Core-SVP 計算量により導出している。

表 3.5: FALCON のパラメータ [51, Table 3.3], [5, Table 8] 公開鍵，秘密鍵，署名サイズの単位はそれぞれ Byte である。

$(n, q, \sigma, \lfloor \beta^2 \rfloor)$	安全性レベル	公開鍵サイズ	秘密鍵サイズ [*4]	署名サイズ
$(\ 512, 12289, 165.736617183, 34034726)$	レベル 1	897	7,553	666
$(1024, 12289, 168.388571447, 70265242)$	レベル 5	1,793	13,953	1,280

変種: 実装の複雑さによるサイドチャネル攻撃からの防御，セキュリティパラメータの多様性確保などを目的とした改良が多数提案されている。

特に，鍵生成と署名生成における離散ガウス分布生成の改良が多い。一例として，ガウス分布生成の演算を浮動小数点から整数演算に変更した Zalcon[49]，ガウス分布の代わりに中心二項分布とした Peregrine [92]，実装が複雑な高速フーリエサンプリングを環上の CVP アルゴリズムをベースとしたより単純なものに置き換えた Mitaka [43] などが存在する。Peregrine は韓国の耐量子計算機暗号公募 KpqC[104] へと提出されているものの，同じ秘密鍵から生成した署名に対する統計的攻撃法による実時間での鍵復元手法が知られている [63]。

[*4] 秘密鍵サイズは仕様書には掲載されていないが，NIST の第 3 ラウンド報告レポート [5, Sect. D] を参照した。

また，Falcon では環の定義多項式が $\phi(x) = x^n + 1, n = 2^k$ の形に制限されていることから安全性レベル 1,5 のパラメータのみが提案されていたが，NTRU 格子をモジュール格子とすることでパラメータ設定の多様性を確保した Mod Falcon [25] も存在する。

上記 Mitaka 内で用いられる離散ガウス分布生成アルゴリズムは実装が比較的単純である半面，生成された鍵および署名ベクトルのノルムが大きく鍵長と署名長が長いという欠点があった。近年では Antrag[77] が両者の中間的な手法として，FFT 表現でのサンプリングを通じて鍵生成における離散ガウス分布のノルムを下げ鍵長と署名長を短くする戦略を取っている。また，SOLMAE[59] も同様のサンプリング手法を用いた上で，エラーベクトルの圧縮表現などを用いて署名長を短縮する技術 [45] と組み合わせ KpqC へと提案されている。

補足情報: 2022 年に NIST より標準化がアナウンスされ，将来的に NIST FIPS 206 (FN-DSA) として出版される予定であるが，他の格子暗号方式 (FIPS 203 および 204) と比較して発表が遅れている。これは基準となる仕様書版 [51] からの修正箇所 [85] が多いことが原因であると考えられる。

鍵生成および署名生成アルゴリズムの中で浮動小数点演算が用いられているため実行環境ごとの結果の不安定性，定数時間での実装が難しいことによるタイミング攻撃の可能性がある。対策として固定小数点を用いた実装への変更が検討されている [85, p. 13]。

また，ML-DSA と比較して beyond unforgeability [29] の性質を完全には持たないことから，署名生成におけるハッシュ値の計算方法の変更が検討されている（[85, p. 15] および [42] を参照）。

3.4 格子に基づく暗号技術に関するまとめ

格子に基づく暗号技術は，LWE 問題，Ring-LWE 問題，NTRU 問題を安全性の根拠とする方式をはじめ，これまで数多く提案されており，米国 NIST PQC 標準化プロジェクトで提案された暗号技術としては最も多くの暗号がこのカテゴリーに分類されている。

この米国 NIST PQC 標準化プロジェクトを通じて 2022 年 7 月に CRYSTALS-Kyber が標準的な暗号方式として，CRYSTALS-Dilithium および FALCON が標準的な署名方式として選定され，CRYSTALS-Kyber と CRYSTALS-Dilithium については，2024 年 8 月に FIPS 203, FIPS 204 として公開されている [79, 78]。また，CRYSTALS-Kyber と CRYSTALS-Dilithium は 2022 年 9 月に米国国家安全保障局の Commercial National Security Algorithm Suite 2.0 (CNSA2.0) にも選定されている [1]。NIST PQC 標準化プロジェクトの選考プロセスから漏れた方式の中でも，米国以外の公的機関において推奨暗号とされているものが存在する。一例として，FrodoKEM が 2020 年 8 月よりドイツ情報セキュリティ庁 (BSI) の推奨暗号に [91]，2022 年 1 月にはオランダ通信・安全委員会 (NLNCSA) により最も安全な暗号の例として推奨されている [11]。Google 社の Chrome ブラウザには，TLS レイヤーの性能試験目的で搭載された耐量子計算機暗号プロトコル CECPQ1[20] および CECPQ2[86] にそれぞれ NewHope の USENIX 発表バージョン [10] と NTRU が実装されていたが，2023 年 1 月現在ではともに削除されている。IBM 製テープドライブのプロトタイプとして，CRYSTALS-Kyber と CRYSTALS-Dilithium の組み合わせにより暗号化を行うものが制作されている [62]。DNS サーバの一種である PowerDNS において，耐量子計算機性を実現する署名として FALCON のテスト用の実装が行われている [54]。オープンソースライブラリへの導入として，WireGuard VPN protocol への SABER の実装 [58]，WolfSSL への CRYSTALS-Kyber，FALCON の実装 [100]，OpenSSH への Streamlined NTRU Prime の実装 [80] などが存在する他，Open Quantum Safe (OQS) プロジェクトによる liboqs ライブラリには暗号化・鍵交換の方式として CRYSTALS-Kyber，NTRU，SABER，FALCON，FrodoKEM，NTRU-Prime が，署名方式として CRYSTALS-Dilithium と FALCON が実装されている [90]。このように格子に基づく暗号技術の社会実装が徐々に進

みつつある。特に，標準化が先行する CRYSTALS-Kyber，CRYSTALS-Dilithium に対するサイドチャネル攻撃とその対策としてマスキング実装が検討されている [93, 97, 26]。

格子に基づく暗号技術の安全性の根拠となる問題としては，先に挙げた LWE 問題，Ring-LWE 問題，NTRU 問題以外にも Compact LWE 問題，Module-LWE 問題，LWR 問題，BDD 問題，SIS 問題他，多くのバリエーションが存在している。一般的な格子問題を解く手法としては，LLL アルゴリズム，BKZ アルゴリズムがよく知られており，LWE 問題については更に SIS 問題や BDD 問題に還元する解析手法が知られている。SVP や LWE/NTRU などの格子問題の解析やそれらの求解アルゴリズムに関する最新研究については [19, 15, 47, 65, 89, 74, 30, 95, 33, 75, 84, 22, 63, 17, 28] を参照。近年，新しい格子問題として格子同型問題 [41] が提案された。（格子同型問題の性質については [14] を参照。）また，格子同型問題の困難性を安全性の根拠とする署名方式 HAWK[38] は，NIST PQC 標準化プロジェクトにおける署名方式の追加公募において，格子に基づく方式の中で第 2 ラウンドにおいて進むことが許された方式である（2024 年 10 月時点）。さらに，量子紛失 LWE サンプリング [34] や，格子問題に対する量子アルゴリズムに関する研究 [24, 27] も近年進展している。

格子問題の困難性をベースとした暗号方式で最初のものは，Ajtai[2] により 1996 年に行われた，SIS 問題が格子問題の最悪時と同等かそれ以上に困難であることの証明およびそれを用いた暗号学的ハッシュ関数の構成である。また，1997 年には Ajtai と Dwork[4] により，unique SVP の最悪困難性を安全性の根拠とした公開鍵暗号が提案されている。この公開鍵暗号方式は翌年，Nguyen らによる解読実験 [76] により必要なパラメータが長大となり実用的でないことが明らかにされたものの，その後の格子に基づく暗号構成の基礎となっている。

1996 年に Hoffstein らによって提案された NTRU 暗号 [55][*5] は，発表当初安全性証明が付けられておらず，攻撃と修正が繰り返されていたが，2011 年 Stehlé ら [94] により方式が修正され，イデアル格子上の問題の困難性に還元可能なことが示されている。一方で，2016 年には subfield attack[6] のような体の構造を使って格子の次元を圧縮する攻撃も提案されており，暗号の構成のためには次元や法の大きさだけでなく，環・体の構造にも注意を払う必要がある。NTRU 格子上の署名方式のサイズ改良 [45]・トラップドア生成 [44] や，NTRU に対する鍵ミスマッチ攻撃の改良 [66]・NTRU 格子の簡約 [13] に関する最新の研究がある。

2005 年に Regev[87] により提案された LWE 問題は，論文発表と同時にそれを暗号の安全性根拠として保障する重要な三つの性質が示された。一つは問題の average-case to worst case reduction，つまりパラメータを固定した際，問題の (秘密ベクトル $\mathbf{s}$ に関する) 平均的な計算量が，最悪計算量 (難しいインスタンスを生成するような $\mathbf{s}$ の集合に対する計算量) と高々多項式倍の違いしか無いことであり，残りの二つは判定 LWE と探索 LWE の等価性，および量子アルゴリズムによる困難な格子問題への還元である。これらの定理を組み合わせることにより，Regev 自身により提案された公開鍵暗号を解読することが平均的に難しいことが示され，その後の様々な LWE ベース暗号の構成の基礎なった。LWE 格子問題への還元に関して，2013 年には古典計算機による還元も示されている [21]。

LWE 問題の欠点である鍵サイズの大きさを改善するため，2010 年には Lyubashevsky ら [69, 70] により Ring-LWE 問題が，2015 年には Langlois ら [61] により Module-LWE 問題が暗号化方式と同時に提案され，LWE 問題における関係と類似の，解読の平均的な困難さが証明されている。一方で，これらの変種とオリジナルの LWE 問題との関係性は自明ではなく，同程度の難しさを持つかどうかは未解決問題である。一般的に Ring(Module)-LWE 問題のインスタンスは LWE 問題のインスタンスとして書きなおすことができるため，LWE 問題は Ring(Module)-LWE 問題よりも困難であるという関係は自明であるが，逆の関係は知られていない。法 q が大きい場合には，Ring-LWE は Module-LWE よりも困難であることが知られている [8]。（Ring/Module LWE 問題の理論解析の最新研究について [98] を参照。）

[*5] 文献上は 1998 年の国際会議 ANTS だが，初出は CRYPTO1996 の Rump Session である。

実装時の問題として，離散 Gauss 分布を正確に生成することは難しいことが挙げられる。ノイズをある整数区間から一様分布として取った場合でも，格子問題へと量子帰着が可能であることが 2013 年に Döttling ら [37] により示された。この方向性の研究として，Bai ら [12] により提案された，理想的な Gauss 分布を用いた暗号方式とそれを近似的な分布に置き換えた方式の間での安全性の低下を Rényi エントロピーを用いて議論するものがある。

格子に基づく暗号技術は，耐量子計算機暗号としてだけでなく，完全準同型暗号や多重署名などの高機能な暗号方式に応用する研究も数多くある [16, 18, 82, 99, 36, 103, 81, 71, 23, 60, 35, 56, 72, 73]。

また，格子問題の計算機による具体的な求解に関して，2016 年より暗号解読コンテスト LWE Challenge[31] が開催されている。3.1 節に，2024 年 11 月現在の状況について記載した。特に 3.3 節で示された各暗号方式のパラメータから見ると，解が得られている値からは，大きな隔たりがみられる。格子に基づく暗号技術は，各方式毎にパラメータ設定手法に対する制約が異なっていることから，解読コンテストのサイズに基づく解読到達レベルを，具体的な暗号方式の安全性の根拠とすることは，難しいところではあるものの，古典計算機での解読困難性を測る上での検討の一つに値すると考えられる。（最新の BKZ の改良や LWE の解読計算量見積もりについては [96, 101] を参照）

格子に基づく暗号技術の安全性の根拠となる問題は，古典計算機・量子計算機のいずれにおいても現時点で効率的な解読手法は見つかっていないが，格子に基づく暗号技術は未だ研究途上にあり，今後も研究の進捗を注視する必要がある。

第 3 章の参照文献

[1] National Security Agency. Announcing the Commercial National Security Algorithm Suite 2.0. https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF. 2022-09. (2024-12-06 閲覧).

[2] M. Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). STOC. ACM, 1996, pp. 99–108.

[3] M. Ajtai. Generating Hard Instances of the Short Basis Problem. ICALP. Vol. 1644. Lecture Notes in Computer Science. Springer, 1999, pp. 1–9.

[4] M. Ajtai, Cynthia Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. STOC. ACM, 1997, pp. 284–293.

[5] G. Alagic et al. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8413, https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413-upd1.pdf. 2022-07.

[6] M. R. Albrecht, S. Bai, L. Ducas. A Subfield Lattice Attack on Overstretched NTRU Assumptions – Cryptanalysis of Some FHE and Graded Encoding Schemes. CRYPTO (1). Vol. 9814. Lecture Notes in Computer Science. Springer, 2016, pp. 153–178.

[7] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, T. Wunderer. Estimate All the {LWE, NTRU} Schemes! SCN. Vol. 11035. Lecture Notes in Computer Science. Springer, 2018, pp. 351–367.

[8] M. R. Albrecht, A. Deo. Large Modulus Ring-LWE $\geq$ Module-LWE. ASIACRYPT (1). Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 267–296.

[9] M. R. Albrecht, L. Ducas, G. Herold, E. Kirshanova, E. W. Postlethwaite, M. Stevens. The General Sieve Kernel and New Records in Lattice Reduction. EUROCRYPT (2). Vol. 11477. Lecture Notes in Computer Science. Springer, 2019, pp. 717–746.

[10] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe. Post-quantum Key Exchange - A New Hope. USENIX Security Symposium. USENIX Association, 2016, pp. 327–343.

[11] General intelligence and security service. Prepare for the threat of quantum computers. https://english.aivd.nl/publications/publications/2022/01/18/prepare-for-the-threat-of-quantumcomputers. 2022-01. (2024-03-04 閲覧).

[12] S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, R. Steinfeld. Improved Security Proofs in Lattice-Based Cryptography: Using the Rényi Divergence Rather than the Statistical Distance. J. Cryptol. Vol. 31, Num. 2 (2018), pp. 610–640.

[13] H. Bambury, P. Q. Nguyen. Improved Provable Reduction of NTRU and Hypercubic Lattices. PQCrypto (1). Vol. 14771. Lecture Notes in Computer Science. Springer, 2024, pp. 343–370.

[14] B. Bencina, A. Budroni, J.-J. Chi-Domínguez, M. Kulkarni. Properties of Lattice Isomorphism as a Cryptographic Group Action. PQCrypto (1). Vol. 14771. Lecture Notes in Computer Science. Springer, 2024, pp. 170–201.

[15] O. Bernard, A. Lesavourey, TH Nguyen, A. Roux-Langlois. Log-S-unit Lattices Using Explicit Stickelberger Generators to Solve Approx Ideal-SVP. ASIACRYPT (3). Vol. 13793. Lecture Notes in Computer Science. Springer, 2022, pp. 677–708.

[16] W. Beullens, S. Dobson, S. Katsumata, Y.-F. Lai, F. Pintore. Group signatures and more from isogenies and lattices: generic, simple, and efficient. Vol. 91. 6. 2023, pp. 2141–2200.

[17] M. Bolboceanu, Z. Brakerski, D. Sharma. On Algebraic Embedding for Unstructured Lattices. Public Key Cryptography (3). Vol. 14603. Lecture Notes in Computer Science. Springer, 2024, pp. 123–154.

[18] C. Boschini, A. Takahashi, M. Tibouchi. MuSig-L: Lattice-Based Multi-signature with Single-Round Online Phase. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 276–305.

[19] K. Boudgoust, E. Gachon, A. Pellet-Mary. Some Easy Instances of Ideal-SVP and Implications on the Partial Vandermonde Knapsack Problem. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 480–509.

[20] M. Braithwaite. Experimenting with post-quantum cryptography. `https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html`. 2023-04. (2024-03-04 閲覧).

[21] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, D. Stehlé. Classical hardness of learning with errors. STOC. ACM, 2013, pp. 575–584.

[22] K. Carrier, T. Debris-Alazard, C. Meyer-Hilfiger, J.-P. Tillich. Reduction from Sparse LPN to LPN, Dual Attack 3.0. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 286–315.

[23] Y. Chen. sfDualMS: Efficient Lattice-Based Two-Round Multi-signature with Trapdoor-Free Simulation. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 716–747.

[24] Y. Chen, Q. Liu, M. Zhandry. Quantum Algorithms for Variants of Average-Case Lattice Problems via Filtering. EUROCRYPT (3). Vol. 13277. Lecture Notes in Computer Science. Springer, 2022, pp. 372–401.

[25] C. Chuengsatiansup, T. Prest, D. Stehlé, A. Wallet, K. Xagawa. ModFalcon: Compact Signatures Based On Module-NTRU Lattices. AsiaCCS. ACM, 2020, pp. 853–866.

[26] J.-S. Coron, F. Gérard, M. Trannoy, R. Zeitoun. Improved Gadgets for the High-Order Masking of Dilithium. Vol. 2023. 4. 2023, pp. 110–145.

[27] R. Cramer, L. Ducas, B. Wesolowski. Mildly Short Vectors in Cyclotomic Ideal Lattices in Quantum Polynomial Time. J. ACM. Vol. 68, Num. 2 (2021), 8:1–8:26.

[28] R. Cramer, L. Ducas, B. Wesolowski. Short Stickelberger Class Relations and Application to Ideal-SVP. EUROCRYPT (1). Vol. 10210. Lecture Notes in Computer Science. 2017, pp. 324–348.

[29] C. Cremers, S. Düzlü, R. Fiedler, M. Fischlin, C. Janson. BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures. Symposium on Security and Privacy (SP). IEEE, 2021, pp. 1696–1714.

[30] D. Dachman-Soled, H. Gong, T. Hanson, H. Kippen. Revisiting Security Estimation for LWE with Hints from a Geometric Perspective. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 748–781.

[31] TU Darmstadt, UC San Diego. LWE Challenge. https://www.latticechallenge.org/lwe_challenge/challenge.php. (2024-03-04 閲覧).

[32] TU Darmstadt, UC San Diego. SVP Challenge, Hall Of Fame. https://www.latticechallenge.org/svp-challenge/halloffame.php. (2024-03-04 閲覧).

[33] D. Das, A. Joux. Key Recovery Attack on the Partial Vandermonde Knapsack Problem. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 205–225.

[34] T. Debris-Alazard, P. Fallahpour, D. Stehlé. Quantum Oblivious LWE Sampling and Insecurity of Standard Model Lattice-Based SNARKs. STOC. ACM, 2024, pp. 423–434.

[35] J. Devevey, A. Passelègue, D. Stehlé. G+G: A Fiat-Shamir Lattice Signature Based on Convolved Gaussians. ASIACRYPT (7). Vol. 14444. Lecture Notes in Computer Science. Springer, 2023, pp. 37–64.

[36] N. Döttling, D. Kolonelos, R. W. F. Lai, C. Lin, G. Malavolta, A. Rahimi. Efficient Laconic Cryptography from Learning with Errors. EUROCRYPT (3). Vol. 14006. Lecture Notes in Computer Science. Springer, 2023, pp. 417–446.

[37] N. Döttling, J. Müller-Quade. Lossy Codes and a New Variant of the Learning-With-Errors Problem. EUROCRYPT. Vol. 7881. Lecture Notes in Computer Science. Springer, 2013, pp. 18–34.

[38] L. Ducas, E. W. Postlethwaite, L. N. Pulles, W. P. J. van Woerden. HAWK: Module LIP Makes Lattice Signatures Fast, Compact and Simple. ASIACRYPT (4). Vol. 13794. Lecture Notes in Computer Science. Springer, 2022, pp. 65–94.

[39] L. Ducas, T. Prest. Fast Fourier Orthogonalization. ISSAC. ACM, 2016, pp. 191–198.

[40] L. Ducas, M. Stevens, W. P. J. van Woerden. Advanced Lattice Sieving on GPUs, with Tensor Cores. EUROCRYPT (2). Vol. 12697. Lecture Notes in Computer Science. Springer, 2021, pp. 249–279.

[41] L. Ducas, W. P. J. van Woerden. On the Lattice Isomorphism Problem, Quadratic Forms, Remarkable Lattices, and Cryptography. EUROCRYPT (3). Vol. 13277. Lecture Notes in Computer Science. Springer, 2022, pp. 643–673.

[42] S. Düzlü, R. Fiedler, M. Fischlin. BUFFing FALCON without Increasing the Signature Size. IACR Cryptol. ePrint Arch. (2024), p. 710.

[43] T. Espitau, P.-A. Fouque, F. Gérard, M. Rossi, A. Takahashi, M. Tibouchi, A. Wallet, Y. Yu. MITAKA: A Simpler, Parallelizable, Maskable Variant of FALCON. EUROCRYPT (3). Vol. 13277. Lecture Notes in Computer Science. Springer, 2022, pp. 222–253.

[44] T. Espitau, T. Thu Quyen Nguyen, C. Sun, M. Tibouchi, A. Wallet. ANTRAG: Annular NTRU trapdoor generation - Making MITAKA as secure as FALCON. ASIACRYPT (7). Vol. 14444. Lecture Notes in Computer Science. Springer, 2023, pp. 3–36.

[45] T. Espitau, M. Tibouchi, A. Wallet, Y. Yu. Shorter Hash-and-Sign Lattice-Based Signatures. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 245–275.

[46] ETSI TR 103 616 V1.1.1 (2021-09) CYBER; Quantum-safe signatures. https://www.etsi.org/deliver/etsi_tr/103600_103699/103616/01.01.01_60/tr_103616v010101p.pdf. 2021-09. (2024-03-04 閲覧).

[47] J. Felderhoff, A. Pellet-Mary, D. Stehlé. On Module Unique-SVP and NTRU. ASIACRYPT (3). Vol. 13793. Lecture Notes in Computer Science. Springer, 2022, pp. 709–740.

[48] A. Fiat, A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. CRYPTO. Vol. 263. Lecture Notes in Computer Science. Springer, 1986, pp. 186–194.

[49] P.-A. Fouque, F. Gérard, M. Rossi, Y. Yu. ℤalcon: An alternative FPA-free NTRU sampler for Falcon. Third PQC Standardization Conference. 2021-06. （2024-03-04 閲覧）.

[50] P.-A. Fouque et al. FALCON: Fast-Fourier lattice-based compact signatures over NTRU. Specification v1.0. `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/Falcon.zip`. （2024-03-04 閲覧）.

[51] P.-A. Fouque et al. FALCON: Fast-Fourier lattice-based compact signatures over NTRU. Specification v1.2 – 01/10/2020. `https://falcon-sign.info/falcon.pdf`. 2020-10. （2024-03-04 閲覧）.

[52] C. Gentry, C. Peikert, V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC. ACM, 2008, pp. 197–206.

[53] O. Goldreich, S. Goldwasser, S. Halevi. Public-Key Cryptosystems from Lattice Reduction Problems. CRYPTO. Vol. 1294. Lecture Notes in Computer Science. Springer, 1997, pp. 112–131.

[54] M. Grillere, P. Thomassen, N. Wisiol. FALCON-512 in PowerDNS. `https://blog.powerdns.com/2022/04/07/falcon-512-in-powerdns/`. 2022-04. （2024-03-04 閲覧）.

[55] J. Hoffstein, J. Pipher, J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. ANTS. Vol. 1423. Lecture Notes in Computer Science. Springer, 1998, pp. 267–288.

[56] D. Hofheinz, K. Hostáková, R. Langrehr, B. Ursu. On Structure-Preserving Cryptography and Lattices. Public Key Cryptography (3). Vol. 14603. Lecture Notes in Computer Science. Springer, 2024, pp. 255–287.

[57] J. Howe, T. Pöppelmann, M. O'Neill, E. O'Sullivan, T. Güneysu. Practical Lattice-Based Digital Signature Schemes. ACM Trans. Embed. Comput. Syst. Vol. 14, Num. 3 (2015), 41:1–41:24.

[58] A. Hülsing, K.-C. Ning, P. S., F. Weber, P. R. Zimmermann. Post-quantum WireGuard. SP. IEEE, 2021, pp. 304–321.

[59] K. Kim, M. Tibouchi, A. Wallet, T. Espitau, A. Takahashi, Y. Yu, S. Guilley. SOLMAE – Algorithm specifications. `https://kpqc.or.kr/images/pdf/SOLMAE.pdf`. （2024-03-04 閲覧）.

[60] R. W. F. Lai, G. Malavolta. Lattice-Based Timed Cryptography. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 782–804.

[61] A. Langlois, D. Stehlé. Worst-case to average-case reductions for module lattices. Des. Codes Cryptogr. Vol. 75, Num. 3 (2015), pp. 565–599.

[62] M. Lantz. World's first quantum computing safe tape drive. `https://www.ibm.com/blogs/research/2019/08/crystals/`. 2019-08. （2024-03-04 閲覧）.

[63] X. Lin, M. Suzuki, S. Zhang, T. Espitau, Y. Yu, M. Tibouchi, M. Abe. Cryptanalysis of the Peregrine Lattice-Based Signature Scheme. Public Key Cryptography (1). Vol. 14601. Lecture Notes in Computer Science. Springer, 2024, pp. 387–412.

[64] R. Lindner, C. Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. CT-RSA. Vol. 6558. Lecture Notes in Computer Science. Springer, 2011, pp. 319–339.

[65] H. Liu, Y. Yu. A Non-heuristic Approach to Time-Space Tradeoffs and Optimizations for BKW. ASIACRYPT (3). Vol. 13793. Lecture Notes in Computer Science. Springer, 2022, pp. 741–770.

[66] Z. Liu, V., J. Ding, C. Cheng, Y. Pan. An Improved Practical Key Mismatch Attack Against NTRU. PQCrypto (1). Vol. 14771. Lecture Notes in Computer Science. Springer, 2024, pp. 322–342.

[67] V. Lyubashevsky. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. ASIACRYPT. Vol. 5912. Lecture Notes in Computer Science. Springer, 2009, pp. 598–616.

[68] V. Lyubashevsky. Lattice Signatures without Trapdoors. EUROCRYPT. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 738–755.

[69] V. Lyubashevsky, C. Peikert, O. Regev. On Ideal Lattices and Learning with Errors over Rings. EUROCRYPT. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 1–23.

[70] V. Lyubashevsky, C. Peikert, O. Regev. On Ideal Lattices and Learning with Errors over Rings. J. ACM. Vol. 60, Num. 6 (2013), 43:1–43:35.

[71] D. Micciancio, M. Schultz. Error Correction and Ciphertext Quantization in Lattice Cryptography. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 648–681.

[72] D. Micciancio, V. Vaikuntanathan. SoK: Learning with Errors, Circular Security, and Fully Homomorphic Encryption. Public Key Cryptography (4). Vol. 14604. Lecture Notes in Computer Science. Springer, 2024, pp. 291–321.

[73] G. De Micheli, D. Kim, D. Micciancio, A. Suhl. Faster Amortized FHEW Bootstrapping Using Ring Automorphisms. Public Key Cryptography (4). Vol. 14604. Lecture Notes in Computer Science. Springer, 2024, pp. 322–353.

[74] G. De Micheli, D. Micciancio, A. Pellet-Mary, N. Tran. Reductions from Module Lattices to Free Module Lattices, and Application to Dequantizing Module-LLL. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 836–865.

[75] G. Mureau, A. Pellet-Mary, G. Pliatsok, A. Wallet. Cryptanalysis of Rank-2 Module-LIP in Totally Real Number Fields. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 226–255.

[76] P. Q. Nguyen, J. Stern. Cryptanalysis of the Ajtai-Dwork Cryptosystem. CRYPTO. Vol. 1462. Lecture Notes in Computer Science. Springer, 1998, pp. 223–242.

[77] Q. Nguyen. ANTRAG: Simplifying and improving Falcon Without Compromising Security. `https://csrc.nist.gov/csrc/media/Presentations/2024/antrag-simplifying-and-improving-falcon/images-media/nguyen-antrag-pqc2024.pdf`. 2024-04. (2024-12-30 閲覧).

[78] NIST. Module-Lattice-Based Digital Signature Standard. NIST FIPS 204, `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf`. 2024-08.

[79] NIST. Module-Lattice-Based Key-Encapsulation Mechanism Standard. NIST FIPS 203, `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf`. 2024-08.

[80] OpenSSH 8.9 was released on 2022-02-23. `https://www.openssh.com/txt/release-8.9`. (2024-03-04 閲覧).

[81] J. Pan, B. Wagner, R. Zeng. Lattice-Based Authenticated Key Exchange with Tight Security. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 616–647.

[82] R. del Pino, S. Katsumata. A New Framework for More Efficient Round-Optimal Lattice-Based (Partially) Blind Signature via Trapdoor Sampling. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 306–336.

[83] T. Plantard, W. Susilo, K. Than Win. A Digital Signature Scheme Based on CVP_∞. Public Key Cryptography. Vol. 4939. Lecture Notes in Computer Science. Springer, 2008, pp. 288–307.

[84] A. Pouly, Y. Shen. Provable Dual Attacks on Learning with Errors. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 256–285.

[85] T. Prest. FALCON Update (2024). https://csrc.nist.gov/csrc/media/Presentations/2024/falcon/images-media/prest-falcon-pqc2024.pdf. 2024-04. (2024-12-30 閲覧).

[86] The Chromium Projects. CECPQ2. https://www.chromium.org/cecpq2/. (2024-03-04 閲覧).

[87] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. STOC. ACM, 2005, pp. 84–93.

[88] O. Regev. The Learning with Errors Problem (Invited Survey). CCC. IEEE Computer Society, 2010, pp. 191–204.

[89] K. Ryan, N. Heninger. Fast Practical Lattice Reduction Through Iterated Compression. CRYPTO (3). Vol. 14083. Lecture Notes in Computer Science. Springer, 2023, pp. 3–36.

[90] Open Quantum Safe. Algorithms in liboq. https://openquantumsafe.org/liboqs/algorithms/. (2024-03-04 閲覧).

[91] Federal office for information security. BSI – Technical guideline (Cryptographic mechanisms: Recommendations and key lengths). https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile&v=10. 2023-01. (2024-03-04 閲覧).

[92] E.-Y. Seo, Y.-S. Kim, J.-W. Lee, J.-S. No. Peregrine: Toward Fastest FALCON Based on GPV Framework. (2022). https://eprint.iacr.org/2022/1495.

[93] H. M. Steffen, G. Land, L. J. Kogelheide, T. Güneysu. Breaking and Protecting the Crystal: Side-Channel Analysis of Dilithium in Hardware. PQCrypto. Vol. 14154. Lecture Notes in Computer Science. Springer, 2023, pp. 688–711.

[94] D. Stehlé, R. Steinfeld. Making NTRU as Secure as Worst-Case Problems over Ideal Lattices. EUROCRYPT. Vol. 6632. Lecture Notes in Computer Science. Springer, 2011, pp. 27–47.

[95] M. J. Steiner. The Complexity of Algebraic Algorithms for LWE. EUROCRYPT (3). Vol. 14653. Lecture Notes in Computer Science. Springer, 2024, pp. 375–403.

[96] L. Wang. Analyzing Pump and Jump BKZ Algorithm Using Dynamical Systems. PQCrypto (1). Vol. 14771. Lecture Notes in Computer Science. Springer, 2024, pp. 406–432.

[97] R. Wang, M. Brisfors, E. Dubrova. A Side-Channel Attack on a Higher-Order Masked CRYSTALS-Kyber Implementation. ACNS (3). Vol. 14585. Lecture Notes in Computer Science. Springer, 2024, pp. 301–324.

[98] Z. Wang, Q. Lai, F.-H. Liu. Ring/Module Learning with Errors Under Linear Leakage – Hardness and Applications. Public Key Cryptography (2). Vol. 14602. Lecture Notes in Computer Science. Springer, 2024, pp. 275–304.

[99] H. Wee, D. J. Wu. Succinct Vector, Polynomial, and Functional Commitments from Lattices. EUROCRYPT (3). Vol. 14006. Lecture Notes in Computer Science. Springer, 2023, pp. 385–416.

[100] wolfSSL. wolfSSL support for Apache httpd and curl (Post-Quantum Edition). https://github.com/wolfSSL/osp/blob/master/apache-httpd/README_post_quantum.md. (2024-03-04 閲覧).

[101] W. Xia, L. Wang, G. Wang, D. Gu, B. Wang. A Refined Hardness Estimation of LWE in Two-Step Mode. Public Key Cryptography (3). Vol. 14603. Lecture Notes in Computer Science. Springer, 2024, pp. 3–35.

[102] W. Xia, L. Wang, G. Wang, D. Gu, B. Wang. Refined Strategy for Solving LWE in Two-step Mode. Cryptology ePrint Archive, Paper 2022/1343. 2022. https://eprint.iacr.org/2022/1343.

[103] Y. Yu, H. Jia, X. Wang. Compact Lattice Gadget and Its Applications to Hash-and-Sign Signatures. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 390–420.

[104] 量子耐性暗号研究団. KpqC. https://kpqc.or.kr/. (2024-12-06 閲覧).

第 4 章

符号に基づく暗号技術

本章では符号に基づく暗号技術についてまとめる。符号に基づく暗号技術の安全性はシンドローム復号（Syndrome Decoding: SD）問題や Learning Parity with Noise: LPN 問題を解く計算の困難性に依存している。

■準備: 本章で使用する記号・用語を以下にまとめる。以下では，q を素数 p のべきとする。すなわち，ある正整数 k が存在して $q = p^k$ である。以下では log の底が省略されている場合は底を 2 とする。自然対数を用いる場合は ln と書く。

有限体: $\mathbb{F}_q$ で位数が q の有限体を表す。

ハミング重みとハミング距離: V_n を有限体 $\mathbb{F}_q$ 上の n 次元ベクトル空間とする。

- 行ベクトル $\boldsymbol{v} = (v_1, v_2, \ldots, v_n) \in V_n$ のハミング重みとは，非ゼロの成分の数である。すなわち，有限集合 X に対して $|X|$ で X の要素数を表すとき，$\mathsf{HW}(\boldsymbol{v}) = |\{i \mid v_i \neq 0\}|$ である。
- ハミング距離を $d_H(\boldsymbol{x}, \boldsymbol{y}) = \mathsf{HW}(\boldsymbol{x} - \boldsymbol{y})$ で定義する。
- $\mathcal{S}_H(n, w)$ でハミング重みが w の n 次元ベクトル全体の集合を表す。
- $\mathcal{S}_H^{\leq}(n, w)$ でハミング重みが w 以下の n 次元ベクトル全体の集合を表す。

■線形符号: 線形符号とは，誤りが発生する通信路において，メッセージを相手に正しく伝えるための技術である。メッセージを冗長にして（符号化という）送信し，受信時に伝送中に生じた誤りを訂正する（復号という）ことで，正しいメッセージを得ることができる。自然数 n および 素数べき q について，$\mathbb{F}_q$ 上の n 次元ベクトル空間の線形部分空間を $\mathbb{F}_q$ 上の線形符号と呼び，$\mathcal{C}$ で表す。$\mathcal{C}$ の要素を符号語と呼び，n を符号長という。このとき $[\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k]$ を $\mathcal{C}$ の基底とする。k を線形符号の次元と呼ぶ。$\mathbb{F}_q$ 上の線形符号の符号長が n，次元が k であるとき，$[n, k]_q$ -線形符号とよぶ。$[n, k]_q$ -線形符号 $\mathcal{C}$ の最小距離 d とは，2 つの異なる符号語間のハミング距離の最小値である。d は符号 $\mathcal{C}$ の全ての非ゼロ符号語のハミング重みの最小値に一致する。すなわち，$d = \min_{\boldsymbol{c} \in \mathcal{C} \setminus \{\boldsymbol{0}\}} \mathsf{HW}(\boldsymbol{c})$ である。

$[n, k]_q$ -線形符号 $\mathcal{C}$ の生成行列とは，符号 $\mathcal{C}$ の基底ベクトルを行とする行列 $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$ であり，メッセージの符号化に用いられる。メッセージ $\boldsymbol{s} \in \mathbb{F}_q^k$ に対して，$\boldsymbol{sG} \in \mathbb{F}_q^n$ は符号語である。メッセージと符号語は一対一対応させることができる。$[n, k]_q$ -線形符号 $\mathcal{C}$ のパリティ検査行列とは，行列 $\boldsymbol{H} \in \mathbb{F}_q^{r \times n}$ で，$\boldsymbol{c} \in \mathbb{F}_q^n$ に対して，$\boldsymbol{c} \in \mathcal{C}$ ならばかつその時に限り $\boldsymbol{cH}^\top = \boldsymbol{0}$ となるものである。$\boldsymbol{H}$ の行が一次独立であれば，$r = n - k$ である。組織符号化とは，行列 $\boldsymbol{H}$ に対して，行列 $\boldsymbol{S} \in \mathbb{F}_q^{(n-k) \times (n-k)}$ を適用し，$\boldsymbol{SH} = [\boldsymbol{I}_{n-k} \mid \boldsymbol{Z}]$ を得る操作を指す。ここで，$\boldsymbol{Z} \in \mathbb{F}_q^{(n-k) \times k}$ である。

線形符号は，生成行列やパリティ検査行列をうまく設計することで，受信時に符号語に加えられた誤りを訂正することができる。誤り訂正には，復号アルゴリズムが用いられる。送信する符号語を $\boldsymbol{c}$ とし，通信路上で乗った誤りを $\boldsymbol{e}$ とする。受信者側は，受信語として $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ を得る。受信者は，復号アルゴリズムを用いて $\boldsymbol{y}$ から $\boldsymbol{c}$ を得る。受信者

がハミング重み t までの誤り $\boldsymbol{e}$ を一意に訂正できるとき，符号の訂正能力が t であるという。一般に，$t \leq \lfloor (d-1)/2 \rfloor$ が成り立つ。復号アルゴリズムには，符号の構造を用いる方式や，パリティ検査行列を用いる方式がある。後者の場合，受信語 $\boldsymbol{y}$ に対して $\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^\top$ を計算する。$\boldsymbol{s}$ はシンドロームと呼ばれる。$\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top$ となることから，$\boldsymbol{s} \neq \boldsymbol{0}$ であれば，誤りを検出・訂正できる。

本稿では，具体的な線形符号（リード・ソロモン符号，リード・マラー符号，Goppa 符号）の詳細については扱わない。符号理論の教科書や，電子情報通信学会 知識の森 「1 群 2 編 符号理論」 [44] などを参照されたい。

4.1 符号に基づく暗号技術の安全性の根拠となる問題

本節では，SD 問題と LPN 問題およびその困難性について報告する。

4.1.1 SD 問題

SD 問題とは，解のハミング重みが指定された条件のもとで，$\mathbb{F}_2$ 上の線形方程式を解けるかどうかという問題である。また，$[n,k]_2$ -線形符号において，パリティ検査行列とシンドローム，受信語に乗ったエラーのハミング重みが与えられたときに，エラーを求める問題とみなすことができる。本問題は符号暗号の安全性の根拠として非常に重要である。実際に， 4.3 節で説明する NIST PQC 標準化プロジェクトの第 4 ラウンドの 3 種類の符号暗号いずれの方式においても，SD 問題が安全性の根拠である。

定義 4.1 (SD 問題) $k, w \leq n$ を満たす正の整数 n, k, w に対して，行列 $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ とベクトル $\boldsymbol{s} \in \mathbb{F}_2^{n-k}$ が与えられる。$\mathsf{SD}_{n,k,w}$ 問題は，$\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$ かつ $\mathsf{HW}(\boldsymbol{e}) = w$ を満たすベクトル $\boldsymbol{e} \in \mathbb{F}_2^n$ を求める問題である。

4.1.2 SD 問題に対する評価

SD 問題の計算の困難性に関して，Berlekamp，McEliece，van Tilborg [6] によって，NP 困難な 3 次元マッチング問題から SD 問題への多項式時間帰着が示されている。これにより，SD 問題が NP 困難であることが判明している。$\mathsf{SD}_{n,k,w}$ 問題や $[n,k]_2$ -線形符号における k/n は符号化レートと呼ばれており，符号化レートが 1/2 付近で SD 問題が最も難しくなることが知られている。また，符号化レートを増加させると，公開鍵に相当する入力行列 $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ のサイズが減少することから，暗号の設計においては，符号化レート 1/2 以上 1 未満の値が採用されることが多い。

以降では，SD 問題の求解手法として最も研究が進んでいる Information Set Decoding: ISD を取り上げる。ISD は，符号化レート 0.42 以上において既存方式の中で漸近計算量が最も小さく，SD 問題の解読チャレンジを通して実時間での計算量解析が進展している。

■Information Set Decoding　$\mathsf{SD}_{n,k,w}$ 問題と対応する $[n,k]$-線形符号の最小距離を d と置く。2 進符号の場合，Gilbert-Varshamov 限界により，$k/n \approx 1 - H(d/n)$ である[*1]。$w \approx d$ の場合の $\mathsf{SD}_{n,k,w}$ 問題を Full Distance Decoding と呼ぶ。$w \approx d$ のとき，$\mathsf{SD}_{n,k,w}$ 問題は解くのが最も難しくなる。$w \gg d$ のとき，$\mathsf{SD}_{n,k,w}$ 問題には複数の解が存在することが期待され, $w \leq d$ のとき，$\mathsf{SD}_{n,k,w}$ 問題の解の個数の期待値は 1 以下である。暗号利用においては，$w \ll d$ が選ばれ，トラップドアを通して唯一解が存在するように設計される。以降は $w \leq d$ を考える。

$\mathsf{SD}_{n,k,w}$ 問題を総当りで解くには，ハミング重みが w の n 次元ベクトル $\boldsymbol{e}$ を列挙すればよい。そのため，時間計算

[*1] ここで $H(p) = -p\log(p) - (1-p)\log(1-p)$。

表 4.1: 確率 1/2 以上で SD 問題を解く場合の漸近計算量（Full Distance Decoding の場合）

	$\log(\text{Time})/n$	$\log(\text{Space})/n$	備考
Pra62（Lee-Brickel）	0.121	–	[41, 27]
Stern89	0.117	0.0135	[42]
MMT11	0.112	0.0530	[31]; [21] によって空間計算量が改良された
BJMM12	0.102	0.0769	[5]
MO15	0.0967	0.0890	[32]
BM17	0.0953	0.0910	[11]; MO15 を最適化したもの
BM18	0.0951	0.0760	[10]; [12, 16] によって時間・空間計算量が修正された
Sieving ISD	0.101	0.0636	[15]

量は $O\left(\binom{n}{w}\right)$ となる。より効率的な手法として，Prange は Information Set Decoding と呼ばれる手法 [41] を提案した。基本アイデアは以下である:

1. 一様ランダムに $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ の列ベクトルを入れ替え，$\tilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{P}$ とする。（$\boldsymbol{P} \in \mathbb{F}_2^{n\times n}$ は置換行列。）
2. ガウスの消去法と対応する行列 $\boldsymbol{S} \in \mathbb{F}_2^{(n-k)\times(n-k)}$ によって $\tilde{\boldsymbol{H}}$ を $[\boldsymbol{I}_{n-k} \mid \boldsymbol{Z}] = \boldsymbol{S}\tilde{\boldsymbol{H}}$ とする。（組織符号化）
3. シンドローム $\boldsymbol{s} \in \mathbb{F}_2^{n-k}$ に対して，$\boldsymbol{s}' = \boldsymbol{s}\boldsymbol{S}^\top$ を計算する。
4. $\boldsymbol{s}'$ のハミング重みが w ならば，$\boldsymbol{e} = (\boldsymbol{s}', \boldsymbol{0}_k)\boldsymbol{P}^\top$ を出力する。そうでなければ，1. に戻る。

$\mathsf{HW}(\boldsymbol{s}') = w$ ならば，$\mathsf{HW}(\boldsymbol{e}) = w$ である。また，$\boldsymbol{e}\boldsymbol{H}^\top = (\boldsymbol{s}', \boldsymbol{0}_k)\boldsymbol{P}^\top\boldsymbol{H}^\top = (\boldsymbol{s}', \boldsymbol{0}_k)\tilde{\boldsymbol{H}}^\top = (\boldsymbol{s}, \boldsymbol{0}_k)\boldsymbol{S}^\top\tilde{\boldsymbol{H}}^\top = (\boldsymbol{s}, \boldsymbol{0}_k)[\boldsymbol{I}_{n-k}\boldsymbol{Z}]^\top = \boldsymbol{s}$ が成立する。よって，ステップ 4 のチェックが通るならば，$\boldsymbol{e}$ は SD 問題の解である。末尾 k 列が全て 0 であるような $\boldsymbol{e}\boldsymbol{P}$ は $\binom{n-k}{w}$ 通りあるため，ある置換行列に対して解が出力される確率は $\binom{n-k}{w}/\binom{n}{w}$ となる。期待計算量は $\mathrm{poly}(n,k) \cdot O\left(\binom{n}{w}/\binom{n-k}{w}\right)$ となり，先ほどの列挙法よりも速くなる。

Stern [42] 以降，空間計算量を犠牲にすることで時間計算量を引き下げる ISD の改良アルゴリズムが多数提案されている。以下では，Both と May [10] による時間計算量の表を，表 4.1 に示す。この表は，時間計算量を最小化した場合の符号化レート k/n の最悪時（1/2 の少し下）の漸近計算量についてまとめられている。したがって，問題のパラメータによっては，表の数値よりも速く解くことが可能となる。

近年は，漸近計算量のみならず，具体的なパラメータに対する $\mathsf{SD}_{n,k,w}$ 問題を求解するために必要なビット計算量を見積もる研究もなされている。Esser，Verbel，Zweydinger，Bellini [20] は，CryptographicEstimators と呼称する符号暗号や多変数多項式暗号のビット計算量を推定するソフトウェアを開発した。Narisada ら [36] は，Becker らの ISD [5] の実用的な改良方式を提案し，NIST PQC 標準化プロジェクト第 4 ラウンドの 3 種類の符号暗号のビット計算量と実時間の計算量を算出した。

■量子アルゴリズム　現在のところ多項式時間で SD 問題を解く量子アルゴリズムは提案されていない。しかし，量子アルゴリズムを利用して，いくつかの古典 ISD を高速化する方法を Kachigar と Tillich [24] が提案している。[*2]2024 年現在，最良の漸近計算量が得られているのは，BJMM 法 [5] の量子アルゴリズムである量子 BJMM 法であり，時間

[*2] Kirshanova [26] が Kachigar と Tillich の結果 [24] の改良を提案していたが，誤りがあったことが報告されている。そのため，2024 年時点でのベストな量子アルゴリズムは Kachigar と Tillich [24] であると考えられる。

計算量が $2^{0.0587n}$，空間計算量が $2^{0.0188n}$ となっている。

量子回路設計の研究に関しては，量子 Prange 法に対して，Perriello，Barenghi，Pelosi [39] がグローバーのアルゴリズムを用いた量子回路を提案した。Esser ら [19] は，量子 Prange 法に対して，一部の演算を古典コンピュータ上で行う量子と古典のハイブリッド法を提案した。Perriello，Barenghi，Pelosi [40] は，量子 Prange におけるガウスの消去法の量子回路を改良し，NIST PQC 標準化プロジェクト第 4 ラウンドの 3 種類の符号暗号の解読に必要な量子回路の深さを最大で 2^{30} 削減した。Chevignard，Fouque，Schrottenloher [14] は，量子 Prange 法に対して，量子回路の深さを犠牲にすることで量子ビット数を削減する，深さと幅のトレードオフ手法を提案した。Stern の ISD [42] 以降に提案されたリスト探索を伴う ISD については，グローバーのアルゴリズムと量子ウォーク探索を組み合わせた複雑な量子回路が必要になると考えられている [24]。現在のところ，これらの量子 ISD に対する量子回路は提案されていない。

■現状の進展　格子の場合と同様に “Decoding Challenge”（`https://decodingchallenge.org/`）というウェブサイトが作成された。実時間での計算量解析を通じて，符号に基づく暗号技術の信頼性を向上させることが目的である。現在，以下 5 つのカテゴリが用意されている。

1. $\mathbb{F}_2$ 係数の一様ランダムな線形符号に対する SD 問題
2. $\mathbb{F}_2$ 係数の一様ランダムな線形符号に対するハミング重みが小さい符号語を探索する問題
3. $\mathbb{F}_3$ 係数の一様ランダム線形符号に対する SD 問題
4. Goppa 符号を用いた Niederreiter 暗号の場合の SD 問題（Classic McEliece に対応 4.3.1）
5. QC-MDPC 符号に基づく SD 問題（BIKE や HQC に対応 4.3.2 4.3.3）

各問題に対して研究および解読が進んでおり，2025 年 1 月現在，解読に成功した最も困難な問題は次のとおりである。

- 1. $n = 570, k = n/2$ に対して $w = 70$（成定，福島，清本，2023/04）
- 2. $n = 1280, k = n/2$ の場合に $w = 204$（成定，岡田，上村，相川，福島，清本，2024/09）
- 3. $n = 200, k = \log_3(n)$ の場合に $w = 198$（Esser，May，Zweydinger，2021/12）
- 4. $n = 1409, k = 0.8n$ に対して $w = 26$（成定，古江，相川，福島，清本，2023/11）
- 5. $n = 3602, k = n/2$ に対して $w = 60$（成定，岡田，上村，相川，福島，清本，2025/1）

1 の結果については，Narisada，Fukushima，Kiyomoto [35] を，4 の結果については，Narisada ら [36] を参照されたい。2，3，5 についての詳細は，Decoding Challenge 上に掲載されている各記録の詳細を参照されたい。

4.1.3　LPN 問題

LPN 問題とは，$\mathbb{F}_2$ 上の誤差付きの線形方程式を解けるかどうかという問題である。また，$[n, k]_2$ -線形符号において，生成行列と受信語が与えられたときに，メッセージを復号する問題とみなすことができる。1993 年に，Blum，Furst，Kearns，Lipton [7] が困難と思われる問題として挙げ，定式化を行った。第 3 章において，この問題を一般化した LWE 問題を既に扱っている。

Ber_τ でパラメータ τ のベルヌーイ分布を表すことにする。（確率 τ で 1，確率 $1-\tau$ で 0 となる $\mathbb{F}_2$ 上の分布である。）また，自然数 $k \geq 1$ について，Ber_τ^k で，Ber_τ から独立に k 個サンプルを取ったときの $\mathbb{F}_2^k$ 上の分布を表す。

定義 4.2 (LPN 問題) $\mathbb{F}_2^k$ から一様ランダムに選ばれた秘密鍵 $\boldsymbol{s}$ およびエラー比 $\tau \in [0, 1/2)$ に対して，以下の LPN サンプルを出力する LPN オラクルを考える。

$$(\boldsymbol{a}, b) = (\boldsymbol{a}, \boldsymbol{s} \cdot \boldsymbol{a}^\top + e),$$

ここで，$\boldsymbol{a}$ は $\mathbb{F}_2^k$ から一様ランダムに選び，e は分布 Ber_τ に従い選ぶ。LPN オラクルを n 回呼び出すとき，$(\boldsymbol{A}, \boldsymbol{b}) \leftarrow \mathsf{LPN}_{k,\tau}^n$ と表記する。これは，n 個の LPN サンプル $(\boldsymbol{a}_1, b_1), (\boldsymbol{a}_2, b_2), \ldots, (\boldsymbol{a}_n, b_n)$ を行列・ベクトル表示して，

$$\boldsymbol{A} = [\boldsymbol{a}_1^\top \boldsymbol{a}_2^\top \ldots \boldsymbol{a}_n^\top] \in \mathbb{F}_2^{k \times n}, \qquad \boldsymbol{b} = \boldsymbol{s}\boldsymbol{A} + \boldsymbol{e} \in \mathbb{F}_2^n$$

としたものである。n をサンプル数と呼ぶ。$\boldsymbol{e}$ は，n 個の LPN サンプルのエラー e を成分とするベクトルである。$\mathsf{LPN}_{k,\tau}$ 問題とは，LPN オラクルへのアクセスが可能なときに，$\boldsymbol{s}$ を求める問題である。

サンプル数が n の $\mathsf{LPN}_{k,\tau}$ 問題は，$\mathsf{SD}_{n,k,n\tau}$ 問題に変換することができる。変種として，体を $\mathbb{F}_q$ に変更した LPN 問題・仮定が用いられることもある。LPN 問題の安全性仮定について詳しく知りたい方は，2022 年度版の CRYPTREC 耐量子計算機暗号の研究動向調査報告書 [1, Section 3.1] を参照して欲しい。

4.1.4 LPN 問題に対する評価

LPN 問題の計算の困難性に関して，サンプル数を固定した場合，NP 困難になることが Berlekamp，McEliece，van Tilborg [6] によって示されている[*3]。また，Håstad [23] により近似版 LPN 問題[*4]の NP 困難性も示されている。しかし，平均時の困難性についてはよく分かっていない。

LPN 問題の古典求解手法として，現在，大別して以下の 5 つのアルゴリズムが知られている。

1. ガウスの消去法に基づく手法 [13]
2. SD 問題における Information Set Decoding に基づく手法 [18]
3. Blum，Kalai，Wasserman [8] の BKW アルゴリズムに基づく手法
4. Arora，Ge [4] の「再線形化」アルゴリズム
5. 2. と 3. を組み合わせたハイブリッド法 [18]

このうち，著者の知る限り漸近的に時間計算量が最も小さい手法は BKW アルゴリズムであり，実用上最も高速な手法はハイブリッド法である。以降で各手法の概要を説明する。

■ガウスの消去法に基づく手法　ガウスの消去法に基づく手法は，2008 年に Carrijo，Tonicelli，Imai，Nascimento [13] によって初めて提案された LPN 問題に対する多項式空間・指数時間アルゴリズムである。この手法は指数回数の LPN オラクルの呼び出しが必要であるが，k 個の LPN サンプルを格納するメモリがあれば良いので，必要な計算資源が少なく，実装が容易である。本手法の計算量は，時間計算量が $\mathrm{poly}(k) \cdot O(2^k)$，空間計算量が $O(k^2)$，サンプル数が $n = O(2^k)$ である。

■Information Set Decoding に基づく手法　$\mathsf{LPN}_{k,\tau}$ は任意のサンプル数 n に対して SD 問題（$\mathsf{SD}_{n,k,\tau n}$）に変換できるため，LPN 問題は SD 問題の効率的な求解手法である Information Set Decoding を用いて解くことができる。Esser，Kübler，May [18] によって提案された実用的なアルゴリズムは，ガウスの消去法に基づく手法を拡張したものである。基本アイデアとして，ガウスの消去法に基づく手法は k 個の LPN サンプルのエラーが全て 0 の場合を考えるが，その拡張として n 個の LPN サンプルのうち k 個の LPN サンプルのエラーが全て 0 となる組み合わせを考える。$n = O(k^2)$ に設定することで，高い確率でこのような組み合わせが存在する。ガウスの消去法に基づく手法と比較して，LPN サンプルの数を $O(2^k)$ から $O(k^2)$ まで減らせる点が利点である。

[*3] $\boldsymbol{A}$ および $\boldsymbol{b}$ を与えられたときに，線形方程式 $\boldsymbol{s}\boldsymbol{a}_i^\top = b_i$ を満たす数を最大化する $\boldsymbol{s}$ を探索する問題を考える。

[*4] $\boldsymbol{A}$ および $\boldsymbol{b}$ を与えられたときに，線形方程式 $\boldsymbol{s}\boldsymbol{a}_i^\top = b_i$ を近似度×最大値以上満たす $\boldsymbol{s}$ を探索する問題。

アルゴリズムの概要は，$\mathsf{LPN}^n_{k,\tau}$ 問題に対して，$\mathsf{SD}_{n,k,\tau n}$ 問題を考え，Information Set Decoding (ISD) を用いて $\mathsf{SD}_{n,k,\tau n}$ 問題を解くと言うものである [18]。ISD には様々な手法があるが，[18] において，MMT 法が実用上最適であることが示されている。本手法の計算量は，ISD の手法によって異なる関数 $c(\tau)$ に対して，時間計算量が $2^{c(\tau)k}$，空間計算量が $O(k^3)$，サンプル数が $O(k^2)$ である。

■BKW アルゴリズムに基づく手法　KW アルゴリズム [8] は，LPN 問題に対する最も著名な手法である。基本アイデアは以下である。オラクルからのサンプル $(\boldsymbol{a}, b)$ が $\boldsymbol{a} = (1, 0, \ldots, 0)$ という形であれば，$b = s_1 + e$ となる。このようなサンプルを大量に集めれば，s_1 を多数決法で求めることが出来る。一般に $\boldsymbol{u}_j$ を j 番目の単位ベクトルとして，$(\boldsymbol{u}_j, b)$ という形のサンプルを集めれば s_j を多数決法で求められる。そこで，LPN オラクルからのサンプルを用いて，このようなサンプルを生成することを目指す。BKW アルゴリズムは，LPN サンプル同士の加算を実施することに起因してノイズが増加する欠点がある。

BKW アルゴリズムの計算量は，時間計算量・空間計算量・サンプル数いずれも $2^{O(k/\log k)}$ である。よって，大きな次元の LPN 問題に対しては，メモリ量の増加が課題となる。その後，Levieil と Fouque [28]，Kirchner [25]，Guo, Johansson, Löndahl [22]，Zhang, Jiao, Wang [43]，Bogos と Vaudenay [9]，Esser, Kübler, May [18]，Esser, Heuer, Kübler, May, Sohler [17] などで改良やメモリと時間のトレードオフの議論がおこなわれている。

■Arora–Ge アルゴリズム　Arora と Ge [4] は多変数多項式問題で古くから用いられている再線形化と呼ばれる手法を用いて，LPN 問題を解くことを考えた。このアルゴリズムをサンプル数 n の $\mathsf{LPN}_{k,\tau}$ 問題に用いた場合，$w = \tau n$ として，$\mathrm{poly}(k^w)$ 時間で解くことができる。$\mathrm{poly}(k^w) = 2^{O(\tau n \log k)}$ であるから，$\tau = o(k/(n\log^2 k))$ のようにエラーが疎であれば，BKW アルゴリズムよりも効率が良い。実際の符号暗号のパラメータ設定では，エラーをこのように疎に設定することはないため，暗号の攻撃アルゴリズムとして用いるには重要度が低い。

■Information Set Decoding と BKW を組み合わせたハイブリッド法　Esser, Kübler, May [18] は，BKW アルゴリズムと Information Set Decoding を組み合わせた実用上高速な手法を提案した。本手法の計算量は，Information Set Decoding に基づく手法の計算量と BKW アルゴリズムに基づく手法の計算量との中間である。

報告によれば，$k = 135, \tau = 1/4$ の LPN 問題に対して，16 コアの CPU および 256GB の RAM を搭載したサーバ 1 台を用いて，5.69 日での求解に成功した。また，$k = 243, \tau = 1/8$ の LPN 問題に対して，同じサーバ 1 台を用いて，15.07 日での求解に成功した。また，暗号設計に用いられるパラメータを持つ LPN 問題に対して，空間計算量を現実的な値にセキュリティマージンを加えたものに制限（2^{60}bit = 128PB および 2^{80}bit = 128ZB）した時のビット計算量が推定された。報告によれば，$k = 512, \tau = 1/8$ の LPN 問題に対するハイブリッド法のビット計算量は 2^{102} であり，$k = 512, \tau = 1/4$ の LPN 問題に対するビット計算量は 2^{151} である。一方で，この空間計算量の範囲では BKW アルゴリズム [22] は動作しないということである。

■量子アルゴリズム　現在のところ，多項式時間で LPN 問題を解く量子アルゴリズムは提案されていない。Esser, Kübler, May [18] は，上述する ISD に基づく手法やハイブリッド法に対して，グローバーのアルゴリズムや量子ウォーク探索を用いることで高速化できる点を指摘している。$k = 512, \tau = 1/8$ の LPN 問題に対する量子ハイブリッド法のビット計算量は 2^{69} であり，$k = 512, \tau = 1/4$ の LPN 問題に対するビット計算量は 2^{112} と推定されている。

4.2　符号に基づく代表的な暗号方式

本節では，符号に基づく代表的な暗号方式の説明を行う。以下では，$\mathrm{GL}_k(\mathbb{F}_q)$ で k 次の $\mathbb{F}_q$ 要素正則行列全体がなす群を表す。また，S_n で n 次対称群を表す。S_n の要素である置換を $\mathrm{GL}_n(\mathbb{F}_q)$ 中の置換行列と同一視することとする。

4.2.1 McEliece 暗号

McEliece [33] が提案した古典的な暗号方式である。以下では $q = 2$ とする。

- k：安全性パラメータ
- n：サンプルの個数
- τ：誤差パラメータ（例: $\tau n = O(k)$）
- t：線形符号の誤り訂正能力（$t = \Omega(\tau n)$）

鍵生成: 誤り訂正能力が t である $[n, k]_2$-線形符号の生成行列 $\boldsymbol{G} \in \mathbb{F}_2^{k \times n}$ を生成する。$\boldsymbol{S} \leftarrow \mathrm{GL}_k(\mathbb{F}_2)$ を一様ランダムに選ぶ。$\boldsymbol{P} \leftarrow S_n$ を一様ランダムに選ぶ。$\tilde{\boldsymbol{G}} = \boldsymbol{SGP}$ とする。
公開鍵を $\tilde{\boldsymbol{G}}$ とし，秘密鍵を $(\boldsymbol{S}, \boldsymbol{G}, \boldsymbol{P})$ とする。
暗号化: 平文を $\boldsymbol{m} \in \mathbb{F}_2^k$ とする。乱数 $\boldsymbol{e} \leftarrow \mathsf{Ber}_\tau^n$ を選び，暗号文 $\boldsymbol{c} = \boldsymbol{m}\tilde{\boldsymbol{G}} + \boldsymbol{e} \in \mathbb{F}_2^n$ を計算する。
復号: $\hat{\boldsymbol{v}} = \boldsymbol{cP}^{-1}$ を計算する。$\hat{\boldsymbol{v}}$ を線形符号で訂正し $\boldsymbol{m}' \in \mathbb{F}_2^k$ を得る。$\boldsymbol{m} = \boldsymbol{m}'\boldsymbol{S}^{-1}$ を出力する。

復号の正当性は以下で確認される。$\boldsymbol{c} = \boldsymbol{m}\tilde{\boldsymbol{G}} + \boldsymbol{e}$ として，$\hat{\boldsymbol{v}} = \boldsymbol{cP}^{-1}$ を計算すると，

$$\hat{\boldsymbol{v}} = \boldsymbol{m}\tilde{\boldsymbol{G}}\boldsymbol{P}^{-1} + \boldsymbol{e}\boldsymbol{P}^{-1} = \boldsymbol{mSG} + \boldsymbol{e}\boldsymbol{P}^{-1}$$

を得る。$\boldsymbol{mSG}$ はランダム化されたメッセージ $\boldsymbol{mS}$ の符号語であり，$\boldsymbol{e}\boldsymbol{P}^{-1}$ は誤りである。$\boldsymbol{e}\boldsymbol{P}^{-1}$ のハミング重みが t 以下であれば，線形符号の復号により，$\boldsymbol{m}' = \boldsymbol{mS}$ を得る。よって，高い確率で復号に成功する。平文 $\boldsymbol{m}$ および生成行列 $\tilde{\boldsymbol{G}}$ が一様ランダムであれば，暗号文 $\boldsymbol{c} \in \mathbb{F}_2^n$ はランダムな n 次元のベクトルと見分けが付かないと考えられている。一方で，平文 $\boldsymbol{m}$ が零ベクトルのとき，暗号文はランダムなベクトルと区別されてしまう。このことから，オリジナルの McEliece 暗号にはセキュリティ上の課題が存在することがわかる。

4.2.2 Niederreiter 暗号

Niederreiter [37] が 1986 年に提案した。のちに McEliece 暗号と安全性が等価であることが示された。詳しくは [29] を参照のこと。以下では $q = 2$ とする。

- k：安全性パラメータ
- n：サンプルの個数
- t：線形符号の誤り訂正能力

鍵生成: 誤り訂正能力が t である $[n, k]_2$ -線形符号のパリティ検査行列 $\boldsymbol{H} \in \mathbb{F}_2^{(n-k) \times n}$ を生成する。$\boldsymbol{T} \leftarrow \mathrm{GL}_{n-k}(\mathbb{F}_2)$ を一様ランダムに選ぶ。$\boldsymbol{Q} \leftarrow S_n$ を一様ランダムに選ぶ。$\tilde{\boldsymbol{H}} = \boldsymbol{THQ}$ とする。
公開鍵を $\tilde{\boldsymbol{H}}$ とし，秘密鍵を $(\boldsymbol{T}, \boldsymbol{H}, \boldsymbol{Q})$ とする。
暗号化: 平文を $\boldsymbol{e} \in \mathcal{S}_H(n, t)$ とする。暗号文 $\boldsymbol{d} = \boldsymbol{e}\tilde{\boldsymbol{H}}^\top \in \mathbb{F}_2^{n-k}$ を計算する。
復号: $\hat{\boldsymbol{w}} = \boldsymbol{dT}^{-\top}$ を計算する。$\hat{\boldsymbol{w}}$ を線形符号で訂正し復号し，誤りとして $\boldsymbol{e}'$ を得る。$\boldsymbol{e} = \boldsymbol{e}'\boldsymbol{Q}^{-\top}$ を出力する。

復号の正当性は以下で確認される。$\boldsymbol{d} = \boldsymbol{e}\tilde{\boldsymbol{H}}^\top$ として，$\hat{\boldsymbol{w}} = \boldsymbol{dT}^{-\top}$ を計算すると，

$$\hat{\boldsymbol{v}} = \boldsymbol{e}\tilde{\boldsymbol{H}}^\top\boldsymbol{T}^{-\top} = \boldsymbol{e}\boldsymbol{Q}^\top\boldsymbol{H}^\top\boldsymbol{T}^\top\boldsymbol{T}^{-\top} = \boldsymbol{e}\boldsymbol{Q}^\top\boldsymbol{H}^\top$$

を得る。$\boldsymbol{e}\boldsymbol{Q}^\top$ はランダムに置換されたエラーであり，$\boldsymbol{e}\boldsymbol{Q}^\top\boldsymbol{H}^\top$ はシンドロームである。$\boldsymbol{e}\boldsymbol{Q}^\top$ のハミング重みが t 以

表 4.2: 符号に基づく暗号の分類

文献	暗号化	鍵交換	署名
Classic McEliece [2]	○	○	–
BIKE [3]	○	○	–
HQC [34]	○	○	–

下であれば，線形符号の復号により，$\boldsymbol{e}' = \boldsymbol{e}\boldsymbol{Q}^\top$ を得る。よって，高い確率で復号に成功する。平文 $\boldsymbol{e}$ およびパリティ検査行列 $\tilde{\boldsymbol{H}}$ が一様ランダムであれば，暗号文 $\boldsymbol{d} \in \mathbb{F}_2^{n-k}$ はランダムな $n-k$ 次元のベクトルと見分けが付かないと考えられている。また，$\tilde{\boldsymbol{H}}$ が一様ランダムであり，適切な t が選択されていれば，暗号文は統計的にランダムなベクトルと見分けが付かないとされている。一方で，オリジナルの Niederreiter 暗号は適応的選択暗号文攻撃（CCA）に対して安全ではないため，次節で示すより安全な方式が提案されている。

4.3 符号に基づく主要な暗号方式

本稿では以下の暗号方式を取り上げる。いずれも NIST PQC 標準化プロジェクトにおいて第 4 ラウンドに進んだものである。

1. Classic McEliece: Niederreiter 暗号を採用し，符号の構成が非常に保守的という観点からこれを取り上げる。
2. BIKE: Niederreiter 暗号を採用し，QC-MDPC 符号を用いて鍵を圧縮している，という観点からこれを取り上げる。
3. HQC: 符号版の LPR 暗号（Lyubashevsky，Peikert，Regev が 2010 年に提案した Ring-LWE 問題に基づく暗号方式 [30] を LPN 問題に基づく方式に変更したもの）を採用，Quasi-Cyclic 符号を用いて鍵を圧縮している，という特徴からこれを取り上げる。

4.3.1 Classic McEliece

- 提案者: Albrecht, Bernstein, Chou, Cid, Gilcher, Lange, Maram, von Maurich, Misoczki, Niederhagen, Paterson, Persichetti, Peters, Schwabe, Sendrier, Szefer, Tjhai, Tomlinson, Wang
- 基本方式の説明: Niederreiter 暗号方式に基づいている。基本符号方式として $\mathbb{F}_2$ 上の Goppa 符号を利用している。（具体的な Goppa 符号の生成方法や符号化および復号の方法については提案方式の仕様書 [2] を参照のこと。）$q = 2^m$ とし，$n \leq q$ を用いる。2 以上の t を $mt < n$ となるように取り，$k = n - mt$ とする。

  鍵生成: 誤り訂正能力が t である Goppa 符号のパリティ検査行列 $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ をランダムに生成する。組織符号化し，$\tilde{\boldsymbol{H}} = [\boldsymbol{I}_{n-k} \mid \boldsymbol{T}]$ とする。公開鍵を $pk = \boldsymbol{T} \in \mathbb{F}_2^{(n-k)\times k}$ とする。符号生成に使ったパラメータを Γ（$\mathbb{F}_q$ 係数の t 次モニック既約多項式と互いに異なる $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_q$）とする。秘密鍵を $sk = \Gamma$ とする。

  暗号化 $\mathsf{Encrypt}(pk, \boldsymbol{e})$：入力を $\boldsymbol{e} \in \mathcal{S}_H(n,t)$ とする。$\tilde{\boldsymbol{H}} = [\boldsymbol{I}_{n-k} \mid \boldsymbol{T}]$ とし，暗号文として $\boldsymbol{c} = \tilde{\boldsymbol{H}}\boldsymbol{e} \in \mathbb{F}_2^{n-k}$ を出力する。

  復号 $\mathsf{Decrypt}(sk, \boldsymbol{c})$：ハミング重み t のベクトル $\boldsymbol{e}$ を復号する。

  1. $\boldsymbol{c}$ に k 個ゼロを加え，$\boldsymbol{v} = (\boldsymbol{c}, \boldsymbol{0}_k) \in \mathbb{F}_2^n$ を考える。

表 4.3: Classic McEliece のパラメータ。公開鍵長，秘密鍵長，暗号文長の単位はそれぞれ Byte とする。

パラメータ名	(m, n, t)	安全性レベル	公開鍵長	秘密鍵長	暗号文長	復号エラー率
mceliece348864	$(12, 3488, 64)$	レベル 1	$261,120$	$6,492$	96	0
mceliece460896	$(13, 4608, 96)$	レベル 3	$524,160$	$13,608$	156	0
mceliece6688128	$(13, 6688, 128)$	レベル 5	$1,044,992$	$13,932$	208	0
mceliece6960119	$(13, 6960, 119)$	レベル 5	$1,047,319$	$13,948$	194	0
mceliece8192128	$(13, 8192, 128)$	レベル 5	$1,357,824$	$14,120$	208	0

2. Goppa 符号の復号アルゴリズムを用いて，$\boldsymbol{v}$ と距離 t 以下にある符号語 $\boldsymbol{d}$ を計算する。（なければ $\perp$ を出力する。）
3. $\boldsymbol{e} = \boldsymbol{v} + \boldsymbol{d}$ とする。
4. $\mathsf{HW}(\boldsymbol{e}) = t$ かつ $\boldsymbol{c} = \tilde{\boldsymbol{H}}\boldsymbol{e}$ ならば $\boldsymbol{e}$ を出力する。（そうでなければ $\perp$ を出力する。）

- 鍵カプセル化方式の説明: 基本方式を決定性の公開鍵暗号とみなし，藤崎–岡本変換の変種をかけたものとみなせる。以下ではハッシュ関数 $\mathsf{H}\colon \{0,1\}^* \to \{0,1\}^{256}$ を用いる。

  鍵生成: ℓ ビットのシード δ から乱数を生成し，鍵生成を行う。（乱数の生成方法は省略する。）公開鍵は同じく $pk = \boldsymbol{T}$ である。n ビットの一様ランダムな文字列 $\boldsymbol{s}$ を生成する。秘密鍵は $sk = (\Gamma, \boldsymbol{s})$ である。

  鍵カプセル化: 1. $\boldsymbol{e} \leftarrow \mathcal{S}_H(n, t)$ をあるアルゴリズムに従ってランダム生成する。
  2. $\boldsymbol{c} = \mathsf{Encrypt}(pk, \boldsymbol{e})$ を計算する。
  3. $K = \mathsf{H}(1, \boldsymbol{e}, \boldsymbol{c})$ とする。
  4. 暗号文を $\boldsymbol{c}$ とし，セッション鍵 K を出力する。

  デカプセル化: 1. $b = 1$ とする。
  2. 受診した $\boldsymbol{c}$ に対して，$\boldsymbol{e} = \mathsf{Decrypt}(sk, \boldsymbol{c})$ とする。$\boldsymbol{e} = \perp$ であれば，$b = 0, \boldsymbol{e} = \boldsymbol{s}$ と上書きする。
  3. $K = \mathsf{H}(b, \boldsymbol{e}, \boldsymbol{c})$ を計算する。
  4. K を出力する。

  以上より，セッション鍵（共通鍵）K を安全に共有することができる。

パラメータセットとして mceliece348864，mceliece348864f，mceliece460896，mceliece460896f，mceliece6688128，mceliece6688128f, mceliece6960119, mceliece6960119f, mceliece8192128, mceliece8192128f が提案されている。表 4.3 に鍵カプセル化方式のパラメータ，鍵長および暗号文長，想定セキュリティレベル，復号エラー率をまとめた。今回末尾に f が付くものは扱っていないが，鍵長・暗号文長は f 無しのものと同一である。Classic McEliece は公開鍵長が非常に大きく，レベル 5 では 1 メガバイトを超える。一方で，暗号文長は非常に小さく，格子暗号に基づく FIPS 標準（FIPS 203）である ML-KEM [38] の暗号文サイズよりも小さい。例えば，[38, Table 3] によれば，ML-KEM のレベル 1 の公開鍵長は 800 Byte，秘密鍵長は 1632 Byte，暗号文長は 768 Byte となっている。

mceliece348864 の速度に関しては，鍵生成に必要な平均 CPU サイクル数が $60,333,686$ Cycle，鍵カプセル化が $37,585$ Cycle，デカプセル化が $127,668$ Cycle である。参考までに，ML-KEM（Kyber-512）の速度は，鍵生成が $33,428$ Cycle，鍵カプセル化が $49,184$ Cycle，デカプセル化が $40,564$ Cycle である [38]。なお，いずれも Haswell CPU 搭載のサーバ上で AVX 命令を使用した C 言語実装を動作させた時の記録である。他のパラメータに関しては，仕様書を参照されたい。

4.3.2 BIKE

- 提案者: Aragon, Barreto, Bettaieb, Bidoux, Blazy, Deneuville, Gaborit, Gueron, Güneysu, Aguilar Melchor, Misoczki, Persichetti, Sendrier, Tillich, Zémor, Vasseur, Ghosh, Richter-Brokmann
- 基本方式の説明: Niederreiter 暗号方式に基づいている。基本となる符号に QC-MDPC 符号を採用し，公開鍵サイズを圧縮している。そのため，鍵や暗号化は格子暗号の一種の NTRU 暗号と非常に近い形をしている点が特徴である。具体的な符号化および復号の方法については提案方式の仕様書 [3] を参照のこと。以下では，$\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1)$ とする。

  鍵生成: $\boldsymbol{h}_0 \in \mathcal{R}$ および $\boldsymbol{h}_1 \in \mathcal{R}$ を $\mathcal{S}_H(n, w/2)$ から一様ランダムに選ぶ。$\boldsymbol{h} = \boldsymbol{h}_1/\boldsymbol{h}_0 \in \mathcal{R}$ とする。$(\boldsymbol{h}_0, \boldsymbol{h}_1)$ を QC-MDPC 符号のパリティ検査行列とし，$(\mathbf{1}, \boldsymbol{h})$ をその組織符号化したものとみなすことができる。公開鍵を $pk = \boldsymbol{h}$ とし，秘密鍵を $sk = (\boldsymbol{h}_0, \boldsymbol{h}_1)$ とする。

  暗号化 $\mathsf{Encrypt}(pk, (\boldsymbol{e}_0, \boldsymbol{e}_1))$: $(\boldsymbol{e}_0, \boldsymbol{e}_1) \in \mathcal{R}^2$ を $\mathcal{S}_H(2n, t)$ 中のベクトルとみなす。$\boldsymbol{c} = \boldsymbol{e}_0 + \boldsymbol{e}_1\boldsymbol{h} \in \mathcal{R}$ を出力する。

  復号 $\mathsf{Decrypt}(sk, \boldsymbol{c})$: ハミング重み t 以下のベクトル $(\boldsymbol{e}_0, \boldsymbol{e}_1)$ を復号する。

  1. $\boldsymbol{c}\boldsymbol{h}_0$ を計算する。
  2. QC-MDPC 符号の復号アルゴリズムを用いて，$\boldsymbol{c}\boldsymbol{h}_0$ をシンドロームとするベクトル $(\boldsymbol{e}_0, \boldsymbol{e}_1)$ を計算する。
- 鍵カプセル化方式の説明: 基本方式を決定性公開鍵暗号方式とみなす。基本方式とハッシュ関数 $\mathsf{L}\colon \{0,1\}^* \to \{0,1\}^{256}$ を用いて，平文 $\boldsymbol{m} \in \{0,1\}^{256}$ と乱数 $(\boldsymbol{e}_0, \boldsymbol{e}_1)$ に対して暗号化 $(\boldsymbol{c}_0 = \mathsf{Encrypt}(pk, (\boldsymbol{e}_0, \boldsymbol{e}_1)))$ および $\boldsymbol{m}$ のマスキング $(\boldsymbol{c}_1 = \boldsymbol{m} \oplus \mathsf{L}(\boldsymbol{e}_0, \boldsymbol{e}_1))$ とを行う IND-CPA 安全な乱択公開鍵暗号を構成する。鍵カプセル化方式は，この乱択公開鍵暗号に藤崎–岡本変換の変種を適用したものとみなせる。以下ではハッシュ関数 $\mathsf{H}, \mathsf{L}\colon \{0,1\}^* \to \{0,1\}^{256}$ と $\mathsf{G}\colon \{0,1\}^* \to \mathcal{S}_H(2n, t)$ を用いる。

  鍵生成: 適切な長さのシード δ から乱数を生成し，鍵生成を行う。公開鍵は同じく $pk = \boldsymbol{h}$ である。ℓ ビットの一様ランダムな文字列 $\boldsymbol{s} \in \{0,1\}^\ell$ を生成する。秘密鍵は $sk = (\boldsymbol{h}_0, \boldsymbol{h}_1, \boldsymbol{s})$ である。

  鍵カプセル化: 1. $\boldsymbol{m} \leftarrow \{0,1\}^{256}$ を一様ランダムに選ぶ。
  2. $(\boldsymbol{e}_0, \boldsymbol{e}_1) = \mathsf{G}(\boldsymbol{m})$ を計算する。
  3. $\boldsymbol{c}_0 = \mathsf{Encrypt}(pk, (\boldsymbol{e}_0, \boldsymbol{e}_1))$ と，$\boldsymbol{c}_1 = \boldsymbol{m} \oplus \mathsf{L}(\boldsymbol{e}_0, \boldsymbol{e}_1)$ を計算する。
  4. $K = \mathsf{H}(\boldsymbol{m}, \boldsymbol{c})$ を計算する。
  5. 暗号文を $C = (\boldsymbol{c}_0, \boldsymbol{c}_1)$ とし，セッション鍵 K を出力する。

  デカプセル化: 1. 受信した C に対して，$(\boldsymbol{e}'_0, \boldsymbol{e}'_1) = \mathsf{Decrypt}(sk, \boldsymbol{c}_0)$ を計算する。
  2. 復号に失敗したら，$\perp$ を出力して停止する。
  3. $\boldsymbol{m}' = \boldsymbol{c}_1 \oplus \mathsf{L}(\boldsymbol{e}'_0, \boldsymbol{e}'_1)$ を計算する。
  4. $(\boldsymbol{e}'_0, \boldsymbol{e}'_1) = \mathsf{G}(\boldsymbol{m}')$ ならば，$K = \mathsf{H}(\boldsymbol{m}', \boldsymbol{c})$ を出力して停止する。
  5. そうでなければ，$K = \mathsf{H}(\boldsymbol{s}, \boldsymbol{c})$ を計算し，出力する。

  以上より，セッション鍵（共通鍵）K を安全に共有することができる。

表 4.4 に鍵カプセル化方式のパラメータ，鍵長，暗号文長および復号エラー率をまとめた。3 つのパラメータセットがそれぞれレベル 1，3，5 相当として提案された。BIKE-Level1 の速度に関しては，鍵生成が 589,000 Cycle，鍵カプセル化が 97,000 Cycle，デカプセル化が 1,135,000 Cycle である（Skylake CPU 搭載のサーバ，AVX 命令を使用）。

表 4.4: BIKE のパラメータ。公開鍵長，秘密鍵長，暗号文長の単位はそれぞれ Byte とする。

パラメータ名	(n, w, t)	安全性レベル	公開鍵長	秘密鍵長	暗号文長	復号エラー率
BIKE-Level1	$(12323, 142, 134)$	レベル 1	$1,541$	281	$1,573$	2^{-128}
BIKE-Level3	$(24659, 206, 199)$	レベル 3	$3,083$	419	$3,115$	2^{-192}
BIKE-Level5	$(40973, 274, 264)$	レベル 5	$5,122$	580	$5,154$	2^{-256}

他のパラメータに関しては，仕様書を参照されたい。

4.3.3 HQC

- 提案者: Aguilar Melchor，Aragon，Bettaieb，Bidoux，Blazy，Deneuville，Gaborit，Persichetti，Zémor，Bos，Dion，Lacan，Robert，Veron
- 基本方式の説明: 符号版の LPR 暗号に基づき，公開鍵暗号を構成している。
  $\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1)$ とする。$n' = n_1 n_2$ とし，$[n', k]$ 線形符号 $\mathcal{C}$ を採用する。具体的な符号化および復号の方法については提案方式の仕様書 [34] を参照のこと。線形符号 $\mathcal{C}$ の符号化・復号アルゴリズムを $\mathsf{encode}, \mathsf{decode}$ とする。$n \geq n'$ を仮定する。以下では，暗号文の第二要素 $\boldsymbol{v}$ を $\mathcal{R}$ 要素（n ビットベクトル）として扱っているが，実際には n' ビットに縮めて用いる。

  鍵生成: $\boldsymbol{x} \in \mathcal{R}$ および $\boldsymbol{y} \in \mathcal{R}$ を $\mathcal{S}_H(n, w)$ から一様ランダムに選び，$\boldsymbol{h} \leftarrow \mathcal{R}$ に対して公開鍵を $pk = (\boldsymbol{h}, \boldsymbol{s}) \in \mathcal{R}^2$ とし，秘密鍵を $sk = (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{R}^2$ とする。

  暗号化 $\mathsf{Encrypt}(pk, \boldsymbol{m}, \boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{e})$: $\boldsymbol{r}_1 \in \mathcal{R}$ および $\boldsymbol{r}_2 \in \mathcal{R}$ を $\mathcal{S}_H(n, w_r)$ から一様ランダムに選び，$\boldsymbol{e} \in \mathcal{R}$ を $\mathcal{S}_H(n, w_e)$ から一様ランダムに選ぶ。$\boldsymbol{u} = \boldsymbol{r}_1 + \boldsymbol{h} \cdot \boldsymbol{r}_2$ および $\boldsymbol{v} = \mathsf{encode}(m) + \boldsymbol{s} \cdot \boldsymbol{r}_2 + \boldsymbol{e}$ を計算する。$\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{v})$ を暗号文として出力する。

  復号 $\mathsf{Decrypt}(sk, \boldsymbol{c})$: $\mathsf{decode}(\boldsymbol{v} - \boldsymbol{u} \cdot \boldsymbol{y})$ を出力する。
- 鍵カプセル化方式: 基本方式を乱択な公開鍵暗号とみなし，藤崎–岡本変換の変種を適用したものとみなせる。以下ではハッシュ関数 $\mathsf{H}, \mathsf{H}' \colon \{0,1\}^* \to \{0,1\}^{256}$ を用いる。また，XOF [*5] として $\mathsf{H}_G \colon \{0,1\}^* \to \{0,1\}^*$ も用いる。（第 4 ラウンドで G への入力に $\mathrm{seed} \in \{0,1\}^{128}$ と $\mathrm{salt} \in \{0,1\}^{128}$ が追加された。）

  鍵生成: 同上。ただし $\boldsymbol{h}$ の生成をシード seed から行うこととし，公開鍵を $pk = (\boldsymbol{s}, \mathrm{seed})$ とする。また，秘密鍵にもシードを加え，$sk = (\boldsymbol{x}, \boldsymbol{y}, \mathrm{seed})$ とする。

  鍵カプセル化:
  1. $\boldsymbol{m} \leftarrow \mathbb{F}_2^k$ を一様ランダムにとる。
  2. $\mathrm{salt} \leftarrow \mathbb{F}_2^{128}$ を一様ランダムにとる。
  3. $\theta = \mathsf{H}_G(\boldsymbol{m}, \mathrm{seed}, \mathrm{salt})$ を計算する。θ から $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{e}$ を生成する。
  4. $\boldsymbol{c} = \mathsf{Encrypt}(pk, \boldsymbol{m}, \boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{e})$ を計算する。$\boldsymbol{d} = \mathsf{H}'(\boldsymbol{m})$ とする。$K = \mathsf{H}(\boldsymbol{m}, \boldsymbol{c})$ とする。
  5. 暗号文を $C = (\boldsymbol{c}, \boldsymbol{d}, \mathrm{salt})$ とし，セッション鍵 K を出力する。

  デカプセル化:
  1. 受信した C に対して，$\boldsymbol{m}' = \mathsf{Decrypt}(sk, \boldsymbol{c})$ を計算する。
  2. $\theta' = \mathsf{H}_G(\boldsymbol{m}', \mathrm{seed}, \mathrm{salt})$ を計算する。θ' から $\boldsymbol{r}_1', \boldsymbol{r}_2', \boldsymbol{e}'$ を生成する。
  3. $\boldsymbol{c}' = \mathsf{Encrypt}(pk, \boldsymbol{m}', \boldsymbol{r}_1', \boldsymbol{r}_2', \boldsymbol{e}')$ を計算する。$\boldsymbol{c} \neq \boldsymbol{c}'$ もしくは $\boldsymbol{d} \neq \boldsymbol{d}'$ ならば $\perp$ を出力して停止する。
  4. $K = \mathsf{H}(\boldsymbol{m}, \boldsymbol{c})$ を出力する。

[*5] eXtendable-Output Function の略。SHAKE128 や SHAKE256 が例として知られている。

表 4.5: HQC のパラメータ。公開鍵長，秘密鍵長，暗号文長の単位はそれぞれ Byte とする。

パラメータ名	$(n_1, n_2, n, w, w_r = w_e)$	安全性レベル	公開鍵長	秘密鍵長	暗号文長	復号エラー率
hqc-128	$(46, 384, 17669, 66, 75)$	レベル 1	$2,249$	40	$4,497$	2^{-128}
hqc-192	$(56, 640, 35851, 100, 114)$	レベル 3	$4,522$	40	$9,042$	2^{-192}
hqc-256	$(90, 640, 57637, 131, 149)$	レベル 5	$7,245$	40	$14,485$	2^{-256}

以上より，セッション鍵（共通鍵）K を安全に共有することができる。

3 つのパラメータセットがそれぞれレベル 1，3，5 相当として提案された。表 4.5 に鍵カプセル化方式のパラメータ，鍵長，暗号文長および復号エラー率をまとめた。表中では，秘密鍵はシードだけ記憶していることにされており，40 バイトしかない。また公開鍵の $\boldsymbol{h}$ の部分もシードから再生成されることと定義されている点に注意されたい。hqc-128 の速度に関しては，鍵生成が $87,000$ Cycle，鍵カプセル化が $204,000$ Cycle，デカプセル化が $362,000$ Cycle である（Skylake CPU 搭載のデスクトップ PC，AVX 命令を使用）。他のパラメータに関しては，仕様書を参照されたい。

4.4 符号に基づく暗号技術に関するまとめ

基本となる McEliece 暗号方式は，McEliece により 40 年以上前に提案されており，パラメータは改訂されているものの，いまだに破られていない。Classic McEliece などのように，公開鍵や秘密鍵は長いものの，暗号文は短い方式が多い。LPN 問題は学習理論や符号理論から派生した問題であり，SD 問題は LPN 問題の特殊な場合である。誤り確率 τ が十分大きい場合の LPN 問題や，重み w が一定の大きさの SD 問題を確率的多項式時間で効率的に解くことは，量子コンピュータを用いても困難であると予想されている。

共通鍵暗号や公開鍵暗号の分野で多くの方式が LPN 問題や SD 問題に基づいて提案されている。LWE 問題と比較した場合，利点としては，

- $\mathbb{F}_2$ およびその拡大体を基に構成するため，ハードウェア構成との相性が良い点
- 誤差分布としてベルヌーイ分布やその一般化した分布を用いるため，誤差のサンプリングが容易である点

が挙げられる。一方，欠点として，

- 鍵や暗号文のサイズが大きくなりやすい点
- 符号の復号アルゴリズムが複雑になりがちな点
- 完全準同型暗号といった発展的な応用が少ない点

が挙げられる。暗号方式のパラメータ設定の際には，4.1 節で挙げたさまざまなアルゴリズムを考慮する必要がある。アルゴリズムの高速化について盛んに研究されており，動向を注視する必要がある。また，符号に基づく暗号技術の信頼性を向上させるためには，理論面における研究だけではなく，実時間の計算量に関する研究も重要である。公開鍵や秘密鍵を圧縮しようと特殊な符号を採用したり，距離の定義を変える提案も多くある。これらは解読攻撃を受けることも多く，評価が確定していない暗号・署名方式については注視が必要である。

第 4 章の参照文献

[1] CRYPTREC 暗号技術調査 WG (耐量子計算機暗号). CRYPTREC 耐量子計算機暗号の研究動向調査報告書. CRYPTREC GL-2004-2022, https://www.cryptrec.go.jp/report/cryptrec-tr-2001-2022.pdf. 2023-03.

[2] M. R. Albrecht et al. Classic McEliece: conservative code-based cryptography. https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/mceliece-Round4.tar.gz. 2022-10. (2024-03-05 閲覧).

[3] N. Aragon et al. BIKE: Bit flipping key encapsulation (Round 4 submission). https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/BIKE-Round4.zip. 2022-10. (2024-03-05 閲覧).

[4] S. Arora, R. Ge. New Algorithms for Learning in Presence of Errors. ICALP (1). Vol. 6755. Lecture Notes in Computer Science. Springer, 2011, pp. 403–415.

[5] A. Becker, A. Joux, A. May, A. Meurer. Decoding Random Binary Linear Codes in $2^{n/20}$: How $1 + 1 = 0$ Improves Information Set Decoding. EUROCRYPT. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 520–536.

[6] E. R. Berlekamp, R. J. McEliece, H. C. A. van Tilborg. On the inherent intractability of certain coding problems (Corresp.) IEEE Trans. Inf. Theory. Vol. 24, Num. 3 (1978), pp. 384–386.

[7] A. Blum, M. L. Furst, M. J. Kearns, R. J. Lipton. Cryptographic Primitives Based on Hard Learning Problems. CRYPTO. Vol. 773. Lecture Notes in Computer Science. Springer, 1993, pp. 278–291.

[8] A. Blum, A. Kalai, H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM. Vol. 50, Num. 4 (2003), pp. 506–519.

[9] S. Bogos, S. Vaudenay. Optimization of LPN Solving Algorithms. ASIACRYPT (1). Vol. 10031. Lecture Notes in Computer Science. 2016, pp. 703–728.

[10] L. Both, A. May. Decoding Linear Codes with High Error Rate and Its Impact for LPN Security. PQCrypto. Vol. 10786. Lecture Notes in Computer Science. Springer, 2018, pp. 25–46.

[11] L. Both, A. May. Optimizing BJMM with nearest neighbors: Full decoding in $2^{2n/21}$ and McEliece security. Workshop on Coding and Cryptography. 2017. https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/paper/bjmm+.pdf.

[12] K. Carrier, T. Debris-Alazard, C. Meyer-Hilfiger, J.-P. Tillich. Statistical Decoding 2.0: Reducing Decoding to LPN. ASIACRYPT (4). Vol. 13794. Lecture Notes in Computer Science. Springer, 2022, pp. 477–507.

[13] J. Carrijo, R. Tonicelli, H. Imai, A. C. A. Nascimento. A Novel Probabilistic Passive Attack on the Protocols HB and HB$^+$. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. Vol. 92-A, Num. 2 (2009), pp. 658–662.

[14] C. Chevignard, P.-A. Fouque, A. Schrottenloher. Reducing the Number of Qubits in Quantum Information Set Decoding. ASIACRYPT (8). Vol. 15491. Lecture Notes in Computer Science. Springer, 2024, pp. 299–329.

[15] L. Ducas, A. Esser, S. Etinski, E. Kirshanova. Asymptotics and Improvements of Sieving for Codes. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 151–180.

[16] A. Esser. Revisiting Nearest-Neighbor-Based Information Set Decoding. IMACC. Vol. 14421. Lecture Notes in Computer Science. Springer, 2023, pp. 34–54.

[17] A. Esser, F. Heuer, R. Kübler, A. May, C. Sohler. Dissection-BKW. CRYPTO (2). Vol. 10992. Lecture Notes in Computer Science. Springer, 2018, pp. 638–666.

[18] A. Esser, R. Kübler, A. May. LPN Decoded. CRYPTO (2). Vol. 10402. Lecture Notes in Computer Science. Springer, 2017, pp. 486–514.

[19] A. Esser, S. Ramos-Calderer, E. Bellini, J. I. Latorre, M. Manzano. Hybrid Decoding – Classical-Quantum Trade-Offs for Information Set Decoding. PQCrypto. Vol. 13512. Lecture Notes in Computer Science. Springer, 2022, pp. 3–23.

[20] A. Esser, J. A. Verbel, F. Zweydinger, E. Bellini. SoK: CryptographicEstimators – a Software Library for Cryptographic Hardness Estimation. AsiaCCS. ACM, 2024.

[21] A. Esser, F. Zweydinger. New Time-Memory Trade-Offs for Subset Sum – Improving ISD in Theory and Practice. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 360–390.

[22] Q. Guo, T. Johansson, C. Löndahl. Solving LPN Using Covering Codes. J. Cryptol. Vol. 33, Num. 1 (2020), pp. 1–33.

[23] J. Håstad. Some optimal inapproximability results. J. ACM. Vol. 48, Num. 4 (2001), pp. 798–859.

[24] G. Kachigar, J.-P. Tillich. Quantum Information Set Decoding Algorithms. PQCrypto. Vol. 10346. Lecture Notes in Computer Science. Springer, 2017, pp. 69–89.

[25] P. Kirchner. Improved Generalized Birthday Attack. Cryptology ePrint Archive, Paper 2011/377. 2011. `https://eprint.iacr.org/2011/377`.

[26] E. Kirshanova. Improved Quantum Information Set Decoding. PQCrypto. Vol. 10786. Lecture Notes in Computer Science. Springer, 2018, pp. 507–527.

[27] P. J. Lee, E. F. Brickell. An Observation on the Security of McEliece's Public-Key Cryptosystem. EUROCRYPT. Vol. 330. Lecture Notes in Computer Science. Springer, 1988, pp. 275–280.

[28] É. Levieil, P.-A. Fouque. An Improved LPN Algorithm. SCN. Vol. 4116. Lecture Notes in Computer Science. Springer, 2006, pp. 348–359.

[29] Y. Li, R. H. Deng, X. Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. IEEE Trans. Inf. Theory. Vol. 40, Num. 1 (1994), pp. 271–273.

[30] V. Lyubashevsky, C. Peikert, O. Regev. On Ideal Lattices and Learning with Errors over Rings. EUROCRYPT. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 1–23.

[31] A. May, A. Meurer, E. Thomae. Decoding Random Linear Codes in $\tilde{\mathcal{O}}(2^{0.054n})$. ASIACRYPT. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, pp. 107–124.

[32] A. May, I. Ozerov. On Computing Nearest Neighbors with Applications to Decoding of Binary Linear Codes. EUROCRYPT (1). Vol. 9056. Lecture Notes in Computer Science. Springer, 2015, pp. 203–228.

[33] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report. Vol. 44 (1978), pp. 114–116.

[34] C. Aguilar Melchor et al. Hamming Quasi-Cyclic (HQC) – Fourth round version (Updated version 01/10/2022). https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/HQC-Round4.zip. 2022-10. (2024-03-05 閲覧).

[35] S. Narisada, K. Fukushima, S. Kiyomoto. Multiparallel MMT: Faster ISD Algorithm Solving High-Dimensional Syndrome Decoding Problem. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. Vol. 106, Num. 3 (2023), pp. 241–252.

[36] S. Narisada, S. Uemura, H. Okada, H. Furue, Y. Aikawa, K. Fukushima. Solving McEliece-1409 in One Day – Cryptanalysis with the Improved BJMM Algorithm. ISC (2). Vol. 15258. Lecture Notes in Computer Science. Springer, 2024, pp. 3–23.

[37] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. Problemy Upravleniia i Teorii Informatsii (Problems of Control and Information Theory). Vol. 15, Num. 2 (1986), pp. 157–166.

[38] NIST. Module-Lattice-Based Key-Encapsulation Mechanism Standard. NIST FIPS 203, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf. 2024-08.

[39] S. Perriello, A. Barenghi, G. Pelosi. A Complete Quantum Circuit to Solve the Information Set Decoding Problem. QCE. IEEE, 2021, pp. 366–377.

[40] S. Perriello, A. Barenghi, G. Pelosi. Improving the Efficiency of Quantum Circuits for Information Set Decoding. ACM Transactions on Quantum Computing. Vol. 4, Num. 4 (2023). https://doi.org/10.1145/3607256.

[41] E. Prange. The use of information sets in decoding cyclic codes. IRE Trans. Inf. Theory. Vol. 8, Num. 5 (1962), pp. 5–9.

[42] J. Stern. A method for finding codewords of small weight. Coding Theory and Applications. Vol. 388. Lecture Notes in Computer Science. Springer, 1988, pp. 106–113.

[43] B. Zhang, L. Jiao, M. Wang. Faster Algorithms for Solving LPN. EUROCRYPT (1). Vol. 9665. Lecture Notes in Computer Science. Springer, 2016, pp. 168–195.

[44] 電子情報通信学会. 知識ベース 知識の森 1 群 (信号・システム) 2 編 (符号理論). https://www.ieice-hbkb.org/portal/01-2/01_02/. (2024-03-05 閲覧).

第 5 章

多変数多項式に基づく暗号技術

多変数公開鍵暗号（Multivariate Public Key Cryptosystems）における暗号方式の特徴は，有限体上の多変数多項式を用いた連立方程式

$$\begin{cases} p_1(x_1, x_2, \ldots, x_n) = 0, \\ p_2(x_1, x_2, \ldots, x_n) = 0, \\ \qquad\qquad \vdots \\ p_m(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

の求解問題（MP 問題）を解く計算の困難性が安全性の根拠として必要ということである。連立線形方程式は多項式時間で求解可能であるから，多変数公開鍵暗号に現れる MP 問題における多項式の最大次数は 2 次以上に限定される。本報告書では，多変数公開鍵暗号の多くの方式で採用されている双極型システムを中心に解説する。

5.1 多変数多項式に基づく暗号技術の安全性の根拠となる問題

$\mathbb{F}_q$ で位数 q の有限体を表し，$\mathbf{x} = (x_1, x_2, \ldots, x_n)$ で（代数的に独立な）変数の集合を表すものとする。$\mathbf{x}$ に関する $\mathbb{F}_q$ 上の多変数多項式の組，すなわち，多変数多項式 $p_i(\mathbf{x})$ $(i = 1, \ldots, m)$ により，$P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ と表されるものを（$\mathbb{F}_q$ 上の）多変数多項式系と呼ぶことにする。この多変数多項式系 $P(\mathbf{x})$ は代入評価により，$\mathbb{F}_q^n$ から $\mathbb{F}_q^m$ への写像を構成する。この（多変数多項式）写像を $P : \mathbb{F}_q^n \to \mathbb{F}_q^m$ と表すことにする。

5.1.1 MP 問題（MQ 問題）

MP 問題は次のように定義される。

MP 問題 多変数多項式系 $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ と $\mathbf{d} = (d_1, d_2, \ldots, d_m) \in \mathbb{F}_q^m$ に対して，変数 $\mathbf{x}$ に関する連立方程式

$$\begin{cases} p_1(x_1, x_2, \ldots, x_n) = d_1, \\ p_2(x_1, x_2, \ldots, x_n) = d_2, \\ \qquad\qquad \vdots \\ p_m(x_1, x_2, \ldots, x_n) = d_m \end{cases} \tag{5.1}$$

の解（が存在するなら）少なくとも 1 つ求めよ。

連立方程式 (5.1) の右辺の各 d_i を左辺に移項して $p_i(\mathbf{x})$ に吸収させることができるので，右辺を 0 として MP 問題を表現する場合もある。MP 問題において，$P(\mathbf{x})$ の全ての成分 $p_i(\mathbf{x})$ が 1 次以下となる場合，MP 問題は単に線形方程

式を解く問題となり，ガウスの消去法などで m, n に関し多項式時間で求解することが可能である。よって，MP 問題を考える場合は通常，各 $p_i(\mathbf{x})$ の次数は 2 以上であると仮定する。特に，$p_i(\mathbf{x})$ の次数が全て 2 となるとき，MP 問題は MQ 問題と呼ばれる。$\mathbb{F}_q = \mathbb{F}_2$ の場合，MQ 問題は NP 完全であることが知られている [14]。

MQ 問題を解くコンテストとして Fukuoka MQ challenge が知られている。扱われている MQ 問題は，有限体は $q = 2, 31, 256$ の 3 種類と m, n に関しては $m = 2n,\ n \approx 1.5m$ の 2 種類の計 6 種類である。投稿され解かれた問題の (m, n) の値の最大は表 5.1 のようになっている。

表 5.1: Fukuoka MQ challenge で解かれた MQ 問題のパラメータの最大値（2024/9/30 時点）

タイプ	I	II	III	IV	V	VI
$\mathbb{F}_q$	$\mathbb{F}_2$	$\mathbb{F}_{31}$	$\mathbb{F}_{256}$	$\mathbb{F}_2$	$\mathbb{F}_{31}$	$\mathbb{F}_{256}$
(m, n)	$m = 2n$	$m = 2n$	$m = 2n$	$n \approx 1.5m$	$n \approx 1.5m$	$n \approx 1.5m$
(m, n) の最大	$(166, 83)$	$(74, 37)$	$(76, 38)$	$(76, 114)$	$(20, 30)$	$(22, 33)$

5.1.2 MinRank 問題

MinRank 問題 正の整数 r と行列 $\mathrm{M}_1, \ldots, \mathrm{M}_k \in \mathbb{F}_q^{m \times n}$ に対し，$\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q$ で，$(\alpha_1, \ldots, \alpha_k) \neq (0, \ldots, 0)$ かつ

$$\mathrm{Rank}\left(\sum_{i=1}^{k} \alpha_i \mathrm{M}_i\right) \leq r$$

なるものを求めよ。（Rank(M) は行列 M のランクを表す。）

MinRank 問題は MP 問題に帰着できることが知られている [19, 10, 3]。また，MinRank 問題を解く計算の困難性をベースとした署名方式などがいくつか提案されている [8, 4, 23, 1]。

5.1.3 IP 問題，EIP 問題

IP（Isomorphism of Polynomials）問題は以下のように定義される。

IP 問題 S, T をそれぞれ，$\mathbb{F}_q^n$, $\mathbb{F}_q^m$ 上のアフィン同型写像とする。多変数多項式系 $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ に対し，多変数多項式系 $\tilde{P}(\mathbf{x})$ を合成により，$\tilde{P}(\mathbf{x}) = T \circ P(\mathbf{x}) \circ S$ で定める。このとき，$P(\mathbf{x})$, $\tilde{P}(\mathbf{x})$ の情報から S, T を求めよ。

IP 問題において，S や T の行列成分やベクトル成分をすべて独立な変数と見た場合，等式 $\tilde{P}(\mathbf{x}) = T \circ P(\mathbf{x}) \circ S$ は連立多項式方程式と見ることができる。すなわち，IP 問題は MP 問題に変換される。

多変数多項式系のクラス $\mathcal{C}$ を 1 つ固定する。ここで多変数多項式系のクラスとは多変数多項式系の集合 $\mathbb{F}_q[\mathbf{x}]^m$ の部分集合のことである。このとき，（クラス $\mathcal{C}$ に関する）EIP（Extended Isomorphism of Polynomials）問題は以下のように定義される。

EIP 問題 多変数多項式系 $\tilde{P}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ は，$\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T とクラス $\mathcal{C}$ に属する多変数多項式系 $P(\mathbf{x})$ により，$\tilde{P}(\mathbf{x}) = T \circ P(\mathbf{x}) \circ S$ で表されるとする。このとき，分解 $\tilde{P}(\mathbf{x}) = T' \circ P'(\mathbf{x}) \circ S'$ で，S', T' は $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像，$P'(\mathbf{x}) \in \mathcal{C}$ なるものを見つけよ。

$\mathcal{C} = \{P(\mathbf{x})\}$ に関する EIP 問題が通常の IP 問題であるから，EIP 問題は IP 問題の拡張である。5.2 節で述べるように，EIP 問題は双極型システムで構成される暗号化方式，署名方式の鍵復元攻撃に対する安全性に関わる。EIP 問題を解く方法はクラス $\mathcal{C}$ の取り方（あるいは方式）に依存する。

5.2 多変数多項式に基づく代表的な暗号方式の説明

5.2.1 双極型システム

IP 問題ベース [22] や MinRank 問題ベース [8, 1, 4, 23] の方式も存在するが，多変数公開鍵暗号の多くの方式が MP 問題をベースとして構成されている。中でも双極型システム [9] と呼ばれる構成方法が多く利用されているため，この構成方法について説明する。（1 次多項式で構成されてなくても）多変数多項式系 $P(\mathbf{x})$ によっては，多くの $\mathbf{d} \in \mathbb{F}_q^m$ に対して MP 問題が効率的に計算できる場合がある。例えば，$n = m$ とし，$P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ が三角型多変数多項式系である，すなわち，

$$
\begin{aligned}
&p_1(\mathbf{x}) = x_1,\\
&p_2(\mathbf{x}) = x_2 + g_2(x_1) \quad (g_2(x_1) \in \mathbb{F}_q[x_1]),\\
&p_3(\mathbf{x}) = x_3 + g_3(x_1, x_2) \quad (g_2(x_1, x_2) \in \mathbb{F}_q[x_1, x_2]),\\
&\qquad \vdots\\
&p_m(\mathbf{x}) = x_m + g_m(x_1, \ldots, x_{m-1}) \quad (g_m(x_1, \ldots, x_{m-1}) \in \mathbb{F}_q[x_1, \ldots, x_{m-1}])
\end{aligned}
$$

の形で表されるとすると，任意の $\mathbf{d} \in \mathbb{F}_q^m$ に対して $P(\mathbf{x}) = \mathbf{d}$ の（唯 1 つの）解が，x_1 から逐次的に求められることが分かる。このことはすなわち，多変数多項式系のクラス $\mathcal{C}$ を三角型多変数多項式系の全体で定めると，任意の $P \in \mathcal{C}$ に対して，$P(\mathbf{x}) = \mathbf{d}$ $(\mathbf{d} \in \mathbb{F}_q^m)$ の解が効率的に計算可能ということである。

双極型システムでは，まず，MP 問題が効率的に計算できる多変数多項式系のクラス $\mathcal{C}_{\mathrm{cent}}$ を見つけ固定する。（例えば，$\mathcal{C}_{\mathrm{cent}}$ として三角型多変数多項式系の集合を取れる。）$G(\mathbf{x}) \in \mathcal{C}_{\mathrm{cent}}$ と $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T をそれぞれ任意にとり，これらを合成した多変数多項式系 $F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ をトラップドア付き一方向関数として利用するのが，双極型システムのアイデアである。ただし，$F(\mathbf{x})$ が実際にトラップドア付き一方向関数となるかどうかは $\mathcal{C}_{\mathrm{cent}}$ のとり方に依存する。

双極型システムの鍵生成は次のように行う。

鍵生成

1. $G(\mathbf{x}) \in \mathcal{C}_{\mathrm{cent}}$ をランダムに選ぶ。
2. $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T をランダムに選ぶ。
3. $F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ とする。

このとき，公開鍵は $F(\mathbf{x})$，秘密鍵は $G(\mathbf{x}), S, T$ となる。$F(\mathbf{x})$ はその係数集合が公開鍵として保管される。また，$G(\mathbf{x})$ を（この方式の）**中心写像**とよぶ。中心写像のクラス $\mathcal{C}_{\mathrm{cent}}$ は 2 次の多変数多項式系で構成されることが多い。これは，公開鍵長（や秘密鍵長）を出来るだけ小さくするためである。双極型システムは暗号化方式，署名方式両方の構成に用いることができる。

暗号化方式の暗号化・復号は次のように行う。

暗号化 平文 $M \in \mathbb{F}_q^n$ に対し，$C = F(M)$ を計算する。C が暗号文となる。

復号 暗号文 $C \in \mathbb{F}_q^m$ に対し，(1) $B_1 = T^{-1}(C)$, (2) $G(B_2) = B_1$ なる B_2 を計算, (3) $M' = S^{-1}(B_2)$ の順に計算

する。M' が平文と一致する。

復号が成功するためには，$G(\mathbf{x})$（あるいは $F(\mathbf{x})$）が単射である必要がある。単射の条件を少し緩めて，「$G(\mathbf{x})$（あるいは $F(\mathbf{x})$）の逆像の個数が十分少ない」とすることもできる。この場合，M' が複数得られることになるので，ハッシュ値などを用いて平文 M と一致する M' を特定する。

双極型システムの署名方式の署名生成・検証は次のように行う。

署名生成 メッセージ（のハッシュ値） $M \in \mathbb{F}_q^m$ に対し，(1) $B_1 = T^{-1}(M)$, (2) $G(B_2) = B_1$ なる B_2 を計算, (3) $\sigma = S^{-1}(B_2)$ の順に計算する。σ が署名となる。

検証 署名 $\sigma \in \mathbb{F}_q^n$ に対し，$M' = F(\sigma)$ を計算する。$M = M'$ ならば署名を受理，それ以外は棄却する。

署名生成がいつでも実行できるためには，どのような $M \in \mathbb{F}_q^m$ に対しても，$B_2 = G^{-1}(B_1)$ の計算ができる，すなわち，$G(\mathbf{x})$（あるいは $F(\mathbf{x})$）が全射である必要がある。

双極型システムでは，中心写像のクラス $\mathcal{C}_{\rm cent}$ の取り方を変えることで幅広い方式の構成が可能である。例えば，$\mathcal{C}_{\rm cent} = \{$ 三角型多変数多項式系 $\}$ とすると暗号化方式が得られる。双極型システムにおいては，$\mathcal{C}_{\rm cent}$ に関する EIP 問題がその安全性に大きく関わってくる。実際，EIP 問題を解けた場合，$F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ の代わりに分解 $F(\mathbf{x}) = T' \circ G'(\mathbf{x}) \circ S'$ を用いても，暗号化方式における復号および，署名方式における署名生成（偽造）が実行可能となる。EIP 問題の困難性はクラス $\mathcal{C}$ の選び方に依存するので，$\mathcal{C}$ の選び方に応じて個々に解析される必要がある。例えば，$\mathcal{C}_{\rm cent} = \{$ 三角型多変数多項式系 $\}$ としたときの EIP 問題は効率的に解けることが知られている [15]。

5.2.2 署名方式 UOV

5.2.2.1 UOV の概要

UOV [18, 7] は，双極型システムを用いた署名方式である。UOV の中心写像は，決まったいくつかの変数に値を代入することで 1 次式に変形でき，連立線形方程式の求解手法を用いて，効率的に署名生成が可能である。v, m を正の整数とし，$n = v + m$ とする。2 次多項式からなる多変数多項式系 $G(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_m(\mathbf{x}))$ を次の形で与える。

$$g_k(\mathbf{x}) = \sum_{\substack{1 \le i \le v \\ v+1 \le j \le n}} \alpha_{i,j}^{(k)} x_i x_j + \sum_{1 \le i \le j \le v} \beta_{i,j}^{(k)} x_i x_j + \sum_{1 \le i \le n} \gamma_i^{(k)} x_i + \eta^{(k)} \quad (k = 1, \ldots, m).$$

ここで，$\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in \mathbb{F}_q$ である。$G(\mathbf{x})$ の形で定義される多変数多項式写像を **UOV 多項式写像**と呼ぶ。$g_k(\mathbf{x})$ の 2 次多項式部分を $\tilde{g}_k(\mathbf{x})$ とすると，

$$\tilde{g}_k(\overbrace{0, \ldots, 0}^{v}, \overbrace{*, \ldots, *}^{m}) = 0 \tag{5.2}$$

となることが，UOV 多項式写像の特徴である。$x_1, \ldots, x_v$ をビィネガ変数，$x_{v+1}, \ldots, x_n$ をオイル変数と呼ぶ。$G(\mathbf{x})$ のヴィネガ変数に（ランダムな）値を代入すると，(5.2) によりオイル変数に関する 1 次式が得られる。$G(\mathbf{x})$ の逆写像は，連立線形方程式の求解手法を用いて効率的に計算できる。具体的に，任意の $\mathbf{c} = (c_1, \ldots, c_m) \in \mathbb{F}_q^m$ に対し，$\mathbf{b} = G^{-1}(\mathbf{c})$（の一つ）が以下のように計算できる。

1. $b_1, \ldots, b_v \in \mathbf{F}_q$ をランダムにとる。
2. $g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})$ に $(x_1, \ldots, x_v) = (b_1, \ldots, b_v)$ を代入して得られる $x_{v+1}, \ldots, x_n$ に関する 1 次式をそれぞれ

$\bar{g}_1(x_{v+1},\ldots,x_n),\ldots,\bar{g}_m(x_{v+1},\ldots,x_n)$ とする。連立線形方程式

$$\begin{cases} \bar{g}_1(x_{v+1},\ldots,x_n)=c_1 \\ \qquad\qquad\vdots \\ \bar{g}_m(x_{v+1},\ldots,x_n)=c_m \end{cases}$$

の解を計算し，それを $b_{v+1},\ldots,b_n$ と置く。もし解がなければ Step 1 に戻る。

3. $\mathbf{b}=(b_1,\ldots,b_n)$.

$\alpha_{i,j}^{(k)},\beta_{i,j}^{(k)},\gamma_i^{(k)},\eta^{(k)}\in\mathbb{F}_q$ を動かしてできる $G(\mathbf{x})$ の集合を $\mathcal{C}_{\mathrm{UOV}}$ としたとき，$\mathcal{C}_{\mathrm{cent}}=\mathcal{C}_{\mathrm{UOV}}$ として構成される双極型システムの署名方式を UOV と呼ぶ。但し，通常の双極型システムで使用するアフィン同型写像 T は，UOV の安全性には貢献しないので必要ない。すなわち，秘密鍵は $G(\mathbf{x})\in\mathcal{C}_{\mathrm{UOV}}$ と $\mathbb{F}_q^n$ 上のアフィン同型写像 S で，公開鍵は $F(\mathbf{x})=G(\mathbf{x})\circ S$ となる。$F(\mathbf{x})$ の成分の 2 次多項式部分を $\tilde{f}_1(\mathbf{x}),\ldots,\tilde{f}_m(\mathbf{x})$ とし，部分空間 $O\,(\subset\mathbb{F}_q^n)$ を

$$O=S^{-1}(\{(\overbrace{0,\ldots,0}^{v},\mathbf{a})\in\mathbb{F}_q^n\mid\mathbf{a}\in\mathbb{F}_q^m\})$$

とすると，(5.2) により $\tilde{f}_i(\mathbf{o})=0\ \ (i=1,\ldots,m,\ \mathbf{o}\in O)$ を満たす。このような性質を持つ部分空間をオイル空間という。逆に，多変数 2 次多項式系 $H(\mathbf{x})$ が o 次元のオイル空間 $O\,(\subset\mathbb{F}_q^n)$ を持ち，$o\geq m$ を満たすならば，$H(\mathbf{x})$ は UOV の公開鍵として使用できる。

5.2.2.2 UOV の公開鍵長の削減

双極型システムの公開鍵 $F(\mathbf{x})$ は，通常，その係数集合の形で記述され，その中でも，2 次多項式部分の係数集合が公開鍵の大部分を占める。$P(\mathbf{x})$ を多変数 2 次多項式系とし，$\tilde{p}_1(\mathbf{x}),\ldots,\tilde{p}_m(\mathbf{x})$ をその 2 次多項式部分とすると，ある行列 $\mathrm{P}_1,\ldots,\mathrm{P}_m\in\mathbb{F}_q^{n\times n}$ により，

$$\tilde{p}_i(\mathbf{x})=\mathbf{x}\,\mathrm{P}_i\,\mathbf{x}^\top\ \ (i=1,\ldots,m)$$

と表すことができる。一般に，行列 $\mathrm{P}=(p_{ij})\in\mathbb{F}_q^{h\times h}$ に対し，上三角行列 $\mathrm{upper(P)}=(c_{ij})\in\mathbb{F}_q^{h\times h}$ を

$$c_{ij}=\begin{cases} p_{ii} & i=j, \\ p_{ij}+p_{ji} & i<j, \\ 0 & i>j \end{cases}$$

で定義すると，$\tilde{p}_i(\mathbf{x})=\mathbf{x}\,\mathrm{P}_i\,\mathbf{x}^\top=\mathbf{x}\,\mathrm{upper}(\mathrm{P}_i)\,\mathbf{x}^\top$ が成り立つ。特に，P_i はすべて上三角行列で選ぶことができる。

UOV の中心写像 $G(\mathbf{x})$ の 2 次多項式部分に対応する上三角行列を $\mathrm{G}_1,\ldots,\mathrm{G}_m$ とし，公開鍵 $F(\mathbf{x})$ の 2 次多項式部分に対応する上三角行列を $\mathrm{F}_1,\ldots,\mathrm{F}_m$ とすると，

$$\mathrm{G}_i=\begin{pmatrix}\mathrm{G}_{i,1} & \mathrm{G}_{i,2}\\ \mathbf{0}_{o\times v} & \mathbf{0}_{o\times o}\end{pmatrix},\quad \mathrm{F}_i=\begin{pmatrix}\mathrm{F}_{i,1} & \mathrm{F}_{i,2}\\ \mathbf{0}_{o\times v} & \mathrm{F}_{i,3}\end{pmatrix}\qquad(\mathrm{G}_{i,1},\mathrm{F}_{i,1}\in\mathbb{F}_q^{v\times v},\ \mathrm{G}_{i,2},\mathrm{F}_{i,2}\in\mathbb{F}_q^{v\times o},\ \mathrm{F}_{i,3}\in\mathbb{F}_q^{o\times o})$$

の形で表すことができる。ここで，$\mathrm{G}_{i,1},\mathrm{G}_{i,3},\mathrm{F}_{i,1},\mathrm{F}_{i,3}$ は上三角行列である。今，アフィン同型写像 S の線形部分を表す行列 S （線形写像は $\mathbf{x}\mapsto\mathbf{x}\,\mathrm{S}$ の形）が

$$\mathrm{S}=\begin{pmatrix}\mathbf{I}_v & \mathbf{0}_{v\times o}\\ \mathrm{S}_0 & \mathbf{I}_o\end{pmatrix}\qquad(\mathrm{S}_0\in\mathbb{F}_q^{o\times v})$$

の形で表される場合に限定する。（但し，$\mathbf{I}_\ell$ は ℓ 次単位行列を表す。）すると，$\mathrm{F}_i = \mathrm{upper}(\mathrm{S}\,\mathrm{G}_i\,\mathrm{S}^\top)$ となるので，次の関係が成り立つ。

$$\begin{aligned}
&\cdot\ \mathrm{G}_{i,1} = \mathrm{F}_{i,1},\ \ \mathrm{G}_{i,2} = \mathrm{F}_{i,2} - (\mathrm{F}_{i,1} + \mathrm{F}_{i,1}^\top)\,\mathrm{S}_0^\top,\\
&\cdot\ \mathrm{F}_{i,3} = \mathrm{upper}(-\mathrm{S}_0\,\mathrm{F}_{i,1}^\top\,\mathrm{S}_0^\top + \mathrm{S}_0\,\mathrm{F}_{i,2}) = \mathrm{upper}(-\mathrm{S}_0\,\mathrm{F}_{i,1}\,\mathrm{S}_0^\top + \mathrm{S}_0\,\mathrm{F}_{i,2}).
\end{aligned} \tag{5.3}$$

これより，$\mathrm{F}_{i,1}$ は任意の上三角行列, $\mathrm{F}_{i,2}$ は任意の行列で選べることが分かる。そこで，$\mathrm{F}_{i,1}, \mathrm{F}_{i,2}$ の成分すべてを公開鍵として記述する代わりに，$\mathrm{F}_{i,1}, \mathrm{F}_{i,2}$ を疑似乱数生成器を用いて構成することにして，そのシードのみを公開鍵として記述することにより公開鍵長を削減できる。このようにして，UOV の公開鍵の 2 次多項式部分は，シードと (5.3) で求められた $\mathrm{F}_{i,3}$ $(i = 1, \ldots, m)$ だけで記述できる。

一般に，双極型システムの公開鍵長は n, m に関して，$\mathcal{O}(mn^2)$ の増大度を持ち，大きくなりやすい。UOV では，上の公開鍵の記述方法を使うことにより，公開鍵長の増大度は $\mathcal{O}(m^3)$ となり，UOV のパラメータが $2m < n$ で選ばれることを踏まえると，一般の双極型システムの公開鍵の記述方法よりも，公開鍵長を削減できる。この削減方法は，5.3 節で記述する（UOV の変種である）QR-UOV や MAYO にも利用されている。

5.2.3 MPC-in-the-Head

5.2.3.1 秘匿マルチパーティ計算

Ishai らによって導入された MPC-in-the-Head [17] は，秘匿マルチパーティ計算からゼロ知識証明を構成し，さらに Fiat-Shamir 変換により署名方式が構成できる。本来，MPC-in-the-Head の枠組みは広く，多変数公開鍵暗号に限定された技術ではないが，多変数公開鍵暗号においては，MQ 問題に付随する MPC-in-the-Head と，MinRank 問題に付随する MPC-in-the-Head が重要であるため，この 2 つの場合に限定して説明する。

$\mathcal{R} \subset \{0,1\}^* \times \{0,1\}^*$ を関係とする。命題 a に対して，$(a, x) \in \mathcal{R}$ であるとき，x は a の証拠（witness）であるという。ここでは，$\mathcal{R}$ として，MQ 問題（あるいは，MinRank 問題）のインスタンスを命題とし，その解を証拠とする関係に限定する。N 組のパーティ $\mathcal{P}_1, \ldots, \mathcal{P}_N$ が存在するとする。加法構造を持つ代数系の元 b に対し，分散 $[\![b]\!]$ は

$$[\![b]\!] = ([\![b]\!]_1, \ldots, [\![b]\!]_N) \text{ であり，} [\![b]\!]_1 + \cdots + [\![b]\!]_N = b$$

なるものを意味するとする。命題 a に対し，以下のような性質を持つ秘匿マルチパーティ計算 f を考える。

- 秘密情報 x の分散 $[\![x]\!]$ に対し，$\mathcal{P}_j$ は $[\![x]\!]_j$ を入力として受け取る。
- f は ‘受理’ か ‘棄却’ を返す。$(a, x) \in \mathcal{R}$ ならば，‘受理’ が返される。
- $N - 1$ 組以下のパーティのビューが集まっても x の情報は全く漏れない。

命題が以下のような MQ 問題のインスタンスである場合を考える。

$$\begin{cases}
\mathbf{x}\,\mathrm{A}_1\,\mathbf{x}^\top + \mathbf{x}\,\mathbf{b}_1^\top = y_1\\
\mathbf{x}\,\mathrm{A}_2\,\mathbf{x}^\top + \mathbf{x}\,\mathbf{b}_2^\top = y_2\\
\qquad\qquad\vdots\\
\mathbf{x}\,\mathrm{A}_m\,\mathbf{x}^\top + \mathbf{x}\,\mathbf{b}_m^\top = y_m
\end{cases} \tag{5.4}$$

ここで，$\mathrm{A}_1, \ldots, \mathrm{A}_m \in \mathbb{F}_q^{n\times n}$, $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{F}_q^n$ である。この場合の（MQOM [11] で利用されている）秘匿マルチパーティ計算 f_{MQ} の入力は，この MQ 問題の解 $\mathbf{x}^*$ の分散 $[\![\mathbf{x}^*]\!]$ である。η を正の整数とする。正の整数 n_1, n_2 を $n_1 n_2 \geq n$ なるように選んでおく。また，$u_1, \ldots, u_{n_2} \in \mathbb{F}_q$ を相異なる元として固定しておく。このとき，f_{MQ} の計算手順は以下の通りである。

秘匿マルチパーティ計算 f_{MQ}

1. ランダムオラクル $\mathcal{O}_R$ により $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{q^\eta}$ が作られ，全パーティに送信する。
2. 各パーティ $\mathcal{P}_j$ は $[\![z]\!]_j = \sum_{i=1}^m \gamma_i \, ([\![y_i]\!]_j - [\![\mathbf{x}^*]\!]_j \, \mathbf{b}_i^\top)$ を計算する。ここで，$[\![y_i]\!] = (y_i, 0, 0, \ldots, 0)$ である。
3. 各パーティ $\mathcal{P}_j$ は $[\![\mathbf{w}]\!]_j = [\![\mathbf{x}^*]\!]_j \, (\sum_{i=1}^m \gamma_i \, \mathrm{A}_i^\top)$ を計算する。
4. ヒントオラクル $\mathcal{O}_H$ により，$a_1, \ldots, a_{n_2} \in \mathbb{F}_{q^\eta}$，$Q'(u) \in \mathbb{F}_{q^\eta}[u]$ の分散 $[\![a_1]\!], \ldots, [\![a_{n_2}]\!]$，$[\![Q'(u)]\!]$ が作られ，対応するパーティに配布される。ここで，$a_1, \ldots, a_{n_2}$ はランダムに選ばれ，$Q'(u)$ は次のように定められる。まず，$n_1 - 1$ 次以下の多項式 $X_\ell(u) \in \mathbb{F}_q[u]$，$W_\ell(u) \in \mathbb{F}_{q^\eta}[u]$ $(\ell = 1, \ldots, n_2)$ を

$$\begin{cases} X_\ell(u_1) = x^*_{(\ell-1)n_1+1} \\ \quad\vdots \\ X_\ell(u_{n_1}) = x^*_{(\ell-1)n_1+n_1} \end{cases} \qquad \begin{cases} W_\ell(u_1) = w_{(\ell-1)n_1+1} \\ \quad\vdots \\ W_\ell(u_{n_1}) = w_{(\ell-1)n_1+n_1} \end{cases}$$

を満たす補間多項式によって計算する。次に，$\tilde{W}_\ell(u) \in \mathbb{F}_{q^\eta}[u]$ $(\ell = 1, \ldots, n_2)$ を

$$\tilde{W}_\ell(u) = W_\ell(u) + a_j \, (u - u_1)(u - u_2) \cdots (u - u_{n_1})$$

とし，$Q(u) \in \mathbb{F}_{q^\eta}[u]$ を $Q(u) = \sum_{\ell=1}^{n_2} X_\ell(u) \, \tilde{W}_\ell(u)$ で定める。最後に，q_0 を $Q(u)$ の定数項とし，$Q(u) = u \cdot Q'(u) + q_0$ で $Q'(u)$ を定める。
5. 各パーティ $\mathcal{P}_j$ は $n_1 - 1$ 次以下の多項式 $[\![X_\ell]\!]_j(u) \in \mathbb{F}_q[u]$，$[\![W_\ell]\!]_j(u) \in \mathbb{F}_{q^\eta}[u]$ $(\ell = 1, \ldots, n_2)$ を

$$\begin{cases} [\![X_\ell]\!]_j(u_1) = [\![x^*_{(\ell-1)n_1+1}]\!]_j \\ \quad\vdots \\ [\![X_\ell]\!]_j(u_{n_1}) = [\![x^*_{(\ell-1)n_1+n_1}]\!]_j \end{cases} \qquad \begin{cases} [\![W_\ell]\!]_j(u_1) = [\![w_{(\ell-1)n_1+1}]\!]_j \\ \quad\vdots \\ [\![W_\ell]\!]_j(u_{n_1}) = [\![w_{(\ell-1)n_1+n_1}]\!]_j \end{cases}$$

を満たす補間多項式によって計算する。（$[\![X_\ell]\!](u)$, $[\![W_\ell]\!](u)$ は，それぞれ $X_\ell(u)$, $W_\ell(u)$ の分散となる。）
6. 各パーティ $\mathcal{P}_j$ は $[\![\tilde{W}_\ell]\!]_j(u) \in \mathbb{F}_{q^\eta}[u]$ $(\ell = 1, \ldots, n_2)$ を

$$[\![\tilde{W}_\ell]\!]_j(u) = [\![W_\ell]\!]_j(u) + [\![a_\ell]\!]_j \, (u - u_1)(u - u_2) \cdots (u - u_{n_1})$$

で定める。（$[\![\tilde{W}_\ell]\!](u)$ は，$\tilde{W}_\ell(u)$ の分散となる。）
7. 各パーティ $\mathcal{P}_j$ は $[\![q_0]\!]_j = n_1^{-1} \cdot ([\![z]\!]_j - \sum_{i=1}^{n_1} u_i \, [\![Q']\!]_j(u_i))$ を計算する。
8. ランダムオラクル $\mathcal{O}_R$ により $r \in \mathbb{F}_{q^\eta} \backslash \{u_1, \ldots, u_{n_1}\}$ を取り，全パーティに送信する。
9. 各パーティ $\mathcal{P}_j$ は $[\![c_\ell]\!]_j = [\![\tilde{W}_\ell]\!]_j(r)$ $(\ell = 1, \ldots, n_2)$ を計算する。
10. 全パーティは $[\![c_\ell]\!]$ $(\ell = 1, \ldots, n_2)$ を共有し，$c_\ell \in \mathbb{F}_{q^\eta}$ $(\ell = 1, \ldots, n_2)$ を計算する。
11. 各パーティ $\mathcal{P}_j$ は $[\![v]\!]_j = r \cdot [\![Q']\!]_j(r) + [\![q_0]\!]_j - \sum_{\ell=1}^{n_2} c_\ell \, [\![X_\ell]\!]_j(r)$ を計算する。
12. 全パーティは $[\![v]\!]$ を共有し，v を計算する。
13. $v = 0$ なら ‘受理’，それ以外は ‘棄却’ を出力する。

この計算について補足する。この計算では，$\mathbf{x}^*$ が (5.4) の解であることを確認する代わりに，

$$\sum_{i=1}^m \gamma_i \, (y_i - \mathbf{x}^* \, \mathrm{A}_i \, {\mathbf{x}^*}^\top - \mathbf{x}^* \, \mathbf{b}_i^\top) = 0$$

であることを確認している。$\mathbf{x}^*$ が (5.4) の解でなくても，この等式は $1/q^\eta$ の確率で成り立つ。等式を書き直すと，

$$\sum_{i=1}^m \gamma_i \, (y_i - \mathbf{x}^* \, \mathbf{b}_i^\top) = \sum_{i=1}^m \gamma_i \, (\mathbf{x}^* \, \mathrm{A}_i \, {\mathbf{x}^*}^\top) = \mathbf{x}^* \left(\sum_{i=1}^m \gamma_i \, \mathrm{A}_i \right) {\mathbf{x}^*}^\top = \langle \mathbf{x}^*, \, \mathbf{w} \rangle, \qquad (\mathbf{w} = \mathbf{x}^* \left(\sum_{i=1}^m \gamma_i \, \mathrm{A}_i^\top \right))$$

となる。よって，$z = \sum_{i=1}^{m} \gamma_i \left(y_i - \mathbf{x}^* \, \mathbf{b}_i^\top\right)$ とおくならば，$z = \langle \mathbf{x}^*, \mathbf{w} \rangle$ を確かめればよい。これは，

$$z = \sum_{i=1}^{n_1} \sum_{\ell=1}^{n_2} X_\ell(u_i) \tilde{W}_\ell(u_i) = \sum_{i=1}^{n_1} Q(u_i) \tag{5.5}$$

と同値である。$v = 0$ であれば，Schwartz-Zippel の補題により，高い確率で (5.5) が満たされることになる。η を大きくすることで，この確率を 1 に近づけることができる。

5.2.3.2 ゼロ知識証明への変換

MPC-in-the-Head では秘匿マルチパーティ計算からゼロ知識証明を構成する。このゼロ知識証明は，命題 a に対して証拠 x を知っているかどうかを検証するものである。秘匿マルチパーティ計算 f_{MQ} に対応するゼロ知識証明の基本設計は以下の通りである。

f_{MQ} に対応するゼロ知識証明

1. 証明者は，証拠 $\mathbf{x}^*$ の分散 $[\![\mathbf{x}^*]\!]$ を作成する。そして，すべての $j \in \{1, \ldots, N\}$ に対し，$[\![\mathbf{x}^*]\!]_j$ のコミットメントを作成し，検証者に送る。
2. 検証者は，ランダムに $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{q^\eta}$ を生成し，これらをチャレンジとして証明者に送る。
3. 証明者は，$a_1, \ldots, a_{n_2}, Q'(u)$ を生成し，これらの分散 $[\![a_1]\!], \ldots, [\![a_{n_2}]\!], [\![Q']\!](u)$ を作成する。そして，すべての $j \in \{1, \ldots, N\}$ に対し，$[\![a_1]\!]_j, \ldots, [\![a_{n_2}]\!]_j, [\![Q']\!]_j(u)$ のコミットメントを作成し，検証者に送る。
4. 検証者は，ランダムに $r \in \mathbb{F}_{q^\eta} \backslash \{u_1, \ldots, u_{n_1}\}$ を生成し，これらをチャレンジとして証明者に送る。
5. 証明者は，（全てのパーティの計算を “頭の中で” 行い，）$[\![c_1]\!], \ldots, [\![c_{n_2}]\!]$，および，$[\![v]\!]$ を作成する。そして，これらを検証者に送る。
6. 検証者は，ランダムに $i^* \in \{1, 2, \ldots, N\}$ を生成し，これをチャレンジとして証明者に送る。
7. 証明者は，すべての $j \in \{1, 2, \ldots, N\} \backslash \{i^*\}$ に対し，$[\![\mathbf{x}]\!]_j, [\![a_1]\!]_j, \ldots, [\![a_{n_2}]\!]_j, [\![Q']\!]_j(u)$ を検証者に開示する。
8. 検証者は，以下を検証し，全て正しければ ‘受理’ を，それ以外は ‘棄却’ を出力する。
   - ✓ すべての $j \in \{1, 2, \ldots, N\} \backslash \{i^*\}$ に対し，$[\![\mathbf{x}^*]\!]_j$ のコミットメント，および，$[\![a_1]\!]_j, \ldots, [\![a_{n_2}]\!]_j, [\![Q']\!]_j(u)$ のコミットメントが正しいこと
   - ✓ すべての $j \in \{1, 2, \ldots, N\} \backslash \{i^*\}$ に対し，$[\![\mathbf{x}^*]\!]_j, [\![a_1]\!]_j, \ldots, [\![a_{n_2}]\!]_j, [\![Q']\!]_j(u)$ から $\mathcal{P}_j$ と同じ計算を行ったとき，$[\![c_1]\!]_j, \ldots, [\![c_{n_2}]\!]_j, [\![v]\!]_j$ の計算結果が一致すること
   - ✓ $v = 0$ であること

このゼロ知識証明について補足する。f_{MQ} における全てのパーティのビューは（もともと開示されるものを除き）コミットメントが作成され，検証者に送られている。また，f_{MQ} においてランダムオラクルが介入する部分は，検証者のチャレンジに置き換えられている。そして，1 つのパーティ以外のすべてのパーティに対するビューが開示され，そのビューから各パーティと同じ計算を行って得られる結果と開示情報が一致すること，および，$v = 0$ であることにより検証者は ‘受理’ を行っている。このゼロ知識証明の健全性誤差（soundness error）ε は約 $1/N$ である。このゼロ知識証明を τ 回繰り返すことにより，全体の健全性誤差を ε^τ にすることができる。

このゼロ知識証明に Fiat-Shamir 変換を施すことで署名方式 MQOM [11] が構成できる。MQOM ついては，5.3.4 節で詳しく述べる。また，5.3.5 節で詳しく述べる MiRitH は MinRank 問題に付随する秘匿マルチパーティ計算から構成される署名方式である。

5.3 多変数多項式に基づく主要な暗号方式

多変数公開鍵暗号で標準化が有力視されるのは効率的な検証と短い署名長を持つ署名方式の UOV である。但し，UOV は公開鍵長が大きくなりやすいという性質を持つため，公開鍵長の削減手法を取り入れている UOV の変種である QR-UOV と MAYO も標準化の有力候補である。

また，MPC-in-the-Head では，MQ 問題に関するマルチパーティ計算に基づく署名方式 MQOM と，MinRank 問題に関するマルチパーティ計算に基づく署名方式 MiRitH が注目されている。

表 5.2: 多変数多項式に基づく暗号の分類

文献	暗号化	鍵交換	署名
UOV [18, 7]			○
QR-UOV [12, 13]			○
MAYO [5, 6]			○
MQOM [11]			○
MiRitH [2]			○

5.3.1 署名方式 UOV

5.3.1.1 UOV の概要

5.2.2.1 節で UOV の基本アルゴリズムは述べたため，この節でのアルゴリズムの記述は割愛する。NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出された UOV [7] のアルゴリズムには，さらに，5.2.2.2 節で述べた公開鍵長の削減手法が取り入れられている。

5.3.1.2 UOV のパラメータ選択

UOV の設計に必要なパラメータは，q, m, n である。NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出されたドキュメント [7] では，以下のように UOV のパラメータ見積もりが公開されている。

(q, m, n)	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ
$(256, 44, 112)$	レベル 1	43,576 Bytes	48 Bytes	128 Bytes
$(16, 64, 160)$	レベル 1	66,576 Bytes	48 Bytes	96 Bytes
$(256, 72, 184)$	レベル 3	189,232 Bytes	48 Bytes	200 Bytes
$(256, 96, 244)$	レベル 5	446,992 Bytes	48 Bytes	260 Bytes

5.3.2 署名方式 QR-UOV

5.3.2.1 QR-UOV の概要

QR-UOV [12, 13] は UOV の変種である。$\mathbb{F}_{q^\ell}$ 上の行列の集合 $\mathbb{F}_{q^\ell}^{n' \times n'}$ は，$\mathbb{F}_q$ 上の行列の集合 $\mathbb{F}_q^{n'\ell \times n'\ell}$ の部分集合と見なすことができる。この部分集合に属する行列は，$\mathbb{F}_q^{n'\ell \times n'\ell}$ の元として表示するよりも，$\mathbb{F}_{q^\ell}^{n' \times n'}$ の元として表示する方が，サイズを $1/\ell$ 倍に圧縮できる。QR-UOV は，この性質を利用して UOV の公開鍵長を削減している。

ℓ, V, M を正の整数とし，$v = \ell \cdot V$, $m = \ell \cdot M$, $n = v + m$ とする。次数 ℓ の既約多項式 $f \in \mathbb{F}_q[t]$ を取り，$\mathbb{F}_q$ の拡大体 $E_f = \mathbb{F}_q[t]/(f)$ に，$\mathbb{F}_q$-基底を $1, t, t^2, \ldots, t^{\ell-1}$ で入れ，$\mathbb{F}_q$ 上のベクトル空間として $\mathbb{F}_q^\ell$ と同一視する。任意の $g \in \mathbb{F}_q[t]$ に対し，写像 $E_f \ni x \mapsto xg \in E_f$ は $\mathbb{F}_q$ 上の線形写像となる。よって，この写像は $\mathbb{F}_q$ 上の $\ell \times \ell$ 行列として表すことができる。この行列を $\Phi_g^f \in \mathbb{F}_q^{\ell\times\ell}$ と表し，$\mathcal{A}_f = \{\Phi_g^f \,|\, g \in E_f\} (\subset \mathbb{F}_q^{\ell\times\ell})$ とおく。$\phi : E_f \to \mathbb{F}_q$ を非自明な $\mathbb{F}_q$-線形写像で固定し，$\mathrm{W} = (\phi(t^{i+j-2}))_{ij} \in \mathbb{F}_q^{\ell\times\ell}$ とすると，任意の $\mathrm{X} \in \mathcal{A}_f$ に対して，$\mathrm{WX} \in \mathbb{F}_q^{\ell\times\ell}$ は対称行列になることが知られている（[12, Theorem 1]）。正の整数 a, b に対し，$\mathcal{A}_f^{a,b}$ を以下のような形で表される $a\ell \times b\ell$ 行列の集合とする：

$$\begin{pmatrix} \mathrm{X}_{11} & \mathrm{X}_{12} & \cdots & \mathrm{X}_{1b} \\ \mathrm{X}_{21} & \mathrm{X}_{22} & \cdots & \mathrm{X}_{2b} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{X}_{a1} & \mathrm{X}_{a2} & \cdots & \mathrm{X}_{ab} \end{pmatrix} \quad (X_{11}, X_{12}, \ldots, \mathrm{X}_{ab} \in \mathcal{A}_f).$$

$\mathrm{W}^{(a)} \in \mathbb{F}_q^{a\ell\times a\ell}$ を W が主対角線に a 個並ぶ対角行列とし，$\mathrm{W}^{(a)}\mathcal{A}_f^{a,b} = \{\mathrm{W}^{(a)}\,\mathrm{X} \,|\, \mathrm{X} \in \mathcal{A}_f^{a,b}\}$ $(\subset \mathbb{F}_q^{a\ell\times b\ell})$ とする。$\mathbb{F}_q^{a\ell\times b\ell}$ に属する一般の行列を記述するには，$\mathbb{F}_q$ の元が $a\,b\,\ell^2$ 個必要であるが，それが $\mathrm{W}^{(a)}\mathcal{A}_f^{a,b}$ に属する場合は，$a\,b\,\ell$ 個で記述できることに注意してほしい。

QR-UOV では，公開鍵 $F(\mathbf{x})$ の成分として 2 次斉次多項式を用いる。QR-UOV は双極型システムであるため，5.2.2.2 節で述べたように，$F(\mathbf{x})$ は，m 個の行列 $(\in \mathbb{F}_q^{n\times n})$ を用いて記述することができる。QR-UOV では，これらの行列がすべて $\mathrm{W}^{(V+M)}\mathcal{A}_f^{V+M,V+M}$ に属するような $F(\mathbf{x})$ だけを用いる。但し，これらの行列は上三角行列の形にはできないので，対称行列で記述する。この影響で，有限体の位数 q は奇数にする必要がある。$\mathrm{W}^{(V+M)}\mathcal{A}_f^{V+M,V+M}$ に属する行列は，一般の $\mathbb{F}_q^{n\times n}$ に属する行列よりも小さいサイズで記述できるため，オリジナルの UOV よりも QR-UOV の公開鍵の方が小さいサイズで記述できる。また，5.2.2.2 節で述べた UOV に対して適用できる公開鍵長削減手法は，QR-UOV に対しても適用可能である。

安全性パラメータを λ とし，以下の関数を用意する。

- $\mathrm{Expand}_{\mathrm{sk}}$： 任意の 2$\lambda$-ビット列から 1 個の $\mathcal{A}_f^{V,M}$ に属する行列を生成する疑似乱数生成関数
- $\mathrm{Expand}_{\mathrm{pk}}$： 任意の 2$\lambda$-ビット列から m 個の $\mathrm{W}^{(V)}\mathcal{A}_f^{V,V}$ に属する対称行列と，m 個の $\mathrm{W}^{(V)}\mathcal{A}_f^{V,M}$ に属する行列を生成する疑似乱数生成関数
- $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$： 暗号学的ハッシュ関数

鍵生成

1. $\mathrm{seed}_{\mathrm{pk}}$, $\mathrm{seed}_{\mathrm{sk}} \in \{0,1\}^{2\lambda}$ をランダムに選ぶ。
2. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathrm{W}^{(V)}\mathcal{A}_f^{V,V}$ (対称行列)，$\mathrm{P}_{i,2} \in \mathrm{W}^{(V)}\mathcal{A}_f^{V,M}$ $(i = 1, \ldots, m)$ を得る.
3. $\mathrm{Expand}_{\mathrm{sk}}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\mathrm{S}_0 \in \mathcal{A}_f^{V,M}$ を得る。
4. $\mathrm{P}_{i,3} = -\mathrm{S}_0^\top \mathrm{P}_{i,1}\mathrm{S}_0 + \mathrm{P}_{i,2}^\top \mathrm{S}_0 + \mathrm{S}_0^\top \mathrm{P}_{i,2} \in \mathbb{F}_q^{m\times m}$ $(i = 1, \ldots, m)$ を計算する。

公開鍵は $\mathrm{pk} = (\mathrm{seed}_{\mathrm{pk}}, \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$，秘密鍵は $\mathrm{sk} = \mathrm{seed}_{\mathrm{sk}}$ である。次に，署名生成である。メッセージを $M \in \{0,1\}^*$ とする。

署名生成

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$ を取り出す.
2. sk から $\mathrm{seed}_{\mathrm{sk}}$ を取り出す.
3. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathrm{W}^{(V)}\mathcal{A}_f^{V,V}$ (対称行列)，$\mathrm{P}_{i,2} \in W^{(V)}\mathcal{A}_f^{V,M}$ $(i = 1, \ldots, m)$ を得る.

4. $\mathrm{Expand}_{\mathrm{sk}}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\mathrm{S}_0 \in \mathcal{A}_f^{V,M}$ を得る。
5. $\mathrm{G}_i = -\mathrm{P}_{i,1}\mathrm{S}_0 + \mathrm{P}_{i,2} \in \mathbb{F}_q^{v\times m}$ $(i = 1, \ldots, m)$ を計算する。
6. $\mathrm{U} = \begin{pmatrix} \mathbf{I}_v & \mathbf{0}_{v\times m} \\ -\mathrm{S}_0^\top & \mathbf{I}_m \end{pmatrix} \in \mathbb{F}_q^{n\times n}$ とおく。
7. $\mathbf{y} = (y_1, \ldots, y_v) \in \mathbb{F}_q^v$ をランダムに選ぶ。
8. $\mathrm{L} = (2\,(\mathbf{y}\,\mathrm{G}_1)^\top, \ldots, 2\,(\mathbf{y}\,\mathrm{G}_m)^\top) \in \mathbb{F}_q^{m\times m}$ を計算する。（縦ベクトルを m 列並べて行列を作る。）
9. $\mathbf{u} = (\mathbf{y}\,\mathrm{P}_{1,1}\,\mathbf{y}^\top, \ldots, \mathbf{y}\,\mathrm{P}_{m,1}\,\mathbf{y}^\top) \in \mathbb{F}_q^m$ を計算する。
10. $\mathrm{salt} \in \{0,1\}^\lambda$ をランダムに選び，$\mathbf{t} = \mathcal{H}(M \,\|\, \mathrm{salt})$ を計算する。
11. 連立線形方程式 $\mathbf{x}\,\mathrm{L} = \mathbf{t} - \mathbf{u}$ の解を計算し，解 $\mathbf{x} = (y_{v+1}, \ldots, y_n) \in \mathbb{F}_q^m$ を得る。もし解がなければ，Step 10 に戻る。
12. $\mathbf{s} = (y_1, \ldots, y_n)\,\mathrm{U}$ を計算する.

$\sigma = (\mathrm{salt}, \mathbf{s})$ が署名となる。最後に検証である。

検証

1. pk から $(\mathrm{seed}_{\mathrm{pk}},\ \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$ を取り出す.
2. σ から $(\mathrm{salt}, \mathbf{s})$ を取り出す.
3. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathrm{W}^{(V)}\mathcal{A}_f^{V,V}$（対称行列），$\mathrm{P}_{i,2} \in \mathrm{W}^{(V)}\mathcal{A}_f^{V,M}$ $(i = 1, \ldots, m)$ を得る.
4. $\mathrm{F}_i = \begin{pmatrix} \mathrm{P}_{i,1} & \mathrm{P}_{i,2} \\ \mathrm{P}_{i,2}^\top & \mathrm{P}_{i,3} \end{pmatrix}$ $(i = 1, \ldots, m)$ とおく。
5. $\mathbf{t} = \mathcal{H}(M \,\|\, \mathrm{salt})$ を計算する.
6. $\mathbf{t}' = (\mathbf{s}\,\mathrm{F}_1\,\mathbf{s}^\top, \ldots, \mathbf{s}\,\mathrm{F}_m\,\mathbf{s}^\top)$ を計算する.
7. $\mathbf{t} = \mathbf{t}'$ ならば ‘受理’ を，それ以外は ‘棄却’ を返す。

5.3.2.2 QR-UOV のパラメータ選択

QR-UOV の設計に必要なパラメータは，λ, q, v, m, ℓ である。NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出されたドキュメント [13] では，以下のように QR-UOV のパラメータ見積もりが公開されている。

(λ, q, v, m, ℓ)	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ
$(128, 7, 740, 100, 10)$	レベル 1	20,657 Bytes	32 Bytes	331 Bytes
$(128, 31, 165, 60, 3)$	レベル 1	23,657 Bytes	32 Bytes	157 Bytes
$(128, 31, 600, 70, 10)$	レベル 1	12,282 Bytes	32 Bytes	435 Bytes
$(128, 127, 156, 54, 3)$	レベル 1	24,271 Bytes	32 Bytes	200 Bytes
$(192, 7, 1100, 140, 10)$	レベル 3	55,173 Bytes	48 Bytes	489 Bytes
$(192, 31, 246, 87, 3)$	レベル 3	71,007 Bytes	48 Bytes	232 Bytes
$(192, 31, 890, 100, 10)$	レベル 3	34,423 Bytes	48 Bytes	643 Bytes
$(192, 127, 228, 78, 3)$	レベル 3	71,915 Bytes	48 Bytes	292 Bytes
$(256, 7, 1490, 190, 10)$	レベル 5	135,439 Bytes	64 Bytes	662 Bytes
$(256, 31, 324, 114, 3)$	レベル 5	158,453 Bytes	64 Bytes	306 Bytes
$(256, 31, 1120, 120, 10)$	レベル 5	58,564 Bytes	64 Bytes	807 Bytes
$(256, 127, 306, 105, 3)$	レベル 5	173,708 Bytes	64 Bytes	392 Bytes

5.3.3 署名方式 MAYO

5.3.3.1 MAYO の概要

MAYO [5, 6] は UOV の変種である。公開鍵を作る基となる変数の個数が少ない多変数多項式系 $P(\mathbf{x})$ を用意しておき，検証者は，検証時（あるいはそれ以前）に $P(\mathbf{x})$ から MAYO の公開鍵 $F(\mathbf{x})$ を構成する。そのため，MAYO の公開鍵は，$F(\mathbf{x})$ の係数集合ではなく，$P(\mathbf{x})$ の係数集合となる。これにより，MAYO は，オリジナルの UOV に比べて公開鍵長を小さくすることができる。

m, v, o, k を正の整数とし，$o < m$, $n = v + o$ とする。2 次斉次多変数多項式写像 $P(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})) : \mathbb{F}_q^n \to \mathbb{F}_q^m$ に対し，ある o 次元部分空間 $O\,(\subset \mathbb{F}_q^n)$ があり，

$$P(\mathbf{o}) = \mathbf{0}_m \quad (\mathbf{o} \in O)$$

を満たすとする。5.2.2.1 節の言葉を使えば，O はオイル空間である。もし $o \geq m$ であれば，$P(\mathbf{x})$ は UOV の公開鍵として使用できるが，$o < m$ なのでそれはできない。$P^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k) : \mathbb{F}_q^{kn} \to \mathbb{F}_q^m$ を次のようにおく。

$$P^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \sum_{i=1}^{k} P(\mathbf{x}_i)\,\mathrm{E}_{i,i} + \sum_{1 \leq i < j \leq k} P'(\mathbf{x}_i, \mathbf{x}_j)\,\mathrm{E}_{i,j} \tag{5.6}$$

ここで，$\mathrm{E}_{i,j} \in \mathbb{F}_q^{m \times m}$ $(1 \leq i \leq j \leq k)$ は正則行列であり，$P'(\mathbf{x}, \mathbf{y})$ は $P(\mathbf{x} + \mathbf{y}) - P(\mathbf{x}) - P(\mathbf{y})$ で定まる双線形写像である。すると，

$$P^\star(\mathbf{o}_1, \ldots, \mathbf{o}_k) = \mathbf{0}_m \quad (\mathbf{o}_1, \ldots, \mathbf{o}_k \in O)$$

を満たすので，O^k が $P^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ のオイル空間となる。$ko = \dim_{\mathbb{F}_q} O^k$ なので，$ko \geq m$ を満たせば，$P^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ は UOV の公開鍵として使用できる。$\mathrm{E}_{i,j}$ $(1 \leq i \leq j \leq n)$ をシステムパラメータとしておけば，$P^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ は (5.6) により，$P(\mathbf{x})$ だけから構成できる。公開鍵を $P(\mathbf{x})$ の係数集合だけで記述することで，公開鍵のサイズを小さくした UOV が MAYO である。さらに，5.2.2.2 節で述べた UOV に対して適用できる公開鍵長削減手法は，MAYO に対しても適用可能である。

安全性パラメータを λ とし，以下の行列，関数を用意する。

- 正則行列 $\mathrm{E}_{i,j} \in \mathbb{F}_q^{m \times m}$ $(1 \leq i \leq j \leq k)$
- $\mathrm{Expand}_{\mathrm{sk}}$： 任意の λ-ビット列から 1 個の $\mathbb{F}_q^{o \times v}$ に属する行列を生成する疑似乱数生成関数
- $\mathrm{Expand}_{\mathrm{pk}}$： 任意の λ-ビット列から m 個の $\mathbb{F}_q^{v \times v}$ に属する上三角行列と，m 個の $\mathbb{F}_q^{v \times o}$ に属する行列を生成する疑似乱数生成関数
- $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$： 暗号学的ハッシュ関数

鍵生成

1. $\mathrm{seed}_{\mathrm{pk}}$, $\mathrm{seed}_{\mathrm{sk}} \in \{0,1\}^\lambda$ をランダムに選ぶ。
2. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathbb{F}_q^{v \times v}$（上三角行列），$\mathrm{P}_{i,2} \in \mathbb{F}_q^{v \times o}$ $(i = 1, \ldots, m)$ を得る.
3. $\mathrm{Expand}_{\mathrm{sk}}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\mathrm{R} \in \mathbb{F}_q^{o \times v}$ を得る。
4. $\mathrm{P}_{i,3} = \mathrm{upper}(-\mathrm{R}\,\mathrm{P}_{i,1}\mathrm{R}^\top - \mathrm{R}\,\mathrm{P}_{i,2}) \in \mathbb{F}_q^{o \times o}$ $(i = 1, \ldots, m)$ を計算する。

公開鍵は $\mathrm{pk} = (\mathrm{seed}_{\mathrm{pk}}, \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$，秘密鍵は $\mathrm{sk} = \mathrm{seed}_{\mathrm{sk}}$ である。次に，署名生成である。メッセージを $M \in \{0,1\}^*$ とする。

署名生成

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$ を取り出す.
2. sk から $\mathrm{seed}_{\mathrm{sk}}$ を取り出す.
3. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathbb{F}_q^{v\times v}$（上三角行列），$\mathrm{P}_{i,2} \in \mathbb{F}_q^{v\times o}$ $(i = 1, \ldots, m)$ を得る.
4. $\mathrm{Expand}_{\mathrm{sk}}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\mathrm{R} \in \mathbb{F}_q^{o\times v}$ を得る。
5. $\mathrm{F}_i = (\mathrm{P}_{i,1} + \mathrm{P}_{i,1}^\top)\,\mathrm{R}^\top + \mathrm{P}_{i,2} \in \mathbb{F}_q^{v\times o}$ $(i = 1, \ldots, m)$ を計算する。
6. $\mathrm{U} = \begin{pmatrix} \mathbf{I}_v & \mathbf{0}_{v\times o} \\ \mathrm{R} & \mathbf{I}_o \end{pmatrix} \in \mathbb{F}_q^{n\times n}$ とおく。
7. $\mathbf{y}_1, \ldots, \mathbf{y}_k \in \mathbb{F}_q^v$（横ベクトル）をランダムに選ぶ。
8. $\mathrm{L}_i = \sum_{j=1}^{i}((\mathbf{y}_j\,\mathrm{F}_1)^\top, \ldots, (\mathbf{y}_j\,\mathrm{F}_m)^\top)\,\mathrm{E}_{j,i} + \sum_{j=i+1}^{k}((\mathbf{y}_j\,\mathrm{F}_1)^\top, \ldots, (\mathbf{y}_j\,\mathrm{F}_m)^\top)\,\mathrm{E}_{i,j} \in \mathbb{F}_q^{o\times m}$ $(i = 1, \ldots, k)$ を計算する.
9. $\mathrm{L} = \begin{pmatrix} \mathrm{L}_1 \\ \vdots \\ \mathrm{L}_k \end{pmatrix} \in \mathbb{F}_q^{ko\times m}$ とおく。
10. $\mathbf{u} = \sum_{i=1}^{k}(\mathbf{y}_i\,\mathrm{P}_{1,1}\,\mathbf{y}_i^\top, \ldots, \mathbf{y}_i\,\mathrm{P}_{m,1}\,\mathbf{y}_i^\top)\,\mathrm{E}_{i,i} + \sum_{1\le i<j\le k}(\mathbf{y}_i\,(\mathrm{P}_{1,1}+\mathrm{P}_{1,1}^\top)\,\mathbf{y}_j^\top, \ldots, \mathbf{y}_i\,(\mathrm{P}_{m,1}+\mathrm{P}_{m,1}^\top)\,\mathbf{y}_j^\top)\,\mathrm{E}_{i,j} \in \mathbb{F}_q^m$ を計算する。
11. $\mathrm{salt} \in \{0,1\}^\lambda$ をランダムに選び，$\mathbf{t} = \mathcal{H}(M \,\|\, \mathrm{salt})$ を計算する。
12. 連立線形方程式 $(\mathbf{x}_1, \ldots, \mathbf{x}_k)\,\mathrm{L} = \mathbf{t} - \mathbf{u}$ の解を計算し，解 $(\mathbf{x}_1, \ldots, \mathbf{x}_k) = (\mathbf{z}_1, \ldots, \mathbf{z}_k) \in \mathbb{F}_q^{ko}$ を得る。もし解がなければ，Step 11 に戻る。
13. $\mathbf{s} = ((\mathbf{y}_1, \mathbf{z}_1)\,\mathrm{U}, \ldots, (\mathbf{y}_k, \mathbf{z}_k)\,\mathrm{U})$ を計算する.

$\sigma = (\mathrm{salt}, \mathbf{s})$ が署名となる。最後に検証である。

検証

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$ を取り出す.
2. σ から $(\mathrm{salt}, (\mathbf{s}_1, \ldots, \mathbf{s}_k))$ を取り出す.
3. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathbb{F}_q^{v\times v}$（上三角行列），$\mathrm{P}_{i,2} \in \mathbb{F}_q^{v\times o}$ $(i = 1, \ldots, m)$ を得る.
4. $\mathrm{P}_i = \begin{pmatrix} \mathrm{P}_{i,1} & \mathrm{P}_{i,2} \\ \mathbf{0}_{v\times o} & \mathrm{P}_{i,3} \end{pmatrix}$ $(i = 1, \ldots, m)$ とおく。
5. $\mathbf{t} = \mathcal{H}(M \,\|\, \mathrm{salt})$ を計算する.
6. $\mathbf{t}' = \sum_{i=1}^{k}(\mathbf{s}_i\mathrm{P}_1\,\mathbf{s}_i^\top, \ldots, \mathbf{s}_i\mathrm{P}_k\,\mathbf{s}_i^\top)\,\mathrm{E}_{i,i} + \sum_{1\le i<j\le k}(\mathbf{s}_i(\mathrm{P}_1+\mathrm{P}_1^\top)\,\mathbf{s}_j^\top, \ldots, \mathbf{s}_i(\mathrm{P}_k+\mathrm{P}_k^\top)\,\mathbf{s}_j^\top)\,\mathrm{E}_{i,j}$ を計算する.
7. $\mathbf{t} = \mathbf{t}'$ ならば ‘受理’ を，それ以外は ‘棄却’ を返す。

5.3.3.2 MAYO のパラメータ選択

MAYO のパラメータは，NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出されたドキュメント [6] に記載されていたが，(MAYO に限定されない) under-defined な MQ 問題に対する解読手法 [16] が適用できることが分かり，パラメータの修正が必要となった。しかし，これは致命的な攻撃ではなく，MAYO は NIST PQC 標準化プロジェクト追加署名第 2 ラウンドへの進出が決まっている。MAYO の Round 2 ドキュメントで修正パラメータが公開される予定であるが，本稿の執筆時点（2025 年 2 月 19 日時点）では公開されていない。

5.3.4 署名方式 MQOM

5.3.4.1 MQOM の概要

MQOM [11] は，5.2.3.1 節で説明した MQ 問題に関する秘匿マルチパーティ計算 f_{MQ} から MPC-in-the-Head で構成された署名方式である。5.2.3.2 節で述べたように，f_{MQ} はゼロ知識証明に変換することができる。さらに，Fiat-Shamir 変換により署名方式が構成できる。以下では，5.2.3.1 節の設定や記号を用いる。安全性パラメータを λ とし，以下の関数を用意する。

- Expand： 任意の λ-ビット列を入力とする疑似乱数生成関数
- $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3 : \{0,1\}^* \to \{0,1\}^{2\lambda}$ ：暗号学的ハッシュ関数
- Commit：コミットメント関数

鍵生成

1. $\mathrm{seed}_{\mathrm{pk}},\ \mathrm{seed}_{\mathrm{sk}} \in \{0,1\}^{\lambda}$ をランダムに選ぶ。
2. Expand($\mathrm{seed}_{\mathrm{pk}}$) の計算により，$\mathrm{A}_i \in \mathbb{F}_q^{n\times n}$（上三角行列），$\mathbf{b}_i \in \mathbb{F}_q^n \ \ (i = 1, \ldots, m)$ を得る.
3. Expand($\mathrm{seed}_{\mathrm{sk}}$) の計算により，$\mathbf{x}^* \in \mathbb{F}_q^n$ を得る。
4. $y_i = \mathbf{x}^* \, \mathrm{A}_i \, \mathbf{x}^{*\top} + \mathbf{x}^* \, \mathbf{b}_i^{\top} \quad (i = 1, \ldots, m)$ を計算する。

公開鍵は $\mathrm{pk} = (\mathrm{seed}_{\mathrm{pk}}, \mathbf{y} = (y_1, \ldots, y_m))$，秘密鍵は $\mathrm{sk} = \mathrm{seed}_{\mathrm{sk}}$ である。次に，署名生成である。メッセージを $M \in \{0,1\}^*$ とする。

署名生成

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \mathbf{y})$ を取り出す。
2. sk から $\mathrm{seed}_{\mathrm{sk}}$ を取り出す。
3. Expand($\mathrm{seed}_{\mathrm{pk}}$) の計算により，$\mathrm{A}_i \in \mathbb{F}_q^{n\times n}$（上三角行列），$\mathbf{b}_i \in \mathbb{F}_q^n \ \ (i = 1, \ldots, m)$ を得る.
4. $\mathrm{salt} \in \{0,1\}^{2\lambda},\ \mathrm{mseed} \in \{0,1\}^{\lambda}$ をランダムに選ぶ。
5. Expand(salt, mseed) の計算により，$\mathrm{rseed}^{[e]} \in \{0,1\}^{\lambda} \ \ (e = 1, \ldots, \tau)$ を得る.
6. $e = 1, \ldots, \tau$ に対して以下を行う：
   6-1. Expand(salt, $\mathrm{rseed}^{[e]}$) の計算により，$\mathrm{seed}_j^{[e]} \in \{0,1\}^{\lambda} \ \ (j = 1, \ldots, N)$ を得る.
   6-2. すべての $j = 1, \ldots, N$ に対し，Expand(salt, $\mathrm{seed}_j^{[e]}$) の計算により以下を得る：
   · $j < N$ ならば，$[\![\mathbf{x}^{*[e]}]\!]_j$, $[\![a_1^{[e]}]\!]_j, \ldots, [\![a_{n_2}^{[e]}]\!]_j$, $[\![Q'^{[e]}]\!]_j(u)$
   · $j = N$ ならば，$[\![a_1^{[e]}]\!]_N, \ldots, [\![a_{n_2}^{[e]}]\!]_N$
   6-3. $[\![\mathbf{x}^{*[e]}]\!]_N = \mathbf{x}^* - \sum_{j=1}^{N-1} [\![\mathbf{x}^{*[e]}]\!]_j$ を計算する。
   6-4. コミットメントを計算する：

$$\mathrm{com}_j^{[e]} = \begin{cases} \mathrm{Commit}(\mathrm{salt},\, e,\, j,\, \mathrm{seed}_j^{[e]}) & j = 1, \ldots, N-1 \\ \mathrm{Commit}(\mathrm{salt},\, e,\, N,\, \mathrm{seed}_N^{[e]},\, [\![\mathbf{x}^{*[e]}]\!]_N) & j = N \end{cases}$$

7. $h_1 = \mathcal{H}_1(M, \mathrm{salt}, \mathrm{com}_1^{[1]}, \ldots, \mathrm{com}_N^{[\tau]})$ を計算する。
8. Expand(h_1) の計算により，$\gamma_1^{[e]}, \ldots, \gamma_m^{[e]} \ \ (e = 1, \ldots, \tau)$ を得る。
9. $e = 1, \ldots, \tau$ に対して以下を行う：
   9-1. f_{MQ} のヒントオラクルと同じ計算により，$\mathbf{x}^*$, $\{\mathrm{A}_i, \mathbf{b}_i\}_{i=1,\ldots,m}$, $\gamma_1^{[e]}, \ldots, \gamma_m^{[e]}$, $a_1^{[e]}, \ldots, a_{n_2}^{[e]}$（但し，$a_i^{[e]} =$

$\sum_{j=1}^{N}[\![a_i^{[e]}]\!]_j$) から，$Q'^{[e]}(u) \in \mathbb{F}_{q^\eta}[u]$ を計算する。

9-2. $[\![Q'^{[e]}]\!]_N(u) = Q'^{[e]}(u) - \sum_{j=1}^{N-1}[\![Q'^{[e]}]\!]_j(u)$ を計算する。

9-3. コミットメント $\mathrm{com}'^{[e]}_N = \mathrm{Commit}(\mathrm{salt}, e, 0, [\![Q'^{[e]}]\!]_N(u))$ を計算する。

10. $h_2 = \mathcal{H}_2(M, \mathrm{salt}, h_1, \mathrm{com}'^{[1]}_N, \ldots, \mathrm{com}'^{[\tau]}_N)$ を計算する。

11. $\mathrm{Expand}(h_2)$ の計算により，$r^{[1]}, \ldots, r^{[\tau]} \in \mathbb{F}_{q^\eta}$ を得る。

12. $e = 1, \ldots, \tau$ に対して以下を行う：

12-1. f_{MQ} における全パーティと同じ計算により，$[\![\mathbf{x}^{*[e]}]\!], [\![a_1^{[e]}]\!], \ldots, [\![a_{n_2}^{[e]}]\!], [\![Q'^{[e]}]\!](u)$ から，$[\![\mathrm{broad}^{[e]}]\!] = ([\![c_1^{[e]}]\!], \ldots, [\![c_{n_2}^{[e]}]\!], [\![v^{[e]}]\!])$ を計算する。

12-2. $\mathrm{broad}^{[e]} = (c_1^{[e]}, \ldots, c_{n_2}^{[e]}, v^{[e]}) = \sum_{j=1}^{N}[\![\mathrm{broad}^{[e]}]\!]$ を計算する。

13. $h_3 = \mathcal{H}_3(M, \mathrm{salt}, h_2, [\![\mathrm{broad}^{[1]}]\!], \ldots, [\![\mathrm{broad}^{[\tau]}]\!])$ を計算する。

14. $\mathrm{Expand}(h_3)$ の計算により，$i^{*[1]}, \ldots, i^{*[\tau]} \in \{1, \ldots, N\}$ を得る。

15. $\mathrm{view}^{[e]}$ $(e = 1, \ldots, \tau)$ を以下のようにおく：

$$\mathrm{view}^{[e]} = \begin{cases} (\{\mathrm{seed}_j^{[e]}\}_{j\in\{1,\ldots,N\}\setminus\{i^{*[e]}\}}, [\![\mathbf{x}^{*[e]}]\!]_N, [\![Q'^{[e]}]\!]_N(u)) & i^{*[e]} \neq N \\ (\{\mathrm{seed}_j^{[e]}\}_{j\in\{1,\ldots,N\}\setminus\{i^{*[e]}\}}) & i^{*[e]} = N \end{cases}$$

16. $\sigma = (\mathrm{salt}, h_1, h_2, h_3, \{\mathrm{view}^{[e]}, \mathrm{broad}^{[e]}, \mathrm{com}^{[e]}_{i^{*[e]}}, \mathrm{com}'^{[e]}_N\}_{e=1,\ldots,\tau})$ とおく。

σ が署名となる。最後に検証である。

検証

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \mathbf{y})$ を取り出す。
2. σ から $(\mathrm{salt}, h_1, h_2, h_3, \{\mathrm{view}^{[e]}, \mathrm{broad}^{[e]}, \mathrm{com}^{[e]}_{i^{*[e]}}, \mathrm{com}'^{[e]}_N\}_{e=1,\ldots,\tau})$ を取り出す。
3. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{A}_i \in \mathbb{F}_q^{n\times n}$（上三角行列），$\mathbf{b}_i \in \mathbb{F}_q^n$ $(i = 1, \ldots, m)$ を得る.
4. $\mathrm{Expand}(h_1)$ の計算により，$\gamma_1^{[e]}, \ldots, \gamma_m^{[e]}$ $(e = 1, \ldots, \tau)$ を得る。
5. $\mathrm{Expand}(h_2)$ の計算により，$r^{[1]}, \ldots, r^{[\tau]} \in \mathbb{F}_{q^\eta}$ を得る。
6. $\mathrm{Expand}(h_3)$ の計算により，$i^{*[1]}, \ldots, i^{*[\tau]} \in \{1, \ldots, N\}$ を得る。
7. $e = 1, \ldots, \tau$ に対して以下を行う：

7-1. $\mathrm{broad}^{[e]}$ から $(c_1^{[e]}, \ldots, c_{n_2}^{[e]}, v^{[e]})$ を取り出す。

7-2. $\mathrm{view}^{[e]}$ から以下を取り出す：

· $i^{*[e]} \neq N$ ならば，$\{\mathrm{seed}_j^{[e]}\}_{j\in\{1,\ldots,N\}\setminus\{i^{*[e]}\}}, [\![\mathbf{x}^{*[e]}]\!]_N, [\![Q'^{[e]}]\!]_N(u)$

· $i^{*[e]} = N$ ならば，$\{\mathrm{seed}_j^{[e]}\}_{j\in\{1,\ldots,N\}\setminus\{i^{*[e]}\}}$

7-3. すべての $j \in \{1, \ldots, N\}\setminus\{i^{*[e]}\}$ に対し，$\mathrm{Expand}(\mathrm{salt}, \mathrm{seed}_j^{[e]})$ の計算により以下を得る：

· $j < N$ ならば，$[\![\mathbf{x}^{*[e]}]\!]_j, [\![a_1^{[e]}]\!]_j, \ldots, [\![a_{n_2}^{[e]}]\!]_j, [\![Q'^{[e]}]\!]_j(u)$

· $j = N$ ならば，$[\![a_1^{[e]}]\!]_N, \ldots, [\![a_{n_2}^{[e]}]\!]_N$

7-4. すべての $j \in \{1, \ldots, N\}\setminus\{i^{*[e]}\}$ に対し，f_{MQ} における $\mathcal{P}_j$ と同じ計算より，$[\![\mathbf{x}^{*[e]}]\!]_j, [\![a_1^{[e]}]\!]_j, \ldots, [\![a_{n_2}^{[e]}]\!]_j, [\![Q'^{[e]}]\!]_j(u), c_1^{[e]}, \ldots, c_{n_2}^{[e]}$ から，$[\![\mathrm{broad}^{[e]}]\!]_j = ([\![c_1^{[e]}]\!]_j, \ldots, [\![c_{n_2}^{[e]}]\!]_j, [\![v^{[e]}]\!]_j)$ を計算する。

7-5. $[\![\mathrm{broad}^{[e]}]\!]_{i^{*[e]}} = \mathrm{broad}^{[e]} - \sum_{j\in\{1,\ldots,N\}\setminus\{i^{*[e]}\}}[\![\mathrm{broad}^{[e]}]\!]_j$ を計算する。

7-6. すべての $j \in \{1, \ldots, N\}\setminus\{i^{*[e]}\}$ に対し，以下のようにコミットメントを計算する：

· $j < N$ ならば，$\mathrm{com}_j^{[e]} = \mathrm{Commit}(\mathrm{salt}, e, j, \mathrm{seed}_j^{[e]})$

· $j = N$ ならば，$\mathrm{com}_j^{[e]} = \mathrm{Commit}(\mathrm{salt}, e, N, \mathrm{seed}_N^{[e]}, [\![\mathbf{x}^{*[e]}]\!]_N)$

$\mathrm{com}''^{[e]}_j = \mathrm{Commit}(\mathrm{salt}, e, 0, [\![Q'^{[e]}]\!]_N(u))$

8. $h'_1 = \mathcal{H}_1(M, \text{salt}, \text{com}_1^{[1]}, \ldots, \text{com}_N^{[\tau]})$ を計算する。
9. $h'_2 = \mathcal{H}_2(M, \text{salt}, h'_1, \text{com}'^{[1]}_N, \ldots, \text{com}'^{[\tau]}_N)$ を計算する。
10. $h'_3 = \mathcal{H}_3(M, \text{salt}, h'_2, [\![\text{broad}^{[1]}]\!], \ldots, [\![\text{broad}^{[\tau]}]\!])$ を計算する。
11. $i^{*[e]} \neq N$ なる $e \in \{1, \ldots, \tau\}$ で，$\text{com}''^{[e]}_N \neq \text{com}'^{[e]}_N$ となるものがあれば，‘棄却’ を返す。
12. $v^{[e]} \neq 0$ なる $e \in \{1, \ldots, \tau\}$ があれば，‘棄却’ を返す。
13. $(h'_1, h'_2, h'_3) \neq (h_1, h_2, h_3)$ ならば，‘棄却’ を返す。それ以外は ‘受理’ を返す。

5.3.4.2 MQOM のパラメータ選択

MQOM の設計に必要なパラメータは，$\lambda, q, m, n, n_1, n_2, \eta, N, \tau$ である。NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出されたドキュメント [11] では，さらに効率性向上のテクニック（hypercube optimization，seed tree など）が追加されており，それを踏まえて以下のように MQOM のパラメータの見積もりが公開されている。

$(\lambda, q, m(=n), n_1, n_2, \eta, N, \tau)$	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ（平均）
$(128, 31, 49, 5, 10, 10, 256, 20)$	レベル 1	47 Bytes	78 Bytes	6,348 Bytes
$(128, 31, 49, 5, 10, 6, 32, 35)$	レベル 1	59 Bytes	102 Bytes	6,575 Bytes
$(128, 251, 43, 4, 11, 5, 256, 22)$	レベル 1	47 Bytes	78 Bytes	7,621 Bytes
$(128, 251, 43, 4, 11, 4, 32, 34)$	レベル 1	59 Bytes	102 Bytes	7,809 Bytes
$(192, 31, 77, 6, 13, 11, 256, 30)$	レベル 3	73 Bytes	122 Bytes	13,837 Bytes
$(192, 31, 77, 6, 13, 7, 32, 51)$	レベル 3	92 Bytes	160 Bytes	14,257 Bytes
$(192, 251, 68, 5, 14, 7, 256, 30)$	レベル 3	73 Bytes	122 Bytes	16,590 Bytes
$(192, 251, 68, 5, 14, 4, 32, 52)$	レベル 3	92 Bytes	160 Bytes	17,161 Bytes
$(256, 31, 106, 6, 18, 10, 256, 42)$	レベル 5	99 Bytes	166 Bytes	24,147 Bytes
$(256, 31, 106, 6, 18, 8, 32, 66)$	レベル 5	125 Bytes	218 Bytes	24,926 Bytes
$(256, 251, 93, 6, 16, 7, 256, 41)$	レベル 5	99 Bytes	166 Bytes	28,917 Bytes
$(256, 251, 93, 6, 16, 5, 32, 66)$	レベル 5	125 Bytes	218 Bytes	29,919 Bytes

5.3.5 署名方式 MiRitH

5.3.5.1 MiRitH の概要

MiRitH [2] は，MinRank 問題に関する秘匿マルチパーティ計算から MPC-in-the-Head で構成された署名方式である。MinRank 問題は 5.1.2 節でも述べたが，次のように表現することもできる。

MinRank 問題（別バージョン） 正の整数 r と行列 $\mathrm{M}_0, \ldots, \mathrm{M}_k \in \mathbb{F}_q^{m \times n}$ に対し，$\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q$ で，

$$\mathrm{Rank}\left(\mathrm{M}_0 + \sum_{i=1}^{k} \alpha_i \mathrm{M}_i\right) \leq r$$

なるものを求めよ。

もし，$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k) \in \mathbb{F}_q^k$ と $\mathrm{K} \in \mathbb{F}_q^{r \times (n-r)}$ が存在し，

$$\left(\mathrm{M}_0 + \sum_{i=1}^{k} \alpha_i \mathrm{M}_i\right) \cdot \begin{pmatrix} \mathbf{I}_{n-r} \\ \mathrm{K} \end{pmatrix} = \mathbf{0}_{m \times (n-r)} \tag{5.7}$$

となるならば，$\boldsymbol{\alpha}$ は MinRank 問題の解である。$\mathbf{M} = (\mathrm{M}_0, \ldots, \mathrm{M}_k)$ に対し，$\mathbf{M}_{\boldsymbol{\alpha}} \in \mathbb{F}_q^{m\times n}$ を

$$\mathbf{M}_{\boldsymbol{\alpha}} = \mathrm{M}_0 + \sum_{i=1}^{k} \alpha_i \mathrm{M}_i$$

とし，$\mathbf{M}^L_{\boldsymbol{\alpha}} \in \mathbb{F}_q^{m\times(n-r)}$，$\mathbf{M}^R_{\boldsymbol{\alpha}} \in \mathbb{F}_q^{m\times r}$ をそれぞれ，$\mathbf{M}_{\boldsymbol{\alpha}}$ の左側 $n-r$ 列，$\mathbf{M}_{\boldsymbol{\alpha}}$ の右側 r 列で定めると，(5.7) は $\mathbf{M}^L_{\boldsymbol{\alpha}} = \mathbf{M}^R_{\boldsymbol{\alpha}} \cdot \mathrm{K}$ と同値である。そこで，（秘匿マルチパーティ計算において設定される関係 $\mathcal{R}$ の）命題を MinRank 問題のインスタンスとし，その証拠を $\boldsymbol{\alpha}, \mathrm{K}$ としておけば，$\boldsymbol{\alpha}$ が MinRank 問題の解であることが（$\mathbf{M}_{\boldsymbol{\alpha}}$ のランクを直接計算をせずとも）効率的に検証できる。以下の検証プロトコルでは，ランダム行列 $\mathrm{R} \in \mathbb{F}_q^{s\times m}$ を用意し，$\mathrm{V} = \mathrm{R}\,(\mathbf{M}^L_{\boldsymbol{\alpha}} - \mathbf{M}^R_{\boldsymbol{\alpha}} \cdot \mathrm{K})$ とおき，$\mathrm{V} = \mathbf{0}_{s\times(n-r)}$ となることで (5.7) を確認する。(5.7) が成り立つときは $\mathrm{V} = \mathbf{0}_{s\times(n-r)}$ が必ず成り立つが，(5.7) が成り立たないときに $\mathrm{V} = \mathbf{0}_{s\times(n-r)}$ となる確率は約 $1/q^s$ である。

安全性パラメータを λ とし，以下の関数を用意する。

- Expand： 任意の λ-ビット列を入力とする疑似乱数生成関数
- $\mathcal{H}_1, \mathcal{H}_2 : \{0,1\}^* \to \{0,1\}^{2\lambda}$ ：暗号学的ハッシュ関数
- Commit：コミットメント関数

鍵生成

1. $\mathrm{seed}_{\mathrm{pk}}$, $\mathrm{seed}_{\mathrm{sk}} \in \{0,1\}^\lambda$ をランダムに選ぶ。
2. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{M}_1, \ldots, \mathrm{M}_k \in \mathbb{F}_q^{m\times n}$ を得る.
3. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q$, $\mathrm{K} \in \mathbb{F}_q^{r\times(n-r)}$, $\mathrm{E}^R \in \mathbb{F}_q^{m\times r}$ を得る。
4. $\mathrm{E}^R\mathrm{K}$ を計算し，左側の $n-r$ 列を $\mathrm{E}^R\mathrm{K}$，右側の r 列を E^R として定まる行列を $\mathrm{E} \in \mathbb{F}_q^{m\times n}$ とする。
5. $\mathrm{M}_0 = \mathrm{E} - \sum_{\ell=1}^{k} \alpha_\ell \mathrm{M}_\ell$ を計算する。

公開鍵は $\mathrm{pk} = (\mathrm{seed}_{\mathrm{pk}}, \mathrm{M}_0)$，秘密鍵は $\mathrm{sk} = \mathrm{seed}_{\mathrm{sk}}$ である。次に，署名生成である。メッセージを $M \in \{0,1\}^*$ とする。

署名生成

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \mathrm{M}_0)$ を取り出す。
2. sk から $\mathrm{seed}_{\mathrm{sk}}$ を取り出す。
3. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{M}_1, \ldots, \mathrm{M}_k \in \mathbb{F}_q^{m\times n}$ を得る.
4. $\mathrm{salt} \in \{0,1\}^{2\lambda}$ をランダムに選ぶ。
5. $e = 1, \ldots, \tau$ に対して以下を計算する：
   - 5-1. $\mathrm{seed}^{[e]} \in \{0,1\}^\lambda$ をランダムに選ぶ。
   - 5-2. $\mathrm{Expand}(\mathrm{salt},\ \mathrm{seed}^{[e]})$ の計算により，$\mathrm{seed}^{[e]}_j \in \{0,1\}^\lambda$ $(j = 1, \ldots, N)$ を得る。
   - 5-3. すべての $j = 1, \ldots, N$ に対し，$\mathrm{Expand}(\mathrm{salt}, \mathrm{seed}^{[e]}_j)$ の計算により以下を得る：
     - $j < N$ ならば，$[\![\mathrm{A}^{[e]}]\!]_j \in \mathbb{F}_q^{s\times r}$, $[\![\alpha^{[e]}_1]\!]_j, \ldots, [\![\alpha^{[e]}_k]\!]_j\ (\in \mathbb{F}_q)$, $[\![\mathrm{K}^{[e]}]\!]_j \in \mathbb{F}_q^{r\times(n-r)}$, $[\![\mathrm{C}^{[e]}]\!]_j \in \mathrm{F}_q^{s\times(n-r)}$
     - $j = N$ ならば，$[\![\mathrm{A}^{[e]}]\!]_N \in \mathbb{F}_q^{s\times r}$
   - 5-4. $[\![\alpha^{[e]}_\ell]\!]_N = \alpha_\ell - \sum_{j=1}^{N-1} [\![\alpha^{[e]}_\ell]\!]_j$ $(\ell = 1, \ldots, k)$ を計算する。
   - 5-5. $[\![\mathrm{K}^{[e]}]\!]_N = \mathrm{K} - \sum_{j=1}^{N-1} [\![\mathrm{K}^{[e]}]\!]_j$，$[\![\mathrm{C}^{[e]}]\!]_N = \mathrm{A}^{[e]}\,\mathrm{K} - \sum_{j=1}^{N-1} [\![\mathrm{C}^{[e]}]\!]_j$ を計算する。
   - 5-6. $\mathrm{state}^{[e]}_j = \begin{cases} (\mathrm{seed}^{[e]}_j) & j = 1, \ldots, N-1 \\ (\mathrm{seed}^{[e]}_N, [\![\alpha^{[e]}_1]\!]_N, \ldots, [\![\alpha^{[e]}_k]\!]_N, [\![\mathrm{K}^{[e]}]\!]_N, [\![\mathrm{C}^{[e]}]\!]_N) & j = N \end{cases}$ とおく。

5-7. コミットメント $\mathrm{com}_j^{[e]} = \mathrm{Commit}(\mathrm{salt}, e, j, \mathrm{state}_j^{[e]})$ $(j = 1, \ldots, N)$ を計算する。

6. $h_1 = \mathcal{H}_1(M, \mathrm{salt}, \mathrm{com}_1^{[1]}, \ldots, \mathrm{com}_N^{[\tau]})$ を計算する。

7. $\mathrm{Expand}(h_1)$ の計算により，$\mathrm{R}^{[e]} \in \mathbb{F}_q^{s\times m}$ $(e = 1, \ldots, \tau)$ を得る.

8. $e = 1, \ldots, \tau$ に対して以下を計算する：

8-1. $[\![\alpha_1^{[e]}]\!], \ldots, [\![\alpha_k^{[e]}]\!]$ より，$[\![\mathbf{M}_{\boldsymbol{\alpha}}^{L\,[e]}]\!]$, $[\![\mathbf{M}_{\boldsymbol{\alpha}}^{R\,[e]}]\!]$ を計算する。

8-2. $[\![\mathrm{S}^{[e]}]\!] = \mathrm{R}^{[e]}\,[\![\mathbf{M}_{\boldsymbol{\alpha}}^{R\,[e]}]\!] + [\![\mathrm{A}^{[e]}]\!]$ を計算する。

8-3. $\mathrm{S}^{[e]} = \sum_{j=1}^{N} [\![\mathrm{S}^{[e]}]\!]_j$ を計算する。

8-4. $[\![\mathrm{V}^{[e]}]\!] = \mathrm{S}^{[e]}\,[\![\mathrm{K}^{[e]}]\!] - \mathrm{R}^{[e]}\,[\![\mathbf{M}_{\boldsymbol{\alpha}}^{L\,[e]}]\!] - [\![\mathrm{C}^{[e]}]\!]$ を計算する。

8-5. $\mathrm{V}^{[e]} = \sum_{j=1}^{N} [\![\mathrm{V}^{[e]}]\!]_j$ を計算する。

8-6. $[\![\mathrm{broad}^{[e]}]\!] = ([\![\mathrm{S}^{[e]}]\!], [\![\mathrm{V}^{[e]}]\!])$, $\mathrm{broad}^{[e]} = (\mathrm{S}^{[e]}, \mathrm{V}^{[e]})$ とおく。

9. $h_2 = \mathcal{H}_2(M, \mathrm{salt}, h_1, [\![\mathrm{broad}^{[1]}]\!], \cdots, [\![\mathrm{broad}^{[\tau]}]\!])$ を計算する。

10. $\mathrm{Expand}(h_2)$ の計算により，$i^{*[1]}, \ldots, i^{*[\tau]} \in \{1, \ldots, N\}$ を得る。

11. $\mathrm{view}^{[e]}$ をすべての $j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\}$ に対する $\mathrm{state}_j^{[e]}$ のリストとする。

12. $\sigma = (\mathrm{salt}, h_1, h_2, \{\mathrm{view}^{[e]}, \mathrm{broad}^{[e]}, \mathrm{com}_{i^{*[e]}}\}_{e=1,\ldots,\tau})$ とおく。

σ が署名となる。最後に検証である。

検証

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \mathrm{M}_0)$ を取り出す。

2. σ から $(\mathrm{salt}, h_1, h_2, \{\mathrm{view}^{[e]}, \mathrm{broad}^{[e]}, \mathrm{com}_{i^{*[e]}}\}_{e=1,\ldots,\tau})$ を取り出す。

3. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{M}_1, \ldots, \mathrm{M}_k \in \mathbb{F}_q^{m\times n}$ を得る.

4. $\mathrm{Expand}(h_1)$ の計算により，$\mathrm{R}^{[e]} \in \mathbb{F}_q^{s\times m}$ $(e = 1, \ldots, \tau)$ を得る.

5. $\mathrm{Expand}(h_2)$ の計算により，$i^{*[1]}, \ldots, i^{*[\tau]} \in \{1, \ldots, N\}$ を得る。

6. $e = 1, \ldots, \tau$ に対して以下を計算する：

6-1. $\mathrm{broad}^{[e]}$ から $(\mathrm{S}^{[e]}, \mathrm{V}^{[e]})$ を取り出す。

6-2. $\mathrm{view}^{[e]}$ から $(\mathrm{state}_j^{[e]})_{j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}}$ を取り出す。

6-3. コミットメント $\mathrm{com}_j^{[e]} = \mathrm{Commit}(\mathrm{salt}, e, j, \mathrm{state}_j^{[e]})$ $(j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\})$ を計算する。

6-4. すべての $j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\}$ に対し，$\mathrm{state}_j^{[e]}$ により以下を得る：

· $j < N$ ならば，$\mathrm{seed}_j^{[e]}$

· $j = N$ ならば，$\mathrm{seed}_N^{[e]}$, $[\![\alpha_1^{[e]}]\!]_N, \ldots, [\![\alpha_k^{[e]}]\!]_N$, $[\![\mathrm{K}^{[e]}]\!]_N$, $[\![\mathrm{C}^{[e]}]\!]_N$

6-5. すべての $j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\}$ に対し，$\mathrm{Expand}(\mathrm{salt}, \mathrm{seed}_j^{[e]})$ の計算により以下を得る：

· $j < N$ ならば，$[\![\mathrm{A}^{[e]}]\!]_j$, $[\![\alpha_1^{[e]}]\!]_j, \ldots, [\![\alpha_k^{[e]}]\!]_j$, $[\![\mathrm{K}^{[e]}]\!]_j$, $[\![\mathrm{C}^{[e]}]\!]_j$

· $j = N$ ならば，$[\![\mathrm{A}^{[e]}]\!]_N$

6-6. すべての $j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\}$ に対し，$[\![\alpha_1^{[e]}]\!]_j, \ldots, [\![\alpha_k^{[e]}]\!]_j$ より，$[\![\mathbf{M}_{\boldsymbol{\alpha}}^{L\,[e]}]\!]_j$, $[\![\mathbf{M}_{\boldsymbol{\alpha}}^{R\,[e]}]\!]_j$ を計算する。

6-7. $[\![\mathrm{S}^{[e]}]\!]_j = \mathrm{R}^{[e]}\,[\![\mathbf{M}_{\boldsymbol{\alpha}}^{R\,[e]}]\!]_j + [\![\mathrm{A}^{[e]}]\!]_j$ $(j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\})$ を計算する。

6-8. $[\![\mathrm{V}^{[e]}]\!]_j = \mathrm{S}^{[e]}\,[\![\mathrm{K}^{[e]}]\!]_j - \mathrm{R}^{[e]}\,[\![\mathbf{M}_{\boldsymbol{\alpha}}^{L\,[e]}]\!]_j - [\![\mathrm{C}^{[e]}]\!]_j$ $(j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\})$ を計算する。

6-9. $[\![\mathrm{broad}^{[e]}]\!]_j = ([\![\mathrm{S}^{[e]}]\!]_j, [\![\mathrm{V}^{[e]}]\!]_j)$ $(j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\})$ とおく。

6-10. $[\![\mathrm{broad}^{[e]}]\!]_{i^{*[e]}} = \mathrm{broad}^{[e]} - \sum_{j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}} [\![\mathrm{broad}^{[e]}]\!]_j$ を計算する。

7. $h_1' = \mathcal{H}_1(M, \mathrm{salt}, \mathrm{com}_1^{[1]}, \ldots, \mathrm{com}_N^{[\tau]})$ を計算する。

8. $h_2' = \mathcal{H}_2(M, \mathrm{salt}, h_1', [\![\mathrm{broad}^{[1]}]\!], \cdots, [\![\mathrm{broad}^{[\tau]}]\!])$ を計算する。

9. $\mathrm{V}^{[e]} \neq \mathbf{0}_{s\times(n-r)}$ なる $e \in \{1, \ldots, \tau\}$ があれば，'棄却' を返す。

10. $(h'_1, h'_2) \neq (h_1, h_2)$ ならば，‘棄却’ を返す。それ以外は ‘受理’ を返す。

5.3.5.2 MiRitH のパラメータ選択

MiRitH の設計に必要なパラメータは，$\lambda, q, m, n, k, r, s, N, \tau$ である。NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出されたドキュメント [2] では，さらに効率性向上のテクニック（hypercube optimization，seed tree など）が追加されており，それを踏まえて以下のように MiRitH のパラメータの見積もりが公開されている。

$(\lambda, q, m(=n), k, r, s, N, \tau)$	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ（平均）
$(128, 16, 15, 78, 6, 5, 16, 39)$	レベル 1	129 Bytes	16 Bytes	7,661 Bytes
$(128, 16, 15, 78, 6, 9, 256, 19)$	レベル 1	129 Bytes	16 Bytes	5,665 Bytes
$(128, 16, 16, 142, 4, 5, 16, 39)$	レベル 1	144 Bytes	16 Bytes	8,800 Bytes
$(128, 16, 16, 142, 4, 9, 256, 19)$	レベル 1	144 Bytes	16 Bytes	6,298 Bytes
$(192, 16, 19, 109, 8, 7, 16, 55)$	レベル 3	205 Bytes	24 Bytes	16,668 Bytes
$(192, 16, 19, 109, 8, 9, 256, 29)$	レベル 3	205 Bytes	24 Bytes	12,423 Bytes
$(192, 16, 19, 167, 6, 7, 16, 55)$	レベル 3	205 Bytes	24 Bytes	17,882 Bytes
$(192, 16, 19, 167, 6, 9, 256, 29)$	レベル 3	205 Bytes	24 Bytes	13,115 Bytes
$(256, 16, 21, 189, 7, 7, 16, 74)$	レベル 5	253 Bytes	32 Bytes	29,568 Bytes
$(256, 16, 21, 189, 7, 10, 256, 38)$	レベル 5	253 Bytes	32 Bytes	21,763 Bytes
$(256, 16, 22, 254, 6, 7, 16, 74)$	レベル 5	274 Bytes	32 Bytes	31,980 Bytes
$(256, 16, 22, 254, 6, 10, 256, 38)$	レベル 5	274 Bytes	32 Bytes	23,144 Bytes

5.4 多変数多項式に基づく暗号技術に関するまとめ

1984 年に，Ong と Schnorr が多変数 2 次多項式を利用した署名方式 [21] を提案した。したがって，多変数多項式を利用した暗号技術は，既に 40 年以上の歴史を持つことになる。Ong と Schnorr の署名方式は，合成数を法とする剰余環を係数としており，合成数の素因数分解ができないことを安全性の仮定としていたが，1988 年に，松本と今井により，初めて有限体を係数とした多変数多項式を利用した暗号化方式 [20] が提案された。これ以降，MQ 問題の解読困難性を安全性の仮定とする方式が数多く提案されており，現在に至る。

多変数公開鍵暗号の暗号化方式，および，署名方式の多くは双極型システムを用いて構成されている。双極型システムは，暗号化や検証が効率的に実行できるという特徴を持つ。双極型システムを用いて構成されている署名方式 UOV も，この特徴を持ち，さらに，署名長が短いという特徴も持っている。一方で，双極型システムは公開鍵長が大きくなりやすいという課題もある。UOV に対しては，公開鍵長を削減する改良を加えた変種として QR-UOV や MAYO が提案されている。

一方で，MQ 問題や MinRank 問題に付随する秘匿マルチパーティ計算から MPC-in-the-Head の枠組みを利用して署名方式を構成することができる。こちらは方式が提案されてからまだ数年しかたっていないということもあり，今後の研究動向を見守る必要がある。

第 5 章の参照文献

[1] G. Adj, L. Rivera-Zamarripa, J. A. Verbel. MinRank in the Head: Short Signatures from Zero-Knowledge Proofs. (2022). https://eprint.iacr.org/2022/1501.

[2] G. Adj, L. Rivera-Zamarripa, J. A. Verbel, E. Bellini, S. Barbero, A. Esser, C. Sanna, F. Zweydinger. MiRitH. 2022-08. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/mirith-submission.zip. Submission to the NIST's Post-Quantum Cryptography: Additional Digital Signature Schemes, round 1.

[3] M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. A. Perlner, D. Smith-Tone, J.-P. Tillich, J. A. Verbel. Improvements of Algebraic Attacks for Solving the Rank Decoding and MinRank Problems. ASIACRYPT (1). Vol. 12491. Lecture Notes in Computer Science. Springer, 2020, pp. 507–536.

[4] E. Bellini, A. Esser, C. Sanna, J. Verbel. MR-DSS – Smaller MinRank-based (Ring-)Signatures. (2022). https://eprint.iacr.org/2022/973.

[5] W. Beullens. MAYO: Practical Post-quantum Signatures from Oil-and-Vinegar Maps. SAC. Vol. 13203. Lecture Notes in Computer Science. Springer, 2021, pp. 355–376.

[6] W. Beullens, F. Campos, S. Celi, B. Hess, M. J. Kannwischer. MAYO. 2022-08. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/mayo-submission.zip. Submission to the NIST's Post-Quantum Cryptography: Additional Digital Signature Schemes, round 1.

[7] W. Beullens et al. UOV. 2022-08. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/UOV-submission.zip. Submission to the NIST's Post-Quantum Cryptography: Additional Digital Signature Schemes, round 1.

[8] N. T. Courtois. Efficient Zero-Knowledge Authentication Based on a Linear Algebra Problem MinRank. ASIACRYPT. Vol. 2248. Lecture Notes in Computer Science. Springer, 2001, pp. 402–421.

[9] J. Ding, J. E. Gower, D. S. Schmidt. Multivariate Public Key Cryptosystems. Vol. 25. Advances in Information Security. Springer, 2006.

[10] J.-C. Faugère, M. S. El Din, P.-J. Spaenlehauer. Computing loci of rank defects of linear matrices using Gröbner bases and applications to cryptology. ISSAC. ACM, 2010, pp. 257–264.

[11] T. Feneuil, M. Rivain. MQOM (MQ on my mind). 2022-08. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/mqom-submission.zip. Submission to the NIST's Post-Quantum Cryptography: Additional Digital Signature Schemes, round 1.

[12] H. Furue, Y. Ikematsu, Y. Kiyomura, T. Takagi. A New Variant of Unbalanced Oil and Vinegar Using Quotient Ring: QR-UOV. ASIACRYPT (4). Vol. 13093. Lecture Notes in Computer Science. Springer, 2021, pp. 187–217.

[13] H. Furue, Y. Ikematsu, Y. Kiyomura, T. Takagi, K. Yasuda, T. Miyazawa, T. Saito, A. Nagai. QR-UOV. 2022-08. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/QR-UOV-submission.zip. Submission to the NIST's Post-Quantum Cryptography: Additional Digital Signature Schemes, round 1.

[14] M. R. Garay, D. S. Johnson. A Guide to the Theory of NP-Completeness. In Computers and Intractability. W.H. Freeman, 1979.

[15] L. Goubin, N.T. Courtois. Cryptanalysis of the TTM Cryptosystem. ASIACRYPT. Vol. 1976. Lecture Notes in Computer Science. Springer, 2000, pp. 44–57.

[16] Y. Hashimoto. An improvement of algorithms to solve under-defined systems of multivariate quadratic equations. JSIAM Letters. Vol. 15 (2023), pp. 53–56. https://www.jstage.jst.go.jp/article/jsiaml/15/0/15_53/_article.

[17] Y. Ishai, E. Kushilevitz, R. Ostrovsky, A. Sahai. Zero-knowledge from secure multiparty computation. STOC. ACM, 2007, pp. 21–30.

[18] A. Kipnis, L. Patarin, L. Goubin. Unbalanced Oil and Vinegar Signature Schemes. EUROCRYPT. Vol. 1592. Lecture Notes in Computer Science. Springer, 1999, pp. 206–222.

[19] A. Kipnis, A. Shamir. Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. CRYPTO. Vol. 1666. Lecture Notes in Computer Science. Springer, 1999, pp. 19–30.

[20] T. Matsumoto, H. Imai. Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. EUROCRYPT. Vol. 330. Lecture Notes in Computer Science. Springer, 1988, pp. 419–453.

[21] H. Ong, C.-P. Schnorr. Signatures through Approximate Representation by Quadratic Forms. CRYPTO. Plenum Press, New York, 1983, pp. 117–131.

[22] J. Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. EUROCRYPT. Vol. 1070. Lecture Notes in Computer Science. Springer, 1996, pp. 33–48.

[23] B. Santoso, Y. Ikematsu, S. Nakamura, T. Yasuda. Three-Pass Identification Scheme Based on MinRank Problem with Half Cheating Probability. arXiv: 2205.03255.

第 6 章

同種写像に基づく暗号技術

本章では同種写像に基づく暗号技術についてまとめる。同種写像に基づく暗号技術の安全性は，同種写像問題を解く計算の困難性及び（それと同値な）自己準同型環計算問題の困難性に依存しており，同種写像暗号に関する研究はこれまで継続して進められている。特に，本報告書の 2022 年度版と比べて，高次元同種写像計算を利用した鍵共有・署名構成に進展が見られている（6.3.1.2 節及び 6.4 節参照）。

6.1 節では，安全性の根拠となる問題として，同種写像問題の一般形を述べた後，自己準同型環計算問題及び SQIsign（Short Quaternion and Isogeny Signature）署名方式 [23] の安全性に関する計算問題の順に，その概要を記述していく。6.2 節では，代表的な暗号方式として，自己準同型環計算問題に基づく GPS（Galbraith–Petit–Silva）署名方式 [27] を取り上げる。6.3 節では，主要な暗号方式として，GPS 署名方式を改良した SQIsign 署名方式を解説する。

本章では，超特異楕円曲線を用いた暗号技術を主に扱う。しかし，通常楕円曲線に基づく CRS （Couveignes–Rostovtsev–Stolbunov）鍵共有法 [14, 49] を改良した De Feo ら [22] の方式は，それ自体は実用的な性能にはまだ遠いが，調査報告書に記載した CSIDH 鍵共有の原型を与えているという点で重要である。また，群作用暗号を量子マネーへ応用した [58, 41] では，通常楕円曲線が用いられている。

同種写像の数学的詳細については，De Feo の概説記事 [21] や Washington の楕円曲線の教科書 [56] を参照のこと。和書では，相川らによる概説書 [59] において，同種写像暗号に必要な数学も詳しく説明されている。また，Galbraith–Vercauteren による同種写像関連問題のサーベイ [28] も参照する。

■記法 $x \leftarrow_R X$ は，x を有限集合 X から一様ランダムにサンプリングすることを表す。以下では，有限体上に定義された楕円曲線のみを扱い，同種写像暗号では，多くの場合，モンゴメリ型の楕円曲線定義式 $E_{a,b} : by^2 = x^3 + ax^2 + x$ が用いられる。標数 p の有限体 $\mathbb{F}$ 上定義された楕円曲線 E に対し，O_E は E の無限遠点であり，$\mathbb{F}$ の拡大体 $\mathbb{K}$ に対して，$\mathbb{K}$-有理点群は $E(\mathbb{K}) := \{(x, y) \in \mathbb{K}^2 \,|\, (x, y) \text{ は } E \text{ の定義式を満たす }\} \cup \{O_E\}$ で与えられる。また，正整数 r に対して E の r-ねじれ部分群は $E[r] := \{P \in E(\overline{\mathbb{F}}_p) \,|\, rP = O_E\}$ で与えられる。ここで $\overline{\mathbb{F}}_p$ は有限体 $\mathbb{F}_p$ の代数閉包を表す。

6.1 同種写像に基づく暗号技術の安全性の根拠となる問題

6.1.1 節で同種写像問題の一般形を述べた後，自己準同型環計算問題と SQIsign 署名方式の安全性に関する計算問題（6.1.2 節）の概要及びそれら問題に対する解析状況について記述していく。

6.1.1 同種写像問題の一般形

同種写像とは，2 つの楕円曲線 E, E' の間の写像 φ であり，E の座標 (x, y) の有理式で与えられると共に，楕円曲線の加法構造に関する準同型性，即ち $\varphi(P+Q) = \varphi(P) + \varphi(Q)$，を有する非零写像である。（その正確な定義は，前掲の各文献を参照のこと。）また，E, E' の間に，同種写像 φ が存在する時に，E と E' は同種であるという。

同種写像 φ は，その核 $C = \ker(\varphi)$ によって決まるので，φ の定義域曲線（始点曲線）E に対して φ の値域となる楕円曲線を E/C と書き表す，すなわち，$\varphi : E \to E/C$。核 $C = \ker(\varphi)$ の位数がセキュリティパラメータ λ の多項式サイズであれば，$C = \ker(\varphi)$ の生成元から φ を効率的に計算するアルゴリズムが Vélu によって与えられている [54]。（モンゴメリ型楕円曲線に対する Vélu の公式に関しては，[46] を参照のこと。） 特に核の位数 $\#C$ が小素数になる同種写像を同種写像基本演算として，それらの合成が同種写像暗号での基本的な暗号演算を与えることになる。そして，その合成における基本演算の組み合わせ方法が，秘密鍵情報を与える。

つまり，同種な楕円曲線の間の同種写像を計算することを要求する次の同種写像問題が，具体的な暗号方式の安全性を根拠づける次節以降の諸問題の基本形となる。（超特異同種写像問題と自己準同型環計算問題との計算量的同値性に関しては 6.1.2 節で触れる。）

定義 6.1 (一般形同種写像問題 [28]) *2 つの同種な楕円曲線 E, E' に対して，同種写像 φ を計算せよ。（φ のコンパクトな表現を与えよ。）*

ここで，「φ のコンパクトな表現」とは，様々な表現方法が考えられる。例えば，$\deg(\varphi)$ が小素数 ℓ_i によって $\prod_i \ell_i^{e_i}$ となっている場合には，この分解に沿って φ を分解した各 ℓ_i 次同種写像の像に現れる値域楕円曲線（又は j 不変量）の列挙で与えることができる。また，SIDH 同種写像問題の設定では，核の生成点が，同種写像のコンパクトな表現を与える。また，CSIDH 鍵共有では虚 2 次整環（オーダー）のイデアル類によって同種写像が表現される。そして，SIDH 鍵共有に対する攻撃法は，楕円曲線同種写像に対する新しい表現法を与えた [48]。最近の高次元同種写像を用いた同種写像暗号の進展は，この新しい同種写像表現法に基づいて行われている。これらに関しては，調査報告書を参照のこと。

定義 6.1 において，φ の次数が多項式サイズであれば，この問題は簡単に解けるので，φ の次数は通常は指数サイズのものを考える。また，CSIDH 鍵共有では $\mathbb{F}_p$-有理な楕円曲線のみを対象とするので，$\overline{\mathbb{F}}_p$-同型であるが $\mathbb{F}_p$-同型でないツイスト曲線を判別する必要性があるが，ツイスト曲線は j 不変量では判別できない。これにより，Galbraith ら [28] は j 不変量を使って同種写像問題を定式化しているが，上ではあえて，より素朴な形を採用して，2 つの同種な楕円曲線 E, E' を使って同種写像問題を提示した。

同種写像問題の初期の考察には，自己準同型環計算を扱った Kohel の博士論文 [34] や Galbraith による同種写像問題に関する研究 [26] 及び Couveignes と Rostovtsev–Stolbunov による初期の暗号応用への提案 [14, 49] がある。その後，Charles らによる同種写像に基づいたハッシュ関数の提案 [9] は，同種写像一方向性関数を一方向性の観点からだけでなく，衝突困難性の観点からも見直すことになり，初期の同種写像暗号の研究では重要な役割を果たした。特に，同種写像グラフがエクスパンダーグラフであることに着目して暗号に応用した意義は大きい。

■超特異同種写像問題と通常同種写像問題　標数 p の有限体上の楕円曲線 E の p-ねじれ部分群 $E[p]$ が，$E[p] = \{O_E\}$ の時，E を超特異楕円曲線といい，そうでない時，E を通常楕円曲線という。超特異楕円曲線の j 不変量は，$\mathbb{F}_{p^2}$ の要素である。つまり，超特異 j 不変量の個数は，有限個であり，具体的に $[p/12] + \epsilon$（但し $\epsilon \in \{0, 1, 2\}$）で与えられる。超特異，通常という楕円曲線の性質は，同種写像によって保存されるため，同種写像問題も，この 2 つの性質によって，

超特異同種写像問題と通常同種写像問題という 2 つの問題に分類される。

■超特異同種写像問題の計算困難性　超特異同種写像問題は，調査報告書で述べられる CSIDH 鍵共有や CSI-FiSh 署名方式の安全性に関する計算問題の一般形であり，その計算困難性を評価することは重要である。また，自己準同型環計算問題との関係性については 6.1.2 節を参照のこと。

超特異同種写像問題の古典計算機による解読時間は $\tilde{O}(\sqrt{p})$，量子計算機による解読時間は $\tilde{O}(\sqrt[4]{p})$ と見積もられている。Kohel [34] による超特異同種写像グラフ上のアルゴリズム解析に基づいて，現在最良の古典解読アルゴリズムは Delfs–Galbraith [16] によるもの及びその改良 [51] で，解読時間は $\tilde{O}(\sqrt{p})$ である。Delfs–Galbraith アルゴリズムでは $\mathbb{F}_p$ 上の超特異楕円曲線からなる部分グラフが利用されている。量子解読アルゴリズムは Biasse ら [5] によって時間計算量が $\tilde{O}(\sqrt[4]{p})$ の量子アルゴリズムが知られている。これは，$\mathbb{F}_p$ 上の超特異楕円曲線の同種写像問題に対する準指数時間量子アルゴリズム [11] と Grover アルゴリズムに基づく $\tilde{O}(\sqrt[4]{p})$ の道探索アルゴリズムを結合したものである。

また，Costello ら [13]，Longa ら [38] による報告，Udovenko–Vitto [53] による$IKEp182 Challenge [12] 解読報告，Jaques–Schanck [31] による同種写像問題に対する（量子）安全性評価報告は，いずれも SIDH 鍵共有（及び SIKE 暗号方式 [30]）法への攻撃として提案されているが，多くの部分は一般的な超特異同種写像問題に関する知見としても有効であることに注意する。更に，固定次数の同種写像を計算する問題に対して，CRYPTO 2024 において Benčina ら [4] により改善された古典/量子アルゴリズムが提案されている。

6.1.2　自己準同型環計算問題と SQIsign 署名方式の安全性に関する計算問題

6.1.2.1　自己準同型環計算問題

同種写像暗号は，Kohel [34]，Galbraith [26]，Couveignes [14] らの先駆的研究にその起源をもつが，特に，Kohel は有限体上の楕円曲線の自己準同型環を計算するアルゴリズムを探求しており，そのために楕円曲線の同種写像からなる「同種写像グラフ」の性質を見極めることから始めて，目的とする自己準同型環計算を同種写像グラフ上のアルゴリズム構成に帰着していく。その後，Kohel–Lauter–Petit–Tignol [35] は，この「同種写像計算」と「自己準同型環計算」を並置しながら考察する視点を，「構成的 Deuring 対応」として計算論的観点から捉え直した（表 6.1 参照）。そこでは，四元数環側での ℓ-同種写像道探索問題を解く KLPT アルゴリズムが鍵となるアルゴリズムである（[33, 20] 参照）。そして，この構成的 Deuring 対応に基づき「同種写像計算」と「自己準同型環計算」の等価性が示されており [18, 19, 57]，現在，自己準同型環計算問題の困難性に基づいた暗号構成の研究が進められている [27, 23, 24]。

■自己準同型環計算問題とその超特異同種写像計算問題との同値性　以下の記述に関しては，例えば [36] を参照する。また，四元数環については Voight の教科書 [55] に詳しい説明がある。有理数体 $\mathbb{Q}$ 上 $\{1, i, j, k\}$ を基底とするベクトル空間でありかつ $a, b \in \mathbb{Q}$ により $i^2 = a, j^2 = b, k = ij = -ji$ という積構造が入った $\mathbb{Q}$ 上の代数（環）を四元数環 $\mathcal{B}$ と呼ぶ。各素点 ν（素数または ∞）における $\mathbb{Q}$ の完備化 $\mathbb{Q}_\nu$ による $\mathcal{B} \otimes \mathbb{Q}_\nu$ が $\nu = p, \infty$ の時にのみ斜体（可除環）になる四元数環 $\mathcal{B} = \mathcal{B}_{p,\infty}$ を扱う。これを，$\mathcal{B}_{p,\infty}$ は p, ∞ の 2 点のみで分岐する四元数環であるといい，$\mathcal{B}_{p,\infty}$ は同型を除いて一意的に決まる。この同じ素数 p を標数とする有限体上の超特異楕円曲線 E の自己準同型写像がなす環 $\mathrm{End}(E)$ は E の自己準同型環と呼ばれて，$\mathrm{End}(E)$ は $\mathcal{B}_{p,\infty}$ の極大整環 $\mathcal{O}$ になっている[*1]。ここで，（四元数環の）整環とは $\mathbb{Z}$ 上階数 4 の加群でありかつ環であるものであり，極大整環とは，そのような整環の中で包含関係に関して極大になっているものを指す。この自己準同型環 $\mathrm{End}(E)$ を計算する以下の問題が基本である。

定義 6.2 (自己準同型環計算問題 [34]) 超特異楕円曲線 E が与えられて，E の自己準同型環 $\mathrm{End}(E)$ を計算せよ。

[*1] 自己準同型写像は英語で endomorphism であるので，その全体を $\mathrm{End}(E)$ で表す。

Eisenträger らの研究 [18, 19] により，超特異同種写像計算問題と（超特異）自己準同型環計算問題の間に多項式時間帰着による計算問題としての同値性が示された。そこではヒューリスティックな仮定が使われていたが，Wesolowski [57] は，一般化されたリーマン予想に基づいて，その同値性に対して厳密な証明を与えた。また，[45, 40] においては，非スカラー自己準同型写像を計算する問題（One Endomorphism Problem）と自己準同型環計算問題の等価性も示されている。

6.1.1 節で，超特異同種写像問題の古典計算機による現在最速の解読時間は $\tilde{O}(\sqrt{p})$ と見積もられていたので，この同値性により，自己準同型環計算問題も同等の計算時間であるが，直接に，自己準同型環計算問題を解く研究も進められており，[19] において，$\tilde{O}(\sqrt{p})$ 時間の自己準同型環計算（古典）アルゴリズムが報告されており，その後 [25] により改良されている。また，神戸ら [32] によって，10 から 30 bits の素数 p に対する自己準同型環計算の実装報告がなされている。

■Deuring 対応　自己準同型環計算問題で与えられる楕円曲線 E から極大整環 $\mathcal{O}$ への対応は，表 6.1 に掲げたように，楕円曲線に関する様々な概念から四元数環に関する概念への対応に拡張される。その詳細に関しては，例えば [36, 第 2 章] を参照していただきたいが，特に基本的な対応としては，同種写像 $\varphi : E \to E_1$ が，極大整環の間の同型 $\mathcal{O} \cong \mathrm{End}(E), \mathcal{O}_1 \cong \mathrm{End}(E_1)$ を通して，左 $\mathcal{O}$-整イデアルかつ右 $\mathcal{O}_1$-整イデアルである I_φ に対応していることである。これにより始点曲線 E を固定すると，同種写像 $\varphi : E \to E_1$ の終点曲線 E_1 が $\mathcal{O}$ のイデアル類と対応することがわかり，超特異 j 不変量（$\in \mathbb{F}_{p^2}$）の集合がイデアル類集合 $\mathrm{cl}(\mathcal{O})$ と一対一に対応していることもわかる。

一般に表 6.1 に示されるように，幾何的な情報から成る楕円曲線側のデータと代数的な情報から成る四元数環側のデータの間に対応関係が存在しており，Deuring 対応と呼ばれる。自己準同型環計算問題（定義 6.2）は Deuring 対応に基づいた問題であり，楕円曲線側の超特異 j 不変量 $j(E)$ から対応する四元数環側の極大整環 $\mathcal{O} = \mathrm{End}(E)$ を計算する問題となっている。そして，この Deuring 対応は，6.2.1 節及び 6.3.1 節での暗号構成を理解する際にも重要な鍵となっている。

表 6.1: Deuring 対応

楕円曲線側	四元数環側
超特異 j 不変量 $j(E) \in \mathbb{F}_{p^2}$（の $\mathbb{F}_{p^2}/\mathbb{F}_p$-Galois 共役類）	$\mathcal{B}_{p,\infty}$ 内の極大整環 $\mathcal{O} = \mathrm{End}(E)$ の自己同型類（タイプ）
同種写像 $\varphi : E \to E_1$ で定まる (E_1, φ)	左 $\mathcal{O}$-整イデアルかつ右 $\mathcal{O}_1$-整イデアルである I_φ
自己準同型写像 $\theta \in \mathrm{End}(E)$	主イデアル $\mathcal{O}\theta$
同種写像の次数 $\deg(\varphi)$	イデアルのノルム $n(I_\varphi)$
双対同種写像 $\hat{\varphi}$	共役イデアル $\overline{I_\varphi}$
同じ定義域・値域の同種写像 $\varphi : E \to E_1, \psi : E \to E_1$	同値なイデアル $I_\varphi \sim I_\psi$
超特異 j 不変量 $j(E) \in \mathbb{F}_{p^2}$ の集合	イデアル類の集合 $\mathrm{cl}(\mathcal{O})$
同種写像の合成 $\tau \circ \rho : E \to E_1 \to E_2$	イデアル積 $I_{\tau\circ\rho} = I_\rho \cdot I_\tau$
N-同種写像の同型類	レベル N の Eichler 整環の類集合

6.1.2.2　SQIsign 署名方式の安全性に関する計算問題

次に，SQIsign 署名方式 [23, 10] の安全性を示すために必要な計算問題を述べる。近年進展が著しい SQIsign2D 署名方式の安全性に関しては [3, 44] などを参照のこと。

■SQIsign 署名方式の健全性に関する計算問題　まずは，SQIsign 署名方式の健全性（偽造不可能性）を示すための計算問題である超特異平滑自己準同型写像計算問題（Smooth Endomorphism Problem: SEP）を定義する。以下では，核が巡回群となる自己準同型写像を巡回自己準同型写像と呼ぶ。

定義 6.3 (超特異平滑自己準同型写像計算問題 [23, 10]) 超特異楕円曲線 E が与えられて，平滑な整数を次数にもつ E 上の（非自明な）巡回自己準同型写像を見つけよ。

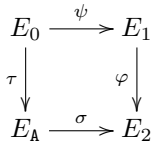
この問題で問うているような非自明な自己準同型写像が計算できれば，[19] で見るように，自己準同型環 $\mathrm{End}(E)$ 全体も計算できることが知られているので，この問題は，本質的に自己準同型環計算問題と同値である [23]。よって，$\tilde{O}(\sqrt{p})$ 時間での古典アルゴリズム [19] が現状最速と見積もられる。

■特殊極値的楕円曲線　次に，SQIsign 署名方式の零知識性を示すための計算問題を述べるが，公開パラメータで重要となる楕円曲線 E_0 を示す。$p = 3 \bmod 4$ の時，j 不変量 $j = 1728$ となる $E_0 : y^2 = x^3 + x$ の $\mathcal{O}_0 = \mathrm{End}(E_0)$ は $i^2 = -1, j^2 = -p$ となる $\mathcal{O}_0 = \mathbb{Z} + \mathbb{Z}i + \mathbb{Z}\frac{i+j}{2} + \mathbb{Z}\frac{1+ij}{2}$ となることが知られている。更に具体的に自己準同型写像 $\iota : (x, y) \mapsto (-x, \sqrt{-1}y), \pi : (x, y) \mapsto (x^p, y^p)$ により $\mathrm{End}(E_0) = \mathbb{Z} + \mathbb{Z}\iota + \mathbb{Z}\frac{\iota+\pi}{2} + \mathbb{Z}\frac{1+\iota\pi}{2}$ で与えられる。

標数 p と ∞ のみで分岐する四元数環 $\mathcal{B}_{p,\infty} := \mathbb{Q}[i, j]$ における極大整環 $\mathcal{O}_0 \subset \mathcal{B}_{p,\infty}$ は，最小判別式の 2 次整環である $\mathfrak{O} \subset \mathcal{O}_0 \cap \mathbb{Q}[i]$ による $\mathfrak{O} + j\mathfrak{O} \subset \mathcal{O}_0$ が部分整環であり $\mathfrak{O} \subset (j\mathfrak{O})^\perp$ と直交分解しているとき[*2]，特殊極値的（special extremal）であるという。詳細は [23, 36] を参照。$p = 7 \bmod 12$ の時，上述の E_0 に対して $\mathrm{End}(E_0)$ は特殊極値的であり，この時，E_0 は特殊極値的曲線と呼ばれる。特殊極値的曲線 E_0 は，その自己準同型環の構造が簡単で計算上扱いやすいため GPS 署名方式 及び SQIsign 署名方式の公開パラメータの一部として必要である。

■SQIsign 署名方式の零知識性に関する計算問題　SQIsign 署名方式では，右図の同種写像 τ が秘密鍵で，超特異楕円曲線 $E_\mathtt{A}$ が公開鍵（の主要な一部）である。署名生成では，同種写像 ψ, φ を適切に生成して得られた合成写像 $\varphi \circ \psi \circ \hat{\tau}$ を「ランダム化」した同種写像 σ を署名とする[*5]。その詳細は 6.3.1.1 節を参照。[23, 24] において定義された E_0 を始点とする同種写像から成るある集合 $\mathcal{P}_{N_\tau}$ を τ によって $E_\mathtt{A}$ を始点とした同種写像に移した集合 $[\tau]_*\mathcal{P}_{N_\tau}$（$\mathcal{P}_{N_\tau}$ の τ による pushforward）を考える。正しく生成された署名同種写像 σ は $[\tau]_*\mathcal{P}_{N_\tau}$ に属するのであるが，それが $E_\mathtt{A}$ を始点とした 2 べき次数 $D\,(= 2^e)$ の巡回同種写像全体 $\mathtt{Iso}_{D,j(E_\mathtt{A})}$ から一様ランダムにサンプリングしたのと区別が付くかという問題が以下であり，SQIsign 署名方式の零知識性を示すために必要である。

SQIsign 同種写像図式



CSI-FiSh，GPS 図式と同様に可換図式ではない。

定義 6.4 (SQIsign 署名方式のランダム識別問題 [23, 10]) $\tau : E_0 \to E_\mathtt{A}$ を秘密同種写像として，楕円曲線 E_0 を含む *SQIsign* 署名方式の公開パラメータ $pp_\mathtt{sqisign}$（詳しくは *6.3.1* 節参照）と公開鍵 $E_\mathtt{A}$ が入力として与えられると共に，$[\tau]_*\mathcal{P}_{N_\tau}$ から一様サンプリングして返すオラクル O_τ への多項式回のアクセスが許される時に，$E_\mathtt{A}$ を始点とする同種写像 σ が与えられて σ が $\mathtt{Iso}_{D,j(E_\mathtt{A})}$ から一様ランダムに選ばれたか，$[\tau]_*\mathcal{P}_{N_\tau}$ から一様ランダムに選ばれたかを判定せよ。

SQIsign 署名方式の提案者によると，現在のところ，SQIsign 署名方式のランダム識別問題を解くのに，E_0 と $E_\mathtt{A}$ の

[*2] $\mathcal{B}_{p,\infty}$ における内積は $\alpha, \beta \in \mathcal{B}_{p,\infty}$ に対して $\frac{1}{2}\mathrm{tr}(\alpha\bar{\beta})$ で与えられて，ここは，その内積に関する直交分解である（$\mathcal{B}_{p,\infty}$ 内のトレース，共役の定義は，例えば [36] を参照のこと）。

[*5] $\hat{\tau}$ は τ の双対同種写像である。表 6.1 も参照のこと。

情報から τ を暴く攻撃法より効率の良い攻撃法はまだ知られていないとのことである [23, 36]。つまり，$\tilde{O}(\sqrt{p})$ 時間を必要とすると見積もられている。

また，上述の SQIsign 署名方式に関する計算問題は，どちらも補助点を問題に含まないことにより，最近の SIDH 同種写像問題に対する攻撃法が適用できないことに注意する。

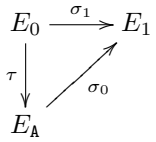
6.2 同種写像に基づく代表的な暗号方式

6.2.1 GPS 署名方式

Galbraith–Petit–Silva（GPS）[27] によって始めて自己準同型環の知識証明に基づく署名方式が提案された。GPS 署名方式は 1 bit チャレンジ空間の零知識証明プロトコルに基づいているため実際に利用するのは困難であろうと思われているが，現在，GPS 署名方式は，SQIsign 署名方式の原型を与えているという点で重要である。6.1.2 節で述べた Deuring 対応と KLPT アルゴリズム [35] が GPS 署名方式の理論的基礎を与える。

右図において E_0 は 6.1.2.2 節で与えた $j(E_0) = 1728$ なる楕円曲線（特殊極値的楕円曲線）であり，そこで見たようにその E_0 に関しては $\mathrm{End}(E_0)$ の構造が簡明な形で与えられている。その楕円曲線 E_0 からの秘密鍵同種写像 $\tau : E_0 \to E_{\mathtt{A}}$ を知っている証明者（署名生成者）は，$E_{\mathtt{A}}$ から別の楕円曲線 E_1 への同種写像 $\sigma_0 : E_{\mathtt{A}} \to E_1$ と τ との合成 $\sigma_0 \circ \tau : E_0 \to E_1$ を KLPT アルゴリズムに基づいて「ランダム化」して同じ始点 E_0 と終点 E_1 をもつ $\sigma_0 \circ \tau$ とは異なる同種写像 $\sigma_1 : E_0 \to E_1$ を得ることができる。

GPS 同種写像図式



さらに，自己準同型環 $\mathrm{End}(E_{\mathtt{A}})$ を計算する問題の困難性に基づけば，このようなランダム化ができるのは，τ を知っている証明者に限られるので，チャレンジ bit $c \in \{0,1\}$ を送って証明者に同種写像 σ_c を答えさせることにより，τ に関する知識の有無を検査することができて，認証・署名方式が構成できる。それが GPS 認証方式，そしてその Fiat–Shamir 変換署名が GPS 署名方式である。ここでは [27, 第 4 章] と [36, 5.1.2 節] に基づいて GPS 署名方式を記述する。また，[27, 第 4 章] では，通常の Fiat–Shamir 変換を施した署名方式と Unruh 変換を施した署名方式の 2 方式が記述されているが，ここでは記述の簡便さを考慮して前者の記述を基にして以下に署名方式を与える。

鍵生成： 既知の特殊極値的自己準同型環 $\mathcal{O}_0$ をもつ超特異楕円曲線 E_0 とする。互いに素な B-べき平滑数 S_1, S_2[*6]を，S_1, S_2 次の同種写像グラフ上ランダムウォークがグラフのエクスパンダー性により一様分布を導く程度に十分大きくとる。セキュリティパラメータ λ に対して $t := \lambda$（または $t := 2\lambda$）として，t bits 出力のハッシュ関数 H を選ぶ。$pp_{\mathtt{gps}} := (E_0, S_1, S_2, H)$ を公開パラメータとする。さらに，E_0 を始点とする S_1 次のランダムな同種写像 $\tau : E_0 \to E_{\mathtt{A}}$ を計算して，$pp_{\mathtt{gps}}$ と $E_{\mathtt{A}}$ を公開鍵として，τ を秘密鍵とする。

署名生成： 各 $i = 1, \ldots, t$ に関して $E_{\mathtt{A}}$ を始点とする S_2 次のランダムな同種写像 $\sigma_{0,i} : E_{\mathtt{A}} \to E_{1,i}$ を計算する。署名対象メッセージ $\mathtt{msg}$ に対してチャレンジ bit 列 $h := b_1 \| \cdots \| b_t := H(j(E_{1,1}), \ldots, j(E_{1,t}), \mathtt{msg}) \in \{0,1\}^t$ をハッシュ関数 H で計算する。各 $i = 1, \ldots, t$ に対して，もし $b_i = 1$ なら KLPT アルゴリズムに基づいて「ランダム化」したランダム同種写像 $\sigma_{1,i} : E_0 \to E_{1,i}$ を計算する。署名を $\sigma := (h, \sigma_{b_1,1}, \ldots, \sigma_{b_t,t})$ とする。

署名検証： 公開鍵 $(pp_{\mathtt{gps}}, E_{\mathtt{A}})$，メッセージ $\mathtt{msg}$ と署名 $\sigma = (h, \sigma_1, \ldots, \sigma_t)$ を入力として，各 $i = 1, \ldots, t$ に対して，同種写像 σ_i を計算して，その終点曲線 $E_{1,i}$ を得る。次に $H(j(E_{1,1}), \ldots, j(E_{1,t}), \mathtt{msg})$ を計算して署名内の h

[*6] S_k（$k = 1, 2$）が B-べき平滑数（powersmooth number）とは，S_k が $\ell_{k,i}^{e_{k,i}} < B$ なる $\ell_{k,i}^{e_{k,i}}$ の積で表される（i.e., $S_k = \prod_i \ell_{k,i}^{e_{k,i}}$）ことである。ただし $\ell_{k,i}$ は互いに異なるものとする。

と一致するかどうか検証して，全ての $i = 1, \ldots, t$ に対して検証が成功すれば受理を出力して，そうでなければ，棄却とする。

GPS 署名方式は，超特異楕円曲線同種写像計算問題またはそれと同値な自己準同型環計算問題（定義 6.2）の困難性を仮定すればランダムオラクルモデルの下で EUF-CMA 安全であることが示されている [27, 定理 10]。GPS 署名方式では，1 bit のチャレンジを用いた Σ-プロトコルに基づいているため，署名サイズが大きくなるのが欠点である。また，署名生成で使われた KLPT アルゴリズムの計算時間改善も課題であった [36, 5.1.2 節]。以上，GPS 署名方式には (1) 署名サイズ 及び (2) KLPT アルゴリズム計算時間 に関する 2 つの課題が存在する。

6.3 同種写像に基づく主要な暗号方式

本節では，公開鍵と署名サイズが小さいことを特長にもつ SQIsign 署名方式について述べる（表 6.2 参照）。

表 6.2: 同種写像に基づく暗号の分類

文献	暗号化	鍵交換	署名
SQIsign [23, 10, 3]			○

6.3.1 SQIsign 署名方式

以下，自己準同型環計算問題（定義 6.2）の困難性に安全性の根拠を置く SQIsign 署名方式を概説する。SQIsign 署名方式は公開鍵と署名を合わせたサイズが小さい方式として注目されている。また，2024 年 10 月に，SQIsign 署名方式が NIST PQC 標準化プロジェクト追加署名第 2 ラウンドに進むことが発表された [2]。以下では，KLPT アルゴリズムに基づいた SQIsign 署名方式 [23, 10] のアルゴリズムとパラメータを述べた後，最新の改良版である SQIsign2D 署名方式 [3] について報告する。

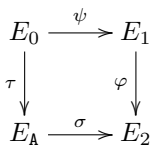
6.3.1.1 KLPT アルゴリズムに基づく SQIsign 署名方式

6.2.1 節で述べた GPS 署名方式を基にして改良を加えた署名方式が SQIsign 署名方式であり，ASIACRYPT 2020 で De Feo–Kohel–Leroux–Petit–Wesolowski [23] により提案された。6.2.1 節末尾に付した GPS 署名方式の 2 つの課題を克服している。チャレンジ空間に同種写像の空間を用いることで，そのサイズをセキュリティパラメータ λ まで大きくして，Σ-プロトコルを 1 度適用するだけで十分な Fiat–Shamir 署名構成とした。これで署名サイズが格段に小さくなった。また，GPS 署名生成においては，表 6.1 の Deuring 対応に基づいて，同種写像のイデアル表現（表 6.1 の四元数環側）をねじれ点を使った表現（表 6.1 の楕円曲線側）に変換する部分で時間が費やされていたが，SQIsign 署名方式ではその処理を速度改善したサブルーチン（IdealToIsogeny）に置き換えるのに成功して現実的な演算効率を達成した（詳細は [23, 24] を参照）。

また，安全性に関しては，健全性は超特異平滑自己準同型写像計算問題（定義 6.3）の困難性に基づき，零知識性は定義 6.4 で述べた SQIsign 署名方式のランダム識別問題の困難性に基づいている。初期提案 [23] では，ノルム方程式を解くサブルーチンに不備があり，生成される署名同種写像 σ に偏りが生じていたことが [24] において指摘された。そして，更に [24] でその不備を除去したアルゴリズム提案が行われた。

■SQIsign 署名アルゴリズム　SQIsign 署名方式では，右図の同種写像 τ が秘密鍵で，超特異楕円曲線 $E_{\mathtt{A}}$ が公開鍵（の主要な一部）である。署名生成では，コミットメント同種写像 ψ とチャレンジ同種写像 φ を適切に生成して得られた合成写像 $\varphi \circ \psi \circ \hat{\tau}$ を一般化された KLPT アルゴリズムに基づいてランダム化した同種写像 σ を署名（Σ-プロトコルのレスポンス）とする。一般化 KLPT アルゴリズムに関しては [10, 2.5.2.2 節] を参照。チャレンジ φ によりセキュリティパラメータ分のランダムネスを与えることができるので，1 度の Σ-プロトコル適用で十分な安全性が達成できる。よって，GPS 署名方式と比べて格段に短い署名サイズが実現できる。

SQIsign 同種写像図式



鍵生成： 既知の特殊極値的自己準同型環 $\mathcal{O}_0$ をもつ超特異楕円曲線 E_0，λ bits の平滑奇数 D_c（λ はセキュリティパラメータ），超特異 2-同種写像グラフの直径より大きな e による $D := 2^e$ を生成して，$pp_{\mathtt{sqisign}} := (E_0, D_c, D)$ を公開パラメータとする。さらに，E_0 を始点とするランダムな同種写像 $\tau : E_0 \to E_{\mathtt{A}}$ を計算して，$pp_{\mathtt{sqisign}}$ と $E_{\mathtt{A}}$ を公開鍵として，τ を秘密鍵とする。

署名生成： E_0 を始点とするランダムな同種写像 $\psi : E_0 \to E_1$ を計算。署名対象メッセージ $\mathtt{msg}$ に対してハッシュ関数 H で計算した $H(j(E_1), \mathtt{msg})$ から決まる D_c 次の巡回同種写像 $\varphi : E_1 \to E_2$ を計算。同種写像の合成 $\varphi \circ \psi \circ \hat{\tau} : E_{\mathtt{A}} \to E_2$ から（一般化された KLPT アルゴリズムを用いて）同じ始点・終点を有して $\hat{\varphi} \circ \sigma$ が巡回同種写像になる D 次のランダム同種写像 $\sigma : E_{\mathtt{A}} \to E_2$ を計算。(E_1, E_2, σ) を $\mathtt{msg}$ の署名として出力。

署名検証： 公開鍵 $(pp_{\mathtt{sqisign}}, E_{\mathtt{A}})$，メッセージ $\mathtt{msg}$ と署名 (E_1, E_2, σ) を入力として，E_1 から E_2 への同種写像 $\varphi := H(j(E_1), \mathtt{msg})$ を計算する。σ が $E_{\mathtt{A}}$ から E_2 への D 次同種写像であることと $\hat{\varphi} \circ \sigma$ が $E_{\mathtt{A}}$ から E_1 への巡回同種写像であることを検証して，共に成立すれば受理を出力して，そうでなければ，棄却とする。

既に述べたように，SQIsign 署名方式の安全性は，超特異平滑自己準同型写像計算問題（定義 6.3）の困難性と，定義 6.4 で述べた SQIsign 署名 σ のランダム識別問題の困難性に基づいている。また，Santos–Eriksen–Meyer–Reijnders [52] は有限拡大体を活用して署名検証を高速に行う方法を提案している。

■SQIsign 署名パラメータ　署名同種写像 σ の次数は $D = 2^e$，チャレンジ同種写像 φ の次数は平滑奇数 D_c である。$\mathbb{F}_p$ 上の超特異楕円曲線 E の位数 $p+1$ のねじれ点及びそのツイスト曲線上の位数 $p-1$ のねじれ点を利用して次数 D, D_c の同種写像を小さい拡大次数の有限体で効率的に計算するために，できるだけ大きい正整数 f，正奇数 T に関して $2^f \cdot T \,|\, p^2 - 1$ が満たされる素数 p（SQIsign 素数）を生成することが必要である。具体的には，ある B に対して B-平滑な T，$T \approx p^{5/4+\epsilon}$（[6] では例えば $0.02 < \epsilon < 0.1$ とする）に対して $2^f \cdot T \,|\, p^2 - 1$ となる素数 p を探索する必要がある。SQIsign 素数の選択基準として，署名検証の効率化には f をできるだけ大きくして，署名生成の効率性にとっては $\sqrt{B}/f$ をできるだけ小さくするのが望ましい [24]。

6.3.1.2　SQIsign2D 署名方式

Dartois ら [15] により高次元同種写像を用いて改善を図った SQIsignHD 署名方式が提案された。さらに 2 次元同種写像によってデータサイズ，演算時間，安全性に関して改善された複数の方式が相次いで発表されている [3, 44, 17, 7]。以下では，それらの中で，特に，SQIsign2D-West 署名方式 [3] に関して，[3] で述べられたパラメータ，データサイズ及び性能報告に関して述べる。SQIsign 署名方式を 2 次元同種写像を用いて改善することができたのは Nakagawa–Onuki [43, 44] の貢献が大きい。

特筆すべきは，素数 p の選択である。上に述べたように，従来の SQIsign 署名方式では B-平滑な T を適切に設定する必要があるなど素数 p の選択には限界が伴っていた。しかし，SQIsign2D-West では，できるだけ小さな c により $p+1 = c \times 2^e$ となる素数 p を用いるため，セキュリティレベルに応じて柔軟なパラメータ選択がしやすい。また，一般化メルセンヌ素数 $p = c \times 2^e - 1$ を用いることで高速実装も可能になり，表 6.3 に示すように，鍵生成・署名生成・署名検証において実用的な実行時間が達成できることが報告されている [3]。

そして，表 6.3 に示されているように，セキュリティパラメータ λ ($\sim \frac{1}{2}\log_2 p$) に対して公開鍵サイズを $4\lambda + 16$ bits，署名サイズを $9\lambda + 16 + 2\log_2(2\lambda)$ bits と小さく抑えることができるのも特長である。

表 6.3: SQIsign2D-West 素数パラメータ p 及び公開鍵・署名サイズ（Bytes），Intel Xeon Gold 6338（Ice Lake, 2GHz）上での鍵生成・署名生成・署名検証の実行時間（ms）[3]

NIST 安全性レベル	1	3	5
素数 p	$5 \cdot 2^{248} - 1$	$65 \cdot 2^{376} - 1$	$27 \cdot 2^{500} - 1$
公開鍵サイズ	66	98	130
署名サイズ	148	222	294
鍵生成	30	85	180
署名生成	80	230	470
署名検証	4.5	14.5	31

6.4 同種写像に基づく暗号技術に関するまとめ

本章では，同種写像に基づいた暗号技術をまとめた。NIST PQC 標準化プロジェクト追加署名第 2 ラウンドに進んだ SQIsign 署名方式についてまとめてきた。また，高次元同種写像の暗号応用に関しても調査結果を報告した。

[14] によると，Couveignes は，1997 年の École Normale Supérieure でのセミナーで既に同種写像に基づく暗号技術を提案しており，ほぼ同時期に Kohel [34] や Galbraith [26] も，同種写像問題に関する研究を始めていた。つまり，同種写像暗号技術の研究は既に 27 年の歴史をもつ。そして，最近になり，耐量子計算機暗号の必要性が高まることで，同種写像暗号技術は注目されて研究が進み，NIST PQC 標準化プロジェクト第 4 ラウンドにも選ばれた SIKE 暗号方式及びその基本形である SIDH 鍵共有は，最近まで堅調に安全性評価を積み重ねてきた。しかし，2022 年の Castryck–Decru の攻撃法 [8] を始めとする一連の攻撃法 [39, 47] は SIDH 鍵共有に対して決定的な結果をもたらした。

一方，本章においても随所に見られるように，Kani の補題に基づいて楕円曲線同種写像を高次元同種写像に埋め込むことで，平滑次数でない同種写像も暗号演算に取り込むことが可能になるなど，SIDH 攻撃法に端を発した全く新しい同種写像暗号研究が現在展開されつつある。例えば，SIDH 攻撃の発案者である Castryck は，“An Attack Became a Tool: Isogeny-based Cryptography 2.0” と題する EUROCRYPT 2024 の招待講演において，同種写像暗号研究が今新しい転換点に差し掛かっており，その技術的な核となるのが高次元同種写像の利用であると述べている。更に，調査報告書で示したレベル構造付き同種写像問題などの新たな安全性解析の枠組みに関しても研究が進んでおり，そのような理論的基盤に基づいて，新しい方式提案も含む活発な研究活動が引き続いて行われている。

現在，特に，公開鍵と署名を合わせたサイズが小さい SQIsign 署名方式が注目されていると共に，調査報告書で述べたように，CSIDH ベースの一方向性群作用に関する研究も注目されており，種々の暗号プロトコルへの応用も視野に入れた研究も進んでいる。それらも含めて，今後，特に注意すべきこと数点について以下にまとめておく。

- SQIsign 署名方式は，公開鍵と署名のサイズの小ささ，補助点なしの署名構成，そして短署名に対する強い社会的ニーズなどを踏まえると，現在有望な同種写像暗号技術と思われる。その一方，零知識性に関する計算問題（定義 6.4）の安全性検討などに関して，まだ安全性評価が不十分であり，その安全性評価は今後の重要な課題の一つである。さらに，今後は，実装研究を進める必要もあり，特にさまざまなプラットフォームでの実装結果を蓄えていく必要がある。また，SQIsign2D-West 論文 [3]（ASIACRYPT 2024）の副題は “The Fast, the Small, and the Safer” となっており，2 次元同種写像の利用により演算速度，データサイズ，安全性と多方面での改善が図られており，この方向性での今後の研究進展に注目していく必要がある。
- SQIsign 署名方式は NIST PQC 標準化プロジェクト追加署名第 2 ラウンドに進むことが決定しており [2]，SQIsign（及び SQIsign2D）署名パラメータに対して，（一般的な）超特異同種写像問題及びそれと同値な自己準同型環計算問題に対する古典・量子アルゴリズムの詳細な解析・見積もりを行うことが今後の重要な課題である。
- 鍵共有方式として，レベル構造付き同種写像問題に基づく M-SIDH 鍵共有，(Q)FESTA 鍵共有，binSIDH 鍵共有を調査報告書で取り上げた。群作用ベースの鍵共有には，例えば，CSIDH，SCALLOP，SiGamal などがあるが，他にも POKE，IS-CUBE，LIT-SiGamal など新たな鍵共有・暗号方式が提案されてきており，これからも同種写像に基づく鍵共有・暗号方式の安全性解析と方式改良（及び新規提案）は大変重要な課題である。
- 調査報告書で述べたリング署名・グループ署名の他にもパスワード認証鍵共有（PAKE）[1, 29] や紛失疑似ランダム関数（OPRF）[50] などといった一方向性群作用の暗号応用に関する研究が進められており，耐量子計算機性をもつ方式として注目する必要がある。更に，近年では量子マネーなどの新しい応用研究 [37, 58, 41, 42] も進んでおり，一方向性群作用の新たな暗号応用を探ることも今後の重要な課題の一つである。
- 上で述べたように高次元同種写像を利用した暗号・署名構成，及び安全性解析は，現在も研究が進展し続けている。全体に，同種写像暗号技術は，まだまだ研究の余地があり，鍵・暗号文・署名サイズの小ささの点で他の耐量子計算機暗号にない特長があるので，さまざまな利用用途を見据えて今後も継続的な研究が望まれる。

第 6 章の参照文献

[1] M. Abdalla, T. Eisenhofer, E. Kiltz, S. Kunzweiler, D. Riepel. Password-Authenticated Key Exchange from Group Actions. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 699–728.

[2] G. Alagic et al. Status Report on the First Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8528, https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8528.pdf. 2024-10.

[3] A. Basso, L. De Feo, P. Dartois, A. Leroux, L. Maino, G. Pope, D. Robert, B. Wesolowski. SQIsign2D-West - The Fast, the Small, and the Safer. 2024.

[4] B. Bencina, P. Kutas, S.-P. Merz, C. Petit, M. Stopar, C. Weitkämper. Improved Algorithms for Finding Fixed-Degree Isogenies Between Supersingular Elliptic Curves. CRYPTO (5). Vol. 14924. Lecture Notes in Computer Science. Springer, 2024, pp. 183–217.

[5] J.-F. Biasse, D. Jao, A. Sankar. A Quantum Algorithm for Computing Isogenies between Supersingular Elliptic Curves. INDOCRYPT. Vol. 8885. Lecture Notes in Computer Science. Springer, 2014, pp. 428–442.

[6] G. Bruno, M. Corte-Real Santos, C. Costello, J. Komada Eriksen, M. Meyer, M. Naehrig, B. Sterner. Cryptographic Smooth Neighbors. Cryptology ePrint Archive, Paper 2022/1439. 2022. https://eprint.iacr.org/2022/1439.

[7] W. Castryck, M. Chen, R. Invernizzi, G. Lorenzon, F. Vercauteren. Breaking and Repairing SQIsign2D-East. Cryptology ePrint Archive, Paper 2024/1453. 2024. https://eprint.iacr.org/2024/1453. to appear in the proceedings of ASIACRYPT 2024 merging with [44].

[8] W. Castryck, T. Decru. An Efficient Key Recovery Attack on SIDH. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 423–447.

[9] D. X. Charles, K. E. Lauter, E. Z. Goren. Cryptographic Hash Functions from Expander Graphs. J. Cryptol. Vol. 22, Num. 1 (2009), pp. 93–113.

[10] J. Chavez-Saab et al. SQIsign: Algorithm specifications and supporting documentation. submission to the NIST's PQC standardization. (2023).

[11] A. M. Childs, D. Jao, V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. J. Math. Cryptol. Vol. 8, Num. 1 (2014), pp. 1–29.

[12] C. Costello. The Case for SIKE: A Decade of the Supersingular Isogeny Problem. Cryptology ePrint Archive, Paper 2021/543. 2021. https://eprint.iacr.org/2021/543.

[13] C. Costello, P. Longa, M. Naehrig, J. Renes, F. Virdia. Improved Classical Cryptanalysis of SIKE in Practice. Public Key Cryptography (2). Vol. 12111. Lecture Notes in Computer Science. Springer, 2020, pp. 505–534.

[14] J.-M. Couveignes. Hard Homogeneous Spaces. Cryptology ePrint Archive, Paper 2006/291. 2006. https://eprint.iacr.org/2006/291.

[15] P. Dartois, A. Leroux, D. Robert, B. Wesolowski. SQIsignHD: New Dimensions in Cryptography. EUROCRYPT (1). Vol. 14651. Lecture Notes in Computer Science. Springer, 2024, pp. 3–32.

[16] C. Delfs, S. D. Galbraith. Computing isogenies between supersingular elliptic curves over F_p. Des. Codes Cryptogr. Vol. 78, Num. 2 (2016), pp. 425–440.

[17] M. Duparc, T. B. Fouotsa. SQIPrime: A Dimension 2 Variant of SQISignHD with Non-smooth Challenge Isogenies. 2024.

[18] K. Eisenträger, S. Hallgren, K. E. Lauter, T. Morrison, C. Petit. Supersingular Isogeny Graphs and Endomorphism Rings: Reductions and Solutions. EUROCRYPT (3). Vol. 10822. Lecture Notes in Computer Science. Springer, 2018, pp. 329–368.

[19] K. Eisenträger, S. Hallgren, C. Leonardi, T. Morrison, J. Park. Computing endomorphism rings of supersingular elliptic curves and connections to pathfinding in isogeny graphs. ANTS 2020. Vol. 4. The Open Book Series 1. Mathematical Sciences Publishers, 2020, pp. 215–232.

[20] J. Komada Eriksen, L. Panny, J. Sotáková, M. Veroni. Deuring for the People: Supersingular Elliptic Curves with Prescribed Endomorphism Ring in General Characteristic. LuCaNT: LMFDB, Computation, and Number Theory. Vol. 796. Contemporary Mathematics. AMS, 2024. https://www.ams.org/books/conm/796/16008/conm796-16008.pdf.

[21] L. De Feo. Mathematics of Isogeny Based Cryptography. 2017. arXiv: 1711.04062.

[22] L. De Feo, J. Kieffer, B. Smith. Towards Practical Key Exchange from Ordinary Isogeny Graphs. ASIACRYPT (3). Vol. 11274. Lecture Notes in Computer Science. Springer, 2018, pp. 365–394.

[23] L. De Feo, D. Kohel, A. Leroux, C. Petit, B. Wesolowski. SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies. ASIACRYPT (1). Vol. 12491. Lecture Notes in Computer Science. Springer, 2020, pp. 64–93.

[24] L. De Feo, A. Leroux, P. Longa, B. Wesolowski. New Algorithms for the Deuring Correspondence – Towards Practical and Secure SQISign Signatures. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 659–690.

[25] J. Fuselier, A. Iezzi, M. Kozek, T. Morrison, C. Namoijam. Computing supersingular endomorphism rings using inseparable endomorphisms. 2023. arXiv: 2306.03051.

[26] S. D. Galbraith. Constructing Isogenies between Elliptic Curves Over Finite Fields. LMS Journal of Computation and Mathematics. Vol. 2 (1999), pp. 118–138.

[27] S. D. Galbraith, C. Petit, J. Silva. Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems. J. Cryptol. Vol. 33, Num. 1 (2020), pp. 130–175.

[28] S. D. Galbraith, F. Vercauteren. Computational problems in supersingular elliptic curve isogenies. Quantum Inf. Process. Vol. 17, Num. 10 (2018), p. 265.

[29] R. Ishibashi, K. Yoneyama. Compact Password Authenticated Key Exchange from Group Actions. ACISP. Vol. 13915. Lecture Notes in Computer Science. Springer, 2023, pp. 220–247.

[30] D. Jao et al. Supersingular Isogeny Key Encapsulation. https://sike.org/files/SIDH-spec.pdf. 2022-09. (2024-11-12 閲覧).

[31] S. Jaques, J. M. Schanck. Quantum Cryptanalysis in the RAM Model: Claw-Finding Attacks on SIKE. CRYPTO (1). Vol. 11692. Lecture Notes in Computer Science. Springer, 2019, pp. 32–61.

[32] Y. Kambe, A. Katayama, Y. Aikawa, Y. Ishihara, M. Yasuda, K. Yokoyama. Computing Endomorphism Rings of Supersingular Elliptic Curves by Finding Cycles in Concatenated Supersingular Isogeny Graphs. Commentarii Mathematici Universitatis Sancti Pauli. Vol. 72, Num. 1 (2024), pp. 19–42.

[33] Y. Kambe, Y. Takahashi, M. Yasuda, K. Yokoyama. On the feasibility of computing constructive Deuring correspondence. NuTMiC 2021. Vol. 126. Banach Center Publications. Institute of Mathematics, Polish Academy od Sciences, 2023. https://www.impan.pl/en/publishing-house/banach-center-publications/all/126/0/115356/on-the-feasibility-of-computing-constructive-deuring-correspondence.

[34] D. Kohel. Endomorphism rings of elliptic curves over finite fields. PhD thesis. University of California at Berkeley, 1996.

[35] D. Kohel, K. Lauter, C. Petit, J.-P. Tignol. On the quaternion ℓ-isogeny path problem. LMS Journal of Computation and Mathematics. Vol. 17 (2014), pp. 418–432. Special Issue A: Algorithmic Number Theory Symposium XI.

[36] A. Leroux. Quaternion algebras and isogeny-based cryptography. PhD thesis. Ecole Polytechnique, 2022.

[37] J. Liu, H. Montgomery, M. Zhandry. Another Round of Breaking and Making Quantum Money: How to Not Build It from Lattices, and More. EUROCRYPT (1). Vol. 14004. Lecture Notes in Computer Science. Springer, 2023, pp. 611–638.

[38] P. Longa, W. Wang, J. Szefer. The Cost to Break SIKE: A Comparative Hardware-Based Analysis with AES and SHA-3. CRYPTO (3). Vol. 12827. Lecture Notes in Computer Science. Springer, 2021, pp. 402–431.

[39] L. Maino, C. Martindale, L. Panny, G. Pope, B. Wesolowski. A Direct Key Recovery Attack on SIDH. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 448–471.

[40] A. Herlédan Le Merdy, B. Wesolowski. The supersingular endomorphism ring problem given one endomorphism. Cryptology ePrint Archive, Paper 2023/1448. 2023. https://eprint.iacr.org/2023/1448.

[41] H. Montgomery, S. Sharif. Quantum Money from Class Group Actions on Elliptic Curves. 2024.

[42] S. Mutreja, M. Zhandry. Quantum State Group Actions. Cryptology ePrint Archive, Paper 2024/1636. 2024. https://eprint.iacr.org/2024/1636.

[43] K. Nakagawa, H. Onuki. QFESTA: Efficient Algorithms and Parameters for FESTA Using Quaternion Algebras. CRYPTO (5). Vol. 14924. Lecture Notes in Computer Science. Springer, 2024, pp. 75–106.

[44] K. Nakagawa, H. Onuki. SQIsign2D-East: A New Signature Scheme Using 2-dimensional Isogenies. Cryptology ePrint Archive, Paper 2024/771. 2024. https://eprint.iacr.org/2024/771. to appear in the proceedings of ASIACRYPT 2024 merging with [7].

[45] A. Page, B. Wesolowski. The Supersingular Endomorphism Ring and One Endomorphism Problems are Equivalent. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 388–417.

[46] J. Renes. Computing Isogenies Between Montgomery Curves Using the Action of $(0,0)$. PQCrypto. Vol. 10786. Lecture Notes in Computer Science. Springer, 2018, pp. 229–247.

[47] D. Robert. Breaking SIDH in Polynomial Time. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 472–503.

[48] D. Robert. On the efficient representation of isogenies (a survey). Cryptology ePrint Archive, Paper 2024/1071. 2024. https://eprint.iacr.org/2024/1071.

[49] A. Rostovtsev, A. Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Paper 2006/145. 2006. https://eprint.iacr.org/2006/145.

[50] C. D. de Saint Guilhem, R. Pedersen. New Proof Systems and an OPRF from CSIDH. Public Key Cryptography (3). Vol. 14603. Lecture Notes in Computer Science. Springer, 2024, pp. 217–251.

[51] M. Corte-Real Santos, C. Costello, J. Shi. Accelerating the Delfs-Galbraith Algorithm with Fast Subfield Root Detection. CRYPTO (3). Vol. 13509. Lecture Notes in Computer Science. Springer, 2022, pp. 285–314.

[52] M. Corte-Real Santos, J. Komada Eriksen, M. Meyer, K. Reijnders. AprèsSQI: Extra Fast Verification for SQIsign Using Extension-Field Signing. EUROCRYPT (1). Vol. 14651. Lecture Notes in Computer Science. Springer, 2024, pp. 63–93.

[53] A. Udovenko, G. Vitto. Revisiting Meet-in-the-Middle Cryptanalysis of SIDH/SIKE with Application to the $IKEp182 Challenge. Cryptology ePrint Archive, Paper 2021/1421. 2021. https://eprint.iacr.org/2021/1421.

[54] J. Vélu. Isogénies entre courbes elliptiques. C. R. Acad. Sci. Paris, Sér. A. Vol. 273 (1971), pp. 305–347.

[55] J. Voight. Quaternion algebras. Springer International Publishing, 2021-06.

[56] L. C. Washington. Elliptic curves : number theory and cryptography. 2nd ed. Discrete mathematics and its applications. Chapman & Hall/CRC, 2008.

[57] B. Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. FOCS. IEEE, 2021, pp. 1100–1111.

[58] M. Zhandry. Quantum Money from Abelian Group Actions. ITCS. Vol. 287. LIPIcs. 2024, 101:1–101:23.

[59] 相川 勇輔, 神戸 祐太, 工藤 桃成, 高島 克幸, 安田 雅哉. 代数曲線の計算理論と暗号への応用. 数学メモアール 10. 日本数学会, 2024.

第 7 章

ハッシュ関数に基づく署名技術

本章ではハッシュ関数に基づく署名技術についてまとめる。ハッシュ関数に基づく署名技術の安全性はハッシュ関数の第二原像攻撃に対する安全性に依存している。

ハッシュ関数に基づく署名技術は，最初に Lamport により one-time signature として提案された [15, 26]。また，この方式を改良した Winternitz one-time signature が Merkle [30] により述べられている。これらの方式は一組の公開鍵と秘密鍵を用いて一つのメッセージに署名を行う 1 回署名方式である。1 回署名方式とマークル木とを用いて複数回署名を行うことを可能とする方式が Merkle [29, 30] により述べられている。

7.1 ハッシュ関数に基づく署名技術の安全性の根拠となる問題

ハッシュ関数は任意長あるいは実用上十分な長さ以下の入力 $\{0,1\}$ 系列に対して固定長の $\{0,1\}$ 系列を出力する関数である。ハッシュ関数を $H : \mathcal{D} \to \mathcal{R}$ とする。ここで，$\mathcal{D}$ は任意長の $\{0,1\}$ 系列の集合 $\{0,1\}^*$ の部分集合であり，$\mathcal{R}$ は固定長の $\{0,1\}$ 系列の集合である。ハッシュ関数の第二原像攻撃は，第一原像 $X \in \mathcal{D}$ が与えられたとき，$X \neq X'$ かつ $H(X) = H(X')$ を満たす第二原像 $X' \in \mathcal{D}$ を求めるという問題を解くことを目的とする攻撃である。なお，第二原像攻撃に対する安全性は，しばしば，ハッシュ関数が各入力に対する出力を無作為に選択するランダム関数であると仮定して評価される。このようなランダム関数はランダムオラクルとも呼ばれる。H がランダムオラクルであるとき，第二原像の計算時間は $\Theta(|\mathcal{R}|)$ である。また，量子コンピュータでは，Grover の探索アルゴリズム [18] を用いることにより，第二原像の計算時間は $\Theta(\sqrt{|\mathcal{R}|})$ となる。

本章で取り上げるハッシュ関数に基づく署名技術では, 米国 NIST の指定する標準ハッシュ関数族である SHA-2 [33], SHA-3 [34] のうちのいくつかのハッシュ関数を用いることが想定されている。

SHA-2 は固定長入出力の圧縮関数からなる Merkle-Damgård 構造 [14, 31] を有するハッシュ関数の族であり, Secure Hash Standard [33] のうち，SHA-1 を除く SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 からなる。SHA-2 の各ハッシュ関数の名称の末尾の数値は出力の bit 長を表す。SHA-3 は固定長入出力の置換を用いたスポンジ構造 [9] を有するハッシュ関数の族であり，SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256 からなる。SHA3-224, SHA3-256, SHA3-384, SHA3-512 については，末尾の数値は出力の bit 長を表す。SHAKE128, SHAKE256 については，出力長は任意に設定できる。

本章で使用する記号・用語を以下にまとめる。

- $\{0,1\}$ 系列 α, β の連接を $\alpha \| \beta$ と表記する。
- $\ll$ は左論理シフトを表す。

- 整数 ν について $[\nu]_l$ は ν の長さ l Bytes の 2 進数表記を表す。
- $\mathbb{B} := \{0,1\}^8$ とする。
- 8080 のように typewriter font で書かれている数字は 16 進数として解釈する。

7.2 ハッシュ関数に基づく代表的な署名方式

7.2.1 Winternitz One-Time Signature

Winternitz one-time signature [30] は，一組の公開鍵と秘密鍵を用いて一つのメッセージに署名を行う 1 回署名方式である。この方式では，署名対象のメッセージのハッシュ値 N を b 進数表記の整数とみなす。N が ℓ_{m} 桁の b 進数 $N_{\ell_{\mathrm{m}}-1}N_{\ell_{\mathrm{m}}-2}\cdots N_1N_0$ で表記されるとする。このとき，$0 \leq k \leq \ell_{\mathrm{m}}-1$ について $N_k \in \{0,1,\ldots,b-1\}$ であり，$N = \sum_{k=0}^{\ell_{\mathrm{m}}-1} N_k 2^k$ である。さらに，N のチェックサムを $C := \sum_{k=0}^{\ell_{\mathrm{m}}-1} (b-1-N_k)$ と定義する。C が ℓ_{c} 桁の b 進数 $N_{\ell_{\mathrm{m}}+\ell_{\mathrm{c}}-1}N_{\ell_{\mathrm{m}}+\ell_{\mathrm{c}}-2}\cdots N_{\ell_{\mathrm{m}}+1}N_{\ell_{\mathrm{m}}}$ で表記されるとする。$\ell := \ell_{\mathrm{m}} + \ell_{\mathrm{c}}$ とする。

■鍵生成アルゴリズム　秘密鍵 $(x_0, x_1, \ldots, x_{\ell-1})$，公開鍵 $(pub_0, pub_1, \ldots, pub_{\ell-1})$ は以下のように生成される。

1. $x_0, x_1, \ldots, x_{\ell-1} \in \mathcal{D}$ を無作為に選択する。
2. $0 \leq k \leq \ell-1$ について $pub_k := H^{b-1}(x_k) := \underbrace{H(H(\cdots(H}_{b-1 \text{ times}}(x_k))\cdots))$ とする。

■署名アルゴリズム　メッセージのハッシュ値 N の署名 $(s_0, s_1, \ldots, s_{\ell-1})$ は以下のように生成される。

1. $0 \leq k \leq \ell-1$ について $s_k := H^{N_k}(x_k)$ とする。

■検証アルゴリズム　メッセージのハッシュ値 N とその署名 $(s_0, s_1, \ldots, s_{\ell-1})$ の検証は以下のように行われる。

1. $0 \leq k \leq \ell-1$ について $pub_k = H^{b-1-N_k}(s_k)$ かつそのときに限り，$(s_0, s_1, \ldots, s_{\ell-1})$ は N の正しい署名である。

仮にチェックサムが導入されていないとすると，N の署名 $(s_0, s_1, \ldots, s_{\ell_{\mathrm{m}}-1})$ が得られたとき，$0 \leq k \leq \ell_{\mathrm{m}}-1$ について $N'_k \geq N_k$ を満たす N' について，$s'_k := H^{N'_k-N_k}(s_k)$ によって，署名 $(s'_0, s'_1, \ldots, s'_{\ell_{\mathrm{m}}-1})$ が容易に偽造できる。

Winternitz one-time signature の偽造不能性は，Dods ら [16] により論じられている。Winternitz one-time signature に基づく方式については，Lafrance と Menezes [25] によりまとめられている。

7.2.2 マークル木を用いた署名方式

1 回署名方式を用いて複数のメッセージに署名を行う場合，メッセージの個数と同じ個数の公開鍵と秘密鍵の組が必要となる。マークル木を用いることにより，このような複数回署名方式の公開鍵の大きさを削減できる [29]。

2^h 個のメッセージに署名を行うための 1 回署名の公開鍵を $pk_0, pk_1, \ldots, pk_{2^h-1}$ とする。このとき，高さが h，すなわち，葉の個数が 2^h のマークル木は以下のように構成される。高さ $j(\geq 0)$ の左から $i(\geq 0)$ 番目の節点を $v_{i,j}$ と表記する。$v_{i,j}$ は以下のように計算される。

1. $0 \leq i \leq 2^h-1$ について，$v_{i,0} := H(pk_i)$ とする。

2. $1 \leq j \leq h$ に対し，$0 \leq i \leq 2^{h-j} - 1$ について，$v_{i,j} := H(v_{2i,j-1} \| v_{2i+1,j-1})$ とする。

この署名方式の公開鍵は $v_{0,h}$ である。秘密鍵は 1 回署名の公開鍵 $pk_0, pk_1, \ldots, pk_{2^h-1}$ に対応するすべての秘密鍵である。i 個目のメッセージの署名を検証するためには，$v_{0,h}$ を用いて pk_i が正しいことを検証する必要がある。このために，i 個目のメッセージの署名には，マークル木の $v_{i,0}$ から $v_{0,h}$ に至る経路上の各節点の，経路上にない子節点が含まれる。これらの節点の列は認証パスと呼ばれる。

7.2.3 マークル木の階層構造による署名方式

前節で述べた一つのマークル木を用いた署名方式では，鍵生成時にすべての 1 回署名の公開鍵と秘密鍵を生成する必要があり，例えば，2^{50} 個の署名を行うために高さ 50 のマークル木を構成することは，所要計算時間の観点から非実用的である。このような多数のメッセージに署名を行う際には，マークル木を用いた署名方式の階層構造による署名方式が提案されている [22]。

この署名方式で構成されるマークル木を用いた署名方式の階層構造の階層数を L とし，根に相当する最上層を第 $(L-1)$ 層，葉に相当する最下層を第 0 層とする。さらに，$0 \leq i \leq L-1$ について，第 i 層のマークル木の高さはすべて等しく h_i であると仮定する。このとき，第 i 層のマークル木は $2^{\sum_{j=i+1}^{L-1} h_j}$ 個存在する。この署名方式では $2^{\sum_{j=0}^{L-1} h_j}$ 個のメッセージに署名できる。

この署名方式では，第 $(L-1)$ 層のマークル木の根が公開鍵となる。この公開鍵を生成する際には，1 回署名の公開鍵と秘密鍵の組を $2^{h_{L-1}}$ 個だけ生成すれば良い。$0 < i \leq L-1$ について，第 i 層の各マークル木は第 $(i-1)$ 層の 2^{h_i} 個のマークル木の根を署名するために使用される。第 0 層のマークル木は，それぞれ 2^{h_0} 個のメッセージの署名に使用される。

この署名方式では，一つのメッセージの署名の際に，各層についてそれぞれ一つのマークル木を生成しておけば十分である。各メッセージの署名は，そのメッセージに対する第 0 層のマークル木による署名と，$0 < i \leq L-1$ について，そのメッセージの署名の際に使用された第 i 層のマークル木による第 $(i-1)$ 層のマークル木の根の署名からなる。この署名方式について，階層数 $L = 3$，各階層のマークル木の高さ $h_0 = h_1 = h_2 = 3$ の模式図を図 7.1 に示す。灰色の節点は認証パスをなす節点である。

7.2.4 プレフィクスとビットマスク

プレフィクスは，ハッシュ関数に基づく署名方式の処理において，すべてのハッシュ関数の計算がそれぞれ異なる入力に対して行われるよう入力に付加される系列である。プレフィクスは，Lighton と Micali [27] により，security string という名称で，ハッシュ関数に基づく署名方式の安全性をハッシュ関数の第二原像攻撃に対する安全性にタイトに帰着するために導入された。なお，プレフィクスは，ハッシュ関数の用途とそれが用いられる位置（例えば，どの 1 回署名方式か，どのマークル木のどの節点か）により自然に定義できることから，現在は通常，アドレスと呼ばれる。

ビットマスクは，Dahmen ら [13] により，ハッシュ関数に基づく署名方式の安全性をハッシュ関数の第二原像攻撃に対する安全性に帰着するために導入された。ビットマスクは乱数系列であり，ハッシュ関数への入力をランダム化するために，bit ごとの排他的論理和により入力に加えられる。

7.3 ハッシュ関数に基づく主要な署名方式

本章で取り上げるハッシュ関数に基づく署名方式を表 7.1 に示す。

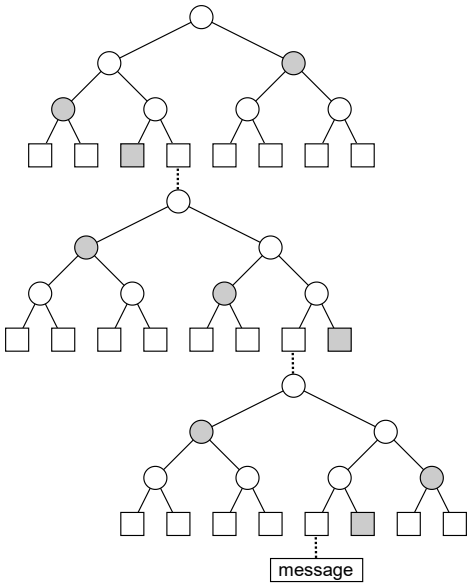


図 7.1: マークル木の階層構造による署名方式

表 7.1: ハッシュ関数に基づく署名方式

文献	暗号化	鍵交換	署　名
eXtended Merkle Signature Scheme (XMSS) [19, 12]			○
Stateless Hash-Based Digital Signature Algorithm (SLH-DSA) [35]			○

NIST SP 800-208 [12] は，以下のハッシュ関数に基づく stateful な署名方式を規定している。

- Lighton-Micali Signatures (LMS), Hierarchical Signature System (HSS) [28]
- eXtended Merkle Signature Scheme (XMSS), multi-tree XMSS (XMSS^{MT}) [19]

LMS は Lighton と Micali による署名方式 [27] に基づく。HSS, XMSS^{MT} はそれぞれ，7.2.3 節で述べられたような，LMS, XMSS の階層構造による署名方式である。ハッシュ関数に基づく stateful な署名方式では，同一の秘密鍵が複数のメッセージの署名に使用されることがないように秘密鍵を管理することが必須である。LMS と HSS は調査報告書で取り上げられている。

NIST FIPS 205 [35] はハッシュ関数に基づく stateless な署名方式 SLH-DSA (Stateless Hash-Based Digital Signature Algorithm) を規定している。stateless な署名方式では，stateful な方式に求められるような秘密鍵の管理は不要である。SLH-DSA は，2022 年 7 月に NIST PQC 標準化プロジェクトで標準化候補アルゴリズムの一つに選出された SPHINCS$^+$ v.3.1 [2] に基づく。SPHINCS$^+$ は SPHINCS [8] の改良版として提案され [6, 7]，その後も NIST PQC 標準化プロジェクトで改良が行われ，v.3.1 となった。

以下では，XMSS，SLH-DSA についてそれぞれ 7.3.1 節，7.3.2 節で述べられるが，どちらを先に読んでも差し支えない。

7.3.1 XMSS: eXtended Merkle Signature Scheme

XMSS は [10, 22] で提案された方式の改良版 [23] に基づく署名方式であり，WOTS$^+$ と呼ばれる Winternitz one-time signature に基づく 1 回署名方式 [20] を用いる[*1]。

XMSS では三つの鍵付きハッシュ関数 $\mathsf{F}, \mathsf{H}, \mathsf{H}_{\mathrm{msg}}$ と擬似ランダム関数 R が用いられる。いずれも出力の byte 長は等しく，これを n とする。F の入力は byte 長 n の鍵と byte 長 n の系列である。H の入力は byte 長 n の鍵と byte 長 $2n$ の系列である。$\mathsf{H}_{\mathrm{msg}}$ の入力は byte 長 $3n$ の鍵と任意 byte 長の系列である。R の入力は byte 長 n の鍵と byte 長 32 の系列である。これらの関数は SHA-2 [33] または SHA-3 [34] を用いて定義される。例えば，$n = 32$ のとき，SHA-256 を用いて以下のように定義される。

$$\begin{aligned}
\mathsf{F}(k, x) &:= \text{SHA-256}([0]_{32} \| k \| x) \\
\mathsf{H}(k, x) &:= \text{SHA-256}([1]_{32} \| k \| x) \\
\mathsf{H}_{\mathrm{msg}}(k, x) &:= \text{SHA-256}([2]_{32} \| k \| x) \\
\mathsf{R}(k, x) &:= \text{SHA-256}([3]_{32} \| k \| x)
\end{aligned}$$

XMSS では，ハッシュ関数の呼び出しをランダム化するために，それぞれのハッシュ関数の呼び出しで，鍵とビットマスクが用いられる。これらは擬似ランダム関数を用いて生成され，入力として byte 系列の seed と長さ 32 Bytes のアドレス ADRS が与えられる。アドレスは 3 種あり，それぞれ OTS ハッシュアドレス，L 木アドレス，ハッシュ木アドレスと呼ばれる。それらの構造を図 7.2 に示す。

layer address	(4 Bytes)
tree address	(8 Bytes)
type = 0	(4 Bytes)
OTS address	(4 Bytes)
chain address	(4 Bytes)
hash address	(4 Bytes)
keyAndMask	(4 Bytes)

(a) OTS ハッシュアドレス

layer address	(4 Bytes)
tree address	(8 Bytes)
type = 1	(4 Bytes)
L-tree address	(4 Bytes)
tree height	(4 Bytes)
tree index	(4 Bytes)
keyAndMask	(4 Bytes)

(b) L 木アドレス

layer address	(4 Bytes)
tree address	(8 Bytes)
type = 2	(4 Bytes)
Padding = 0	(4 Bytes)
tree height	(4 Bytes)
tree index	(4 Bytes)
keyAndMask	(4 Bytes)

(c) ハッシュ木アドレス

図 7.2: アドレスの構造

7.3.1.1 WOTS$^+$

$w \in \{4, 16\}$ は Winternitz パラメータと呼ばれる。$\ell := \ell_1 + \ell_2$ は公開鍵，秘密鍵，署名を構成する byte 長 n の要素の個数を表す。ここで，

$$\ell_1 := \lceil 8n / \log_2 w \rceil, \quad \ell_2 := \lfloor \log_2(\ell_1(w-1)) / \log_2 w \rfloor + 1$$

である。

[*1] 7.3.2 節の SLH-DSA で用いられる 1 回署名方式とマークル木を用いた署名方式もそれぞれ WOTS$^+$，XMSS と呼ばれるが，アルゴリズムには相違点が存在する。

■チェイニング関数　チェイニング関数 chain の入力は，長さ n Bytes の系列 X，スタートインデクス i，ステップ数 s，長さ 32 Bytes のアドレス ADRS，長さ n Bytes のシード seed であり，以下のように定義される。

$$\mathsf{chain}(X, i, s, \mathsf{seed}, \mathrm{ADRS}) := \begin{cases} X & s = 0 \text{ のとき} \\ \mathtt{NULL} & i + s \geq w \text{ のとき} \\ \mathsf{F}(Key, \mathsf{chain}(X, i, s-1, \mathsf{seed}, \mathrm{ADRS}) \oplus BM) & \text{それ以外のとき} \end{cases}$$

ここで，

$$Key := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [i+s-1]_4 \| [0]_4), \quad BM := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [i+s-1]_4 \| [1]_4)$$

である。なお，ADRS′ は ADRS の上位 24 Bytes であり，例えば，$\mathrm{ADRS}' \| [i+s-1]_4 \| [0]_4$ は図 7.2a の ADRS の hash address, keyAndMask の値をそれぞれ，$[i+s-1]_4, [0]_4$ とすることを表している。

■鍵生成アルゴリズム　入力は ADRS, seed である。

1. $0 \leq i \leq \ell - 1$ について，$sk_i \in \{0,1\}^{8n}$ を無作為に選択する。
2. $0 \leq i \leq \ell - 1$ について，ADRS の chain address の値を $[i]_4$ とし，

   $$pk_i := \mathsf{chain}(sk_i, 0, w-1, \mathsf{seed}, \mathrm{ADRS})$$

   とする。この計算を図 7.3 に示す。この図で

   $$Key_j := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [j]_4 \| [0]_4), \quad BM_j := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [j]_4 \| [1]_4)$$

   である。

公開鍵は $pk := (pk_0, pk_1, \ldots, pk_{\ell-1})$ である。秘密鍵は $sk := (sk_0, sk_1, \ldots, sk_{\ell-1})$ である。

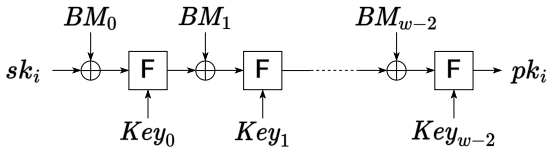


図 7.3: pk_i の計算

■署名アルゴリズム　入力は byte 長 n のメッセージ M，秘密鍵 sk，アドレス ADRS，シード seed である。

1. M をそれぞれ長さ $\log_2 w$ bits の ℓ_1 個のブロックに分割し，先頭から順に $M_0, M_1, \ldots, M_{\ell_1-1}$ とする。これらを整数とみなすと，$0 \leq i \leq \ell_1 - 1$ について，$M_i \in \{0, 1, \ldots, w-1\}$ である。
2. $C := \sum_{i=0}^{\ell_1-1} (w - 1 - M_i)$ とする。
3. $C \cdot 2^{8-(\ell_2 \log_2 w \bmod 8)}$ を長さ $\lceil (\ell_2 \log_2 w)/8 \rceil$ Bytes の系列とみなし，それぞれ長さ $\log_2 w$ bits の ℓ_2 個のブロックに分割し，先頭から順に $M_{\ell_1}, M_{\ell_1+1}, \ldots, M_{\ell-1}$ とする。
4. $0 \leq i \leq \ell - 1$ について，ADRS の chain address の値を i とし，

   $$sig_i := \mathsf{chain}(sk_i, 0, M_i, \mathsf{seed}, \mathrm{ADRS})$$

   とする。

メッセージ M に対する署名は $sig_0, sig_1, \ldots, sig_{\ell-1}$ である。

■検証アルゴリズム　鍵生成と署名のアルゴリズムより容易に導出される。詳細は [19] を参照のこと。

7.3.1.2　XMSS

XMSS はマークル木を用いた署名方式であり，公開鍵と秘密鍵の各組は完全二分木に対応付けられる。

XMSS のハッシュ木の構成のために，ランダム化ハッシュ関数 RH が導入されている。RH の入力は長さ n Bytes の $LEFT, RIGHT$，長さ n Bytes のシード seed，長さ 32 Bytes のアドレス ADRS であり，以下のように定義される。

$$\mathsf{RH}(LEFT, RIGHT, \mathsf{seed}, \mathrm{ADRS}) := \mathsf{H}(Key, (LEFT \oplus BM_0) \| (RIGHT \oplus BM_1))$$

ここで，

$$Key := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [0]_4), \quad BM_0 := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [1]_4), \quad BM_1 := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [2]_4)$$

である。なお，ADRS$'$ は ADRS の上位 28 Bytes であり，例えば，ADRS$' \| [0]_4$ は ADRS の図 7.2 の keyAndMask の値を $[0]_4$ とすることを表している。

秘密鍵の生成には [10] に示されているような擬似ランダム鍵生成法を用いることが許容されているが，その安全性は少なくとも XMSS の安全性と同等でなければならない。

■鍵生成アルゴリズム　鍵生成アルゴリズムではマークル木が構成され，その各葉には WOTS$^+$ の公開鍵が対応する。WOTS$^+$ の公開鍵に対して L 木と呼ばれるハッシュ木が構成され，その木の根のハッシュ値が XMSS のマークル木の葉に割り当てられる。L 木の高さ $j(\geq 0)$ の左から $i(\geq 0)$ 番目の節点を $\mathrm{Node}_{i,j}$ と表記する。L 木は以下にしたがって構成される。入力は WOTS$^+$ の公開鍵 $pk := (pk_0, pk_1, \ldots, pk_{\ell-1})$，L 木アドレス ADRS，シード seed である。

1. $0 \leq i \leq \ell - 1$ について，$\mathrm{Node}_{i,0} := pk_i$ とする。
2. $j \geq 0$ について，根が得られるまで以下にしたがって $\mathrm{Node}_{i,j+1}$ を計算する。なお，値の定義された $\mathrm{Node}_{i,j}$ の個数を ℓ' とする。
   (a) $0 \leq i < \lfloor \ell'/2 \rfloor$ について，$\mathrm{Node}_{i,j+1} := \mathsf{RH}(\mathrm{Node}_{2i,j}, \mathrm{Node}_{2i+1,j}, \mathsf{seed}, \mathrm{ADRS})$ とする。ここで，ADRS の tree height を $[j]_4$，tree index を $[i]_4$ とする。さらに，ℓ' が奇数のとき，$\mathrm{Node}_{\lfloor \ell'/2 \rfloor, j+1} := \mathrm{Node}_{\ell'-1,j}$ とする。
   (b) $j \leftarrow j + 1$ とする。

鍵生成アルゴリズムで構成されるマークル木の高さを h とすると，このマークル木には 2^h 個の葉が存在する。このマークル木に対応する 2^h 個の WOTS$^+$ の公開鍵，それらの L 木，さらに，このマークル木の計算に用いられる OTS ハッシュアドレス，L 木アドレス，ハッシュ木アドレスの layer address，tree address はすべて，それぞれ $[0]_4$，$[0]_8$ である。左から $k\,(\geq 0)$ 番目の葉に対応する OTS ハッシュアドレスの OTS address，L 木アドレスの L-tree address は $[k]_4$ である。

鍵生成アルゴリズムで構成されるマークル木の葉は対応する L 木の根である。葉以外の節点は L 木の節点と同じ方法で計算される。なお，このマークル木は完全二分木なので，上述の L 木の計算手続きで，ℓ' は常に偶数となる。

秘密鍵は，2^h 個の WOTS$^+$ の秘密鍵，次の署名に使用される WOTS$^+$ の秘密鍵に対応するマークル木の葉の番号 idx，署名されるメッセージのハッシュの計算に使用される SK_{PRF}，マークル木の根 $root$，seed である。公開鍵は，マークル木の根，seed である。ここで，SK_{PRF} と seed はこの鍵生成アルゴリズムで無作為に選択される長さ n Bytes の系列である。また，公開鍵には識別子 OID が付される。

■署名アルゴリズム　メッセージ M の署名は，署名に使用される WOTS$^+$ の秘密鍵の番号 idx，M のダイジェストの計算に使用される乱数 r，WOTS$^+$ による署名，マークル木の idx 番目の葉の認証パスからなる。

1. M のダイジェストを $M' := \mathsf{H}_{\mathrm{msg}}(r \| root \| [idx]_n, M)$ とする。ここで，$r := \mathsf{R}(SK_{\mathrm{PRF}}, [idx]_4)$ である。
2. WOTS$^+$ の idx 番目の秘密鍵を用いて M' に署名し，マークル木の idx 番目の葉の認証パスを計算する。

WOTS$^+$ の同じ秘密鍵が 2 回以上使用されないよう，idx は $idx \leftarrow idx + 1$ により更新される。

■検証アルゴリズム　鍵生成と署名のアルゴリズムより容易に導出される。詳細は [19] を参照のこと。

7.3.1.3　XMSS^{MT}

XMSS^{MT} は，7.2.3 節のマークル木の階層構造による署名方式に相当する。XMSS^{MT} 木はハイパー木と呼ばれ，d 層の XMSS 木からなる。ここで，XMSS 木は 7.3.1.2 節の鍵生成アルゴリズムで生成される L 木とマークル木からなる木を表す。第 $(d-1)$ 層と第 0 層はそれぞれ，XMSS^{MT} 木の根と葉に相当する。すべての XMSS 木の高さは等しく，Winternitz パラメータもすべて同じ値が用いられる。第 x 層の左から y 番目の XMSS 木の構成で使用される OTS ハッシュアドレス，L 木アドレス，ハッシュ木アドレスの layer address と tree address は，それぞれ $[x]_4$，$[y]_4$ である。

XMSS^{MT} の鍵生成，署名，検証の各アルゴリズムについての詳細は [19] を参照のこと。

7.3.1.4　パラメータの設定と安全性

Hülsing [23] らは，XMSS について安全性証明を与え，選択文書攻撃に対する存在偽造不能性（EUF-CMA）を満たすことを鍵付きハッシュ関数 $\mathsf{F}, \mathsf{H}, \mathsf{H}_{\mathrm{msg}}$ と擬似ランダム関数 R の以下の安全性に帰着している。

- F が以下の性質を満たすこと
  - multi-function, multi-target second preimage resistance (MM-SPR)
  - すべての出力が 2 個以上の原像を持つこと
- H が MM-SPR を満たすこと
- $\mathsf{H}_{\mathrm{msg}}$ が multi-target extended target collision resistance (M-ETCR) を満たすこと
- R が擬似ランダム関数 (PRF) であること

ここで，MM-SPR, M-ETCR は，$\mathsf{F}, \mathsf{H}, \mathsf{H}_{\mathrm{msg}}$ の構成に用いられるハッシュ関数の第二原像攻撃に対する安全性に基づく性質である。一方，PRF は，秘密鍵入力を有するハッシュ関数が擬似ランダム関数であることを要求する。さらに，R による鍵とビットマスクの生成については，ハッシュ関数がランダムオラクルであることが仮定される。

IRTF RFC 8391 [19] では，上述の XMSS の安全性に関する結果に基づいて，$n = 32, 64$ のとき，それぞれ，256 bit 安全性，512 bit 安全性が提供されると記されている。また，量子計算機を用いた攻撃に対してはそれぞれ，128 bit 安全性，256 bit 安全性が提供されると記されている。

IRTF RFC 8391 [19] では，ハッシュ関数として SHA-256 を用いることが要求されているが，オプションとして SHAKE128/256, SHA-512, SHAKE256/512 を用いることが記されている。一方, NIST SP 800-208 では, SHA-256, SHA-256/192, SHAKE256/256, SHAKE256/192 を用いることが認可されている。NIST SP 800-208 [12] と IRTF RFC 8391 [19] の両方に掲載されている SHA-256 を用いる場合の WOTS$^+$, XMSS, XMSS^{MT} のパラメータセットの値の一覧をそれぞれ表 7.2, 7.3, 7.4 に示す。

表 7.2: WOTS$^+$ のパラメータセット

名称	n	w	ℓ
WOTSP-SHA2_256	32	16	67

表 7.3: XMSS のパラメータセットと署名長（単位は Byte）

名称	n	w	ℓ	h	署名長
XMSS-SHA2_10_256	32	16	67	10	2,500
XMSS-SHA2_16_256	32	16	67	16	2,692
XMSS-SHA2_20_256	32	16	67	20	2,820

表 7.4: XMSS^{MT} のパラメータセットと署名長（単位は Byte）

名称	n	w	ℓ	h	d	署名長
XMSSMT-SHA2_20/2_256	32	16	67	20	2	4,963
XMSSMT-SHA2_20/4_256	32	16	67	20	4	9,251
XMSSMT-SHA2_40/2_256	32	16	67	40	2	5,605
XMSSMT-SHA2_40/4_256	32	16	67	40	4	9,893
XMSSMT-SHA2_40/8_256	32	16	67	40	8	18,469
XMSSMT-SHA2_60/3_256	32	16	67	60	3	8,392
XMSSMT-SHA2_60/6_256	32	16	67	60	6	14,824
XMSSMT-SHA2_60/12_256	32	16	67	60	12	27,688

7.3.2 SLH-DSA

SLH-DSA [35] は 7.2.3 節のマークル木の階層構造による署名方式に基づく stateless な署名方式である。SLH-DSA で用いられる 1 回署名方式とマークル木を用いた署名方式はそれぞれ WOTS$^+$ (Winternitz One-Time Signature Plus scheme)，XMSS (eXtended Merkle Signature Scheme) と呼ばれる[*2]。また，XMSS で構成されるマークル木は XMSS 木と呼ばれる。SLH-DSA が 7.2.3 節で述べられた方式と異なる点は，FORS (Forest of Random Subsets) と呼ばれるハッシュ関数に基づく数回 (few-time) 署名方式が導入されている点である。数回署名方式は，一組の公開鍵と秘密鍵の組を用いて，複数個のメッセージに署名できる。SLH-DSA では，メッセージは FORS を用いて署名され，FORS の公開鍵が hypertree と呼ばれる XMSS 木の階層構造による署名方式を用いて署名される。SLH-DSA は，数回署名方式を導入して署名可能な回数を増加させることにより，stateless であることを達成している。なお，WOTS$^+$，XMSS，hypertree，FORS は SLH-DSA の構成要素として使用されるのみであり，それぞれの単独での使用は許容されていない。

SLH-DSA の公開鍵は長さ n Bytes の 2 つの系列 **PK**.root と **PK**.seed である。**PK**.root は hypertree の最上層の XMSS 木の根である。**PK**.seed は無作為に選択される。SLH-DSA の秘密鍵は n Bytes の 2 つの系列 **SK**.seed と **SK**.prf であり，いずれも無作為に選択される。なお，NIST FIPS 205 [35] では，**PK**.seed, **SK**.seed, **SK**.prf の生成に SP 800-90A [4]，SP 800-90B [39]，SP 800-90C [5] で規定されているランダム bit 生成器を使用することが求められている。WOTS$^+$ と FORS のすべての秘密鍵は，**SK**.seed を用いて擬似ランダム関数により生成される。**SK**.prf は，メッセージダイジェストの計算に使用される乱数系列の生成に使用される。

SLH-DSA の署名では，メッセージダイジェストは上記の乱数系列を用いたランダム化されたハッシュ関数により生

[*2] これらの名称は 7.3.1 節の XMSS の対応する署名方式の名称と同一であるが，アルゴリズムには相違点が存在する。

成され，そのメッセージダイジェストの一部を用いてメッセージの署名に用いる FORS の公開鍵と秘密鍵の組が選択される。

SLH-DSA では以下の関数が用いられる。

- $\mathbf{PRF}_{msg} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^* \to \mathbb{B}^n$ はメッセージダイジェストの計算に使用される乱数系列を生成する擬似ランダム関数である。
- $\mathbf{H}_{msg} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^* \to \mathbb{B}^m$ はメッセージダイジェストを計算するハッシュ関数である。
- $\mathbf{PRF} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^{32} \to \mathbb{B}^n$ は WOTS$^+$，FORS の秘密鍵を生成する擬似ランダム関数である。
- $\mathbf{T}_\ell : \mathbb{B}^n \times \mathbb{B}^{32} \times \mathbb{B}^{\ell n} \to \mathbb{B}^n$ は WOTS$^+$，XMSS 木，FORS で用いられるハッシュ関数である。

さらに，$\mathbf{T}_1, \mathbf{T}_2$ について，$\mathbf{F} := \mathbf{T}_1$, $\mathbf{H} := \mathbf{T}_2$ の表記が用いられる。

SLH-DSA では，図 7.4 に示す 7 種のアドレスが用いられる。どのアドレスも長さは 32 Bytes である。各アドレスの layer address と tree address は XMSS 木の階層構造で，ハッシュ関数がどの XMSS 木で用いられるかを表す。これに基づき，FORS 木アドレス，FORS 木根圧縮アドレス，FORS 鍵生成アドレスの layer address の値はすべて 0 と定められている。

layer address	(4 Bytes)
tree address	(12 Bytes)
type = 0	(4 Bytes)
key pair address	(4 Bytes)
chain address	(4 Bytes)
hash address	(4 Bytes)

(a) WOTS$^+$ ハッシュアドレス

layer address	(4 Bytes)
tree address	(12 Bytes)
type = 1	(4 Bytes)
key pair address	(4 Bytes)
0	(4 Bytes)
0	(4 Bytes)

(b) WOTS$^+$ 公開鍵圧縮アドレス

layer address	(4 Bytes)
tree address	(12 Bytes)
type = 2	(4 Bytes)
0	(4 Bytes)
tree height	(4 Bytes)
tree index	(4 Bytes)

(c) ハッシュ木アドレス

layer address = 0	(4 Bytes)
tree address	(12 Bytes)
type = 3	(4 Bytes)
key pair address	(4 Bytes)
tree height	(4 Bytes)
tree index	(4 Bytes)

(d) FORS 木アドレス

layer address = 0	(4 Bytes)
tree address	(12 Bytes)
type = 4	(4 Bytes)
key pair address	(4 Bytes)
0	(4 Bytes)
0	(4 Bytes)

(e) FORS 木根圧縮アドレス

layer address	(4 Bytes)
tree address	(12 Bytes)
type = 5	(4 Bytes)
key pair address	(4 Bytes)
chain address	(4 Bytes)
0	(4 Bytes)

(f) WOTS$^+$ 鍵生成アドレス

layer address = 0	(4 Bytes)
tree address	(12 Bytes)
type = 6	(4 Bytes)
key pair address	(4 Bytes)
0	(4 Bytes)
tree index	(4 Bytes)

(g) FORS 鍵生成アドレス

図 7.4: アドレスの構造

7.3.2.1 WOTS$^+$

WOTS$^+$ は Winternitz one-time signature に基づく 1 回署名方式である。WOTS$^+$ は 2 つのパラメータ n と lg_w を用いる。n はセキュリティパラメータであり，署名されるメッセージ，公開鍵，秘密鍵，署名を構成する系列の byte 長である。lg_w は Winternitz パラメータと呼ばれる正整数 w について $lg_w := \log_2 w$ と定義される。WOTS$^+$ では $lg_w = 4$ と定められており，$w = 16$ である。

WOTS$^+$ の公開鍵，秘密鍵，署名を構成する系列の個数は $len := len_1 + len_2$ で表される。ここで，

$$len_1 := \lceil 8n/lg_w \rceil, \quad len_2 := \lfloor \log_2(len_1(w-1))/lg_w \rfloor + 1$$

である。$lg_w = 4$ なので，$len_1 = 2n$，$len_2 = 3$，$len = 2n + 3$ である。

■チェイニング関数 チェイニング関数 chain の入力は，長さ n Bytes の系列 X，スタートインデクス i，ステップ数 s，**PK**.seed，WOTS$^+$ ハッシュアドレス **ADRS** であり，以下のように定義される。

1. $tmp \leftarrow X$ とする。
2. $i \le j \le i + s - 1$ について，**ADRS** の hash address を j とし，$tmp \leftarrow \mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, tmp)$ とする。
3. tmp を返す。

■鍵生成アルゴリズム 入力は **SK**.seed, **PK**.seed, WOTS$^+$ ハッシュアドレス **ADRS** である。なお，**ADRS** の chain address, hash address の値はいずれも 0 である。

1. $0 \le i \le len - 1$ について，**ADRS** の chain address の値を i とし，

   $$sk_i \leftarrow \mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \text{skADRS}) \qquad pk_i \leftarrow \mathsf{chain}(sk_i, 0, w-1, \mathbf{PK}.\text{seed}, \mathbf{ADRS})$$

   とする。なお，skADRS は WOTS$^+$ 鍵生成アドレスであり，layer address, tree address, key pair address, chain addresss については **ADRS** と同じ値が用いられる。
2. $pk \leftarrow \mathbf{T}_{len}(\mathbf{PK}.\text{seed}, \text{wotspkADRS}, pk_0 \| \cdots \| pk_{len-1})$ とする。ここで，wotspkADRS は WOTS$^+$ 公開鍵圧縮アドレスであり，layer address, tree address, key pair address については **ADRS** と同じ値が用いられる。

公開鍵は pk である。秘密鍵は $sk := (sk_0, sk_1, \ldots, sk_{len-1})$ である。

■署名アルゴリズム 入力は byte 長 n のメッセージ M，**SK**.seed，**PK**.seed，WOTS$^+$ ハッシュアドレス **ADRS** である。**ADRS** の layer address, tree address, key pair address で指定される WOTS$^+$ の秘密鍵を用いて署名が生成される。なお，**ADRS** の chain address, hash address の値はいずれも 0 である。

1. M をそれぞれ長さ lg_w bits の len_1 個のブロックに分割し，先頭から順に $msg_0, msg_1, \ldots, msg_{len_1-1}$ とする。これらを整数とみなすと，$0 \le i \le len_1 - 1$ について，$msg_i \in \{0, 1, \ldots, w-1\}$ である。
2. $csum \leftarrow \sum_{i=0}^{len_1-1} (w - 1 - msg_i)$ とする。
3. $csum \cdot 2^{(8-(len_2 \cdot lg_w \bmod 8)) \bmod 8}$ を長さ $\lceil (len_2 \cdot lg_w)/8 \rceil$ Bytes の系列とみなし，それぞれ長さ lg_w bits の len_2 個のブロックに分割し，先頭から順に $msg_{len_1}, msg_{len_1+1}, \ldots, msg_{len-1}$ とする。
4. $0 \le i \le len - 1$ について，**ADRS** の chain address の値を i とし，

   $$sk_i \leftarrow \mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \text{skADRS}) \qquad sig_i \leftarrow \mathsf{chain}(sk_i, 0, msg_i, \mathbf{PK}.\text{seed}, \mathbf{ADRS})$$

とする。なお，skADRS は WOTS$^+$ 鍵生成アドレスであり，layer address, tree address, key pair address, chain addresss については **ADRS** と同じ値が用いられる。

メッセージ M に対する署名は $sig_0, sig_1, \ldots, sig_{len-1}$ である。

■検証アルゴリズム　SLH-DSA では WOTS$^+$ が単独で使用されることが想定されていないため，検証アルゴリズムは明示されておらず，Winternitz one-time signature の検証に必須の，メッセージと署名の組から対応する公開鍵の候補を計算するアルゴリズムが示されている。なお，このアルゴリズムは，鍵生成と署名のアルゴリズムより容易に導出される。詳細は NIST FIPS 205 [35] を参照のこと。

7.3.2.2　XMSS

XMSS はマークル木を用いた署名方式であり，WOTS$^+$ を用いて構成される。

■鍵生成アルゴリズム　XMSS では，WOTS$^+$ の公開鍵を各葉にもつ高さ h' のマークル木（XMSS 木）を構成することにより，公開鍵が生成される。XMSS 木の高さ $z\,(\geq 0)$ の左から $i\,(\geq 0)$ 番目の節点を $node_{i,j}$ と表記する。入力は **SK**.seed, **PK**.seed, **ADRS** である。

1. $0 \leq i \leq 2^{h'} - 1$ について，$node_{i,0} \leftarrow pk_i$ とする。ここで，pk_i は **SK**.seed, **PK**.seed を用いて計算される WOTS$^+$ の公開鍵である。なお，pk_i の計算に用いられるアドレスの layer address と tree address の値は **ADRS** のそれらと等しく，key pair address の値は $[i]_4$ である。
2. $1 \leq z \leq h'$ について，それぞれ，$0 \leq i \leq 2^{h'-z} - 1$ について，

$$node_{i,z} \leftarrow \mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, node_{2i,z-1} \| node_{2i+1,z-1})$$

とする。ここで，**ADRS** はハッシュ木アドレスであり，tree height の値は $[z]_4$，tree index の値は $[i]_4$ である。

公開鍵は XMSS 木の根 $node_{0,h'}$ であり，秘密鍵は $2^{h'}$ 個の WOTS$^+$ の秘密鍵である。

なお，XMSS の単独での使用が想定されていないことから，NIST FIPS 205 [35] では，鍵生成アルゴリズムは明示されておらず，XMSS 木の各節点を計算する再帰的アルゴリズムが示されている。

■署名アルゴリズム　SLH-DSA では，XMSS で署名されるメッセージは XMSS の公開鍵あるいは FORS の公開鍵のみである。入力は M, **SK**.seed, idx, **PK**.seed, **ADRS** である。M は長さ n Bytes のメッセージ，idx は M の署名に使用される WOTS$^+$ の鍵の key pair address である。

WOTS$^+$ の idx 番目の秘密鍵を用いて M に署名し，XMSS 木の idx 番目の葉の認証パスを計算する。

■検証アルゴリズム　NIST FIPS 205 [35] では，検証に必須の，メッセージと署名の組から対応する公開鍵の候補を計算するアルゴリズムが示されている。このアルゴリズムは鍵生成と署名のアルゴリズムより容易に導出される。詳細は NIST FIPS 205 [35] を参照のこと。

7.3.2.3　Hypertree

SLH-DSA では，hypertree と呼ばれる XMSS 木の階層構造が用いられる。hypertree は d 層の XMSS 木からなり，すべての XMSS 木の高さは等しい。第 $(d-1)$ 層と第 0 層はそれぞれ hypertree の根と葉に相当する。第 x 層の左から y 番目の XMSS 木の構成で使用される WOTS$^+$ ハッシュアドレス，WOTS$^+$ 公開鍵圧縮アドレス，WOTS$^+$ 鍵生成アドレス，ハッシュ木アドレスの layer address と tree address はそれぞれ $[x]_4$，$[y]_{12}$ である。

hypertree の公開鍵は第 $(d-1)$ 層の XMSS の公開鍵である。hypertree の署名，検証の各アルゴリズムについての詳細は NIST FIPS 205 [35] を参照のこと。

7.3.2.4 FORS

FORS は，数回署名方式 HORS [38] に基づく HORST [8] の改良版である。FORS は $k, t := 2^a$ をパラメータとし，長さ ka bits の系列に署名を行う。

■鍵生成アルゴリズム 入力は **SK**.seed, **PK**.seed, FORS 木アドレス **ADRS** である。なお，**ADRS** の layer address, tree address, key pair address は，生成された FORS の公開鍵の署名に用いられる WOTS$^+$ の鍵の生成に用いられるアドレスのそれらの値と等しい。

1. $0 \leq i \leq kt-1$ について，

$$sk_i \leftarrow \mathbf{PRF}(\mathbf{PK}.\mathrm{seed}, \mathbf{SK}.\mathrm{seed}, \mathrm{skADRS}) \qquad node_{i,0} \leftarrow \mathbf{F}(\mathbf{PK}.\mathrm{seed}, \mathbf{ADRS}, sk_i)$$

   とする。ここで，skADRS は FORS 鍵生成アドレスであり，layer address, tree address, key pair address については **ADRS** と同じ値が用いられ，tree index の値は $[i]_4$ である。また，**ADRS** の tree index の値は $[i]_4$ である。
2. $1 \leq z \leq a$ について，それぞれ，$0 \leq i \leq k \cdot 2^{a-z} - 1$ について，

$$node_{i,z} \leftarrow \mathbf{H}(\mathbf{PK}.\mathrm{seed}, \mathbf{ADRS}, node_{2i,z-1} \| node_{2i+1,z-1})$$

   とする。ここで，**ADRS** の tree height の値は $[z]_4$，tree index の値は $[i]_4$ である。
3. $pk \leftarrow \mathbf{T}_k(\mathbf{PK}.\mathrm{seed}, \mathrm{forspkADRS}, node_{0,a} \| \cdots \| node_{k-1,a})$ とする。ここで，forspkADRS は FORS 木根圧縮アドレスであり，layer address, tree address, key pair address については **ADRS** と同じ値が用いられる。

このアルゴリズムにより，$node_{0,a}, node_{1,a}, \ldots, node_{k-1,a}$ を根とする k 個のマークル木が構成されている。公開鍵は pk である。秘密鍵は $sk_0, sk_1, \ldots, sk_{kt-1}$ である。

■署名アルゴリズム 長さ $k \cdot a$ bits のメッセージダイジェスト md をそれぞれ長さ a bits の k 個のブロック $md_0, md_1, \ldots, md_{k-1}$ に分割する。すなわち，$md = md_0 \| md_1 \| \cdots \| md_{k-1}$ である。さらに，md_i を 2 進数表記の非負整数とみなす。md の署名は $sk_{0 \cdot t + md_0}, sk_{1 \cdot t + md_1}, \ldots, sk_{(k-1)t + md_{k-1}}$ と，$0 \leq i \leq k-1$ について，$node_{i,a}$ を根とするマークル木の $node_{i \cdot t + md_i, 0}$ の認証パスである。詳細は NIST FIPS 205 [35] を参照のこと。

■検証アルゴリズム NIST FIPS 205 [35] では，検証に必須の，メッセージと署名の組から対応する公開鍵の候補を計算するアルゴリズムが示されている。詳細は NIST FIPS 205 [35] を参照のこと。

7.3.2.5 SLH-DSA

前節までの構成要素を用いて SLH-DSA の署名が構成される。SLH-DSA のパラメータは以下のとおりである。

- セキュリティパラメータ n（単位は Byte）
- hypertree のパラメータ $h, d, h'(= h/d)$
- FORS のパラメータ a, k
- WOTS$^+$ のパラメータ lg_w
- メッセージダイジェストの byte 長 $m = \lceil (h - h')/8 \rceil + \lceil h'/8 \rceil + \lceil (k \cdot a)/8 \rceil$

■鍵生成アルゴリズム $\mathbf{SK}.\mathrm{seed}, \mathbf{SK}.\mathrm{prf} \in \mathbb{B}^n$ はいずれも無作為に選択される。$\mathbf{PK}.\mathrm{seed} \in \mathbb{B}^n$ は無作為に選択される。$\mathbf{PK}.\mathrm{root} \in \mathbb{B}^n$ は hypertree の第 $(d-1)$ 層の XMSS 木の根である。秘密鍵は $\mathbf{SK}.\mathrm{seed}, \mathbf{SK}.\mathrm{prf}, \mathbf{PK}.\mathrm{seed}, \mathbf{PK}.\mathrm{root}$ である。公開鍵は $\mathbf{PK}.\mathrm{seed}, \mathbf{PK}.\mathrm{root}$ である。したがって，秘密鍵，公開鍵のサイズはそれぞれ，$4n$ Bytes，$2n$ Bytes である。

■署名アルゴリズム メッセージ M の署名は以下のように生成される。

1. $R := \mathbf{PRF}_{msg}(\mathbf{SK}.\mathrm{prf}, opt\_rand, M)$ とする。ここで，opt_rand を $\mathbb{B}^n$ の乱数とすることがデフォルトとされており，特にサイドチャネル攻撃が懸念される場合については強く推奨されているが，乱数生成器が利用可能でない場合は $opt\_rand = \mathbf{PK}.\mathrm{seed}$ とすることが許容されている。
2. $digest := \mathbf{H}_{msg}(R, \mathbf{PK}.\mathrm{seed}, \mathbf{PK}.\mathrm{root}, M)$ とする。$digest$ の最初の $\lceil (k \cdot a)/8 \rceil$ Bytes，次の $\lceil (h - h')/8 \rceil$ Bytes，その次の $\lceil h'/8 \rceil$ Bytes をそれぞれ md，整数 idx_{tree} の 2 進数表記，整数 idx_{leaf} の 2 進数表記とする。
3. hypertree の第 0 層の左から idx_{tree} 番目の XMSS 木の左から idx_{leaf} 番目の葉に対応する FORS の鍵を用いて md の先頭 $k \cdot a$ bits に対する署名を生成する。
4. 上の署名で用いられた FORS の公開鍵への hypertree による署名を生成する。

M の署名は R，md への FORS による署名，md への署名の検証に用いられる FORS の公開鍵への hypertree による署名からなる。したがって，署名のサイズは $(1 + k(a+1) + h + d \cdot len)n$ Bytes である。

SLH-DSA では，署名アルゴリズムに与えられるメッセージ M を署名対象の内容から生成する二つの方法が定められている。これらは pure 版，pre-hash 版と呼ばれている。署名アルゴリズムに対して，pure 版ではコンテクストと署名対象の内容とが与えられ，pre-hash 版ではコンテクストと署名対象の内容のハッシュ値とが与えられる。なお，コンテクストは長さが高々 255 Bytes の系列であり，デフォルトでは空列である。詳細については NIST FIPS 205 [35] を参照のこと。

■検証アルゴリズム 署名アルゴリズムより容易に導出されるので，詳細については NIST FIPS 205 [35] を参照のこと。

7.3.2.6 パラメータの設定と安全性

SLH-DSA については，表 7.5 の 12 個のパラメータセットが示されている。この表の最左欄の名称は，使用されるハッシュ関数とセキュリティパラメータ n の bit 長を単位とした値を示している。さらに，s と f はそれぞれ，署名サイズ，計算時間が小さくなるよう定められたパラメータセットであることを示している。また，安全性レベルは NIST PQC 標準化プロジェクトの Call for Proposals に記された安全性強度のカテゴリである。これらのパラメータセットは，一組の公開鍵と秘密鍵により高々 2^{64} 個のメッセージが署名される場合の選択文書攻撃に対する存在偽造不能性を考慮して定められている。

Hülsing と Kudinov [21] は，SPHINCS$^+$ が選択文書攻撃に対する存在偽造不能性 (EUF-CMA) を満たすことをハッシュ関数 $\mathbf{T}_\ell$，$\mathbf{H}_{msg}$，擬似ランダム関数 $\mathbf{PRF}, \mathbf{PRF}_{msg}$ の以下の安全性に帰着している。

- $\mathbf{T}_\ell$ が以下の性質を満たすこと
  - single-function, multi-target collision resistance (SM-TCR)
  - single-function, multi-target preimage resistance (SM-PRE)
  - single-function, multi-target decisional second preimage resistance (SM-DSPR)
  - single-function, multi-target undetectability (SM-UD)

表 7.5: SLH-DSA のパラメータセット。公開鍵長，署名長の単位は Byte である。

名称	n	h	d	h'	a	k	lg_w	m	安全性レベル	公開鍵長	署名長
SLH-DSA-SHA2-128s SLH-DSA-SHAKE-128s	16	63	7	9	12	14	4	30	レベル 1	32	7,856
SLH-DSA-SHA2-128f SLH-DSA-SHAKE-128f	16	66	22	3	6	33	4	34	レベル 1	32	17,088
SLH-DSA-SHA2-192s SLH-DSA-SHAKE-192s	24	63	7	9	14	17	4	39	レベル 3	48	16,224
SLH-DSA-SHA2-192f SLH-DSA-SHAKE-192f	24	66	22	3	8	33	4	42	レベル 3	48	35,664
SLH-DSA-SHA2-256s SLH-DSA-SHAKE-256s	32	64	8	8	14	22	4	47	レベル 5	64	29,792
SLH-DSA-SHA2-256f SLH-DSA-SHAKE-256f	32	68	17	4	9	35	4	49	レベル 5	64	49,856

- $\mathbf{H}_{msg}$ が interleaved target subset resilience (ITSR) を満たすこと
- $\mathbf{PRF}, \mathbf{PRF}_{msg}$ が擬似ランダム関数 (PRF) であること

ここで，SM-TCR, SM-DSPR, ITSR は，$\mathbf{T}_\ell, \mathbf{H}_{msg}$ の構成に用いられるハッシュ関数の第二原像攻撃に対する安全性に基づく性質であり，SM-PRE は原像攻撃に対する安全性に基づく性質である。一方，SM-UD, PRF は，秘密鍵入力を有するハッシュ関数が擬似ランダム関数であることを要求する。

Barbosa ら [3] は，SPHINCS$^+$ で用いられている XMSS について，コンピュータで検証された安全性証明を与えている。

さらに，NIST FIPS 205 [35] には，SLH-DSA の実装をサイドチャネル攻撃 [24] や故障攻撃 [1, 11, 17, 40] から保護するための注意が払われなければならないことが記されている。

7.3.2.7 ハッシュ関数の実現法

SLH-DSA の関数はすべて，SHAKE256，SHA-2 のうちのいずれかを用いて構成される。これらの構成は，SPHINCS$^+$ で simple な実現と呼ばれる構成であり，7.2.4 節で述べられたビットマスクは用いられていない。

SHAKE256 を用いた構成は以下のとおりである。

$$\begin{aligned}
\mathbf{H}_{msg}(R, \mathbf{PK}.\text{seed}, \mathbf{PK}.\text{root}, M) &:= \text{SHAKE256}(R \| \mathbf{PK}.\text{seed} \| \mathbf{PK}.\text{root} \| M, 8m) \\
\mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \mathbf{ADRS}) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \| \mathbf{ADRS} \| \mathbf{SK}.\text{seed}, 8n) \\
\mathbf{PRF}_{msg}(\mathbf{SK}.\text{prf}, opt_rand, M) &:= \text{SHAKE256}(\mathbf{SK}.\text{prf} \| opt_rand \| M, 8n) \\
\mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_1) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \| \mathbf{ADRS} \| M_1, 8n) \\
\mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_2) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \| \mathbf{ADRS} \| M_2, 8n) \\
\mathbf{T}_\ell(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_\ell) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \| \mathbf{ADRS} \| M_\ell, 8n)
\end{aligned}$$

安全性レベル 1 に対する SHA-2 を用いた構成は以下のとおりである。

$$\begin{aligned}
\mathbf{H}_{msg}(R, \mathbf{PK}.\text{seed}, \mathbf{PK}.\text{root}, M) &:= \text{MGF1-SHA-256}(R \| \mathbf{PK}.\text{seed} \| \text{SHA-256}(R \| \mathbf{PK}.\text{seed} \| \mathbf{PK}.\text{root} \| M, m)) \\
\mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \mathbf{ADRS}) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| \mathbf{SK}.\text{seed})) \\
\mathbf{PRF}_{msg}(\mathbf{SK}.\text{prf}, opt_rand, M) &:= \mathsf{Trunc}_n(\text{HMAC-SHA-256}(\mathbf{SK}.\text{prf} \| opt_rand \| M)) \\
\mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_1) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| M_1)) \\
\mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_2) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| M_2)) \\
\mathbf{T}_\ell(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_\ell) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| M_\ell))
\end{aligned}$$

安全性レベル 3, 5 に対する SHA-2 を用いた構成は以下のとおりである。

$$\begin{aligned}
\mathbf{H}_{msg}(R, \mathbf{PK}.\text{seed}, \mathbf{PK}.\text{root}, M) &:= \text{MGF1-SHA-512}(R \| \mathbf{PK}.\text{seed} \| \text{SHA-512}(R \| \mathbf{PK}.\text{seed} \| \mathbf{PK}.\text{root} \| M, m)) \\
\mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \mathbf{ADRS}) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| \mathbf{SK}.\text{seed})) \\
\mathbf{PRF}_{msg}(\mathbf{SK}.\text{prf}, opt_rand, M) &:= \mathsf{Trunc}_n(\text{HMAC-SHA-512}(\mathbf{SK}.\text{prf} \| opt_rand \| M)) \\
\mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_1) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| M_1)) \\
\mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_2) &:= \mathsf{Trunc}_n(\text{SHA-512}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 128-n) \| \mathbf{ADRS}^c \| M_2)) \\
\mathbf{T}_\ell(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_\ell) &:= \mathsf{Trunc}_n(\text{SHA-512}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 128-n) \| \mathbf{ADRS}^c \| M_\ell))
\end{aligned}$$

ここで，MGF1-SHA-256，MGF1-SHA-512 は RFC 8017 [32] の Appendix B.2.1 に記載されている MGF1 であり，HMAC-SHA-256，HMAC-SHA-512 は FIPS 198-1 [36] の HMAC である。また，$\mathsf{Trunc}_l(x)$ は byte 系列 x の左端から l Bytes を出力する関数であり，$\mathsf{toByte}(z, l)$ は整数 z を長さ l Bytes の byte 系列に変換する関数である。さらに，$\mathbf{ADRS}^c$ は $\mathbf{ADRS}$ の layer address，tree address，type をそれぞれ 1 Byte，8 Bytes，1 Byte に短縮した長さ 22 Bytes のアドレスである。

SPHINCS$^+$ では，当初，SHA-2 を用いた実現で SHA-256 のみが用いられていたが，SHA-256 を用いた実現では安全性のレベル 5 が達成できないことを示す攻撃 [37] が示されたことから，SPHINCS$^+$ v.3.1 では，安全性レベル 3, 5 について，$\mathbf{H}_{msg}$, $\mathbf{PRF}_{msg}$, $\mathbf{H}$, $\mathbf{T}_\ell$ が SHA-512 を用いて実現されることとなり，SLH-DSA でもそれに従っている。

7.4 ハッシュ関数に基づく署名技術に関するまとめ

本章では，ハッシュ関数に基づく署名技術として，XMSS と SLH-DSA を取り上げた。これらはいずれも 7.2 節で述べた代表的なハッシュ関数に基づく署名方式に基づく構造を有する。XMSS [19] は NIST の推奨アルゴリズムであり [12]，SLH-DSA は NIST PQC 標準化プロジェクトで選出された SPHINCS$^+$[2] に基づく標準アルゴリズムである。

ハッシュ関数に基づく署名技術の安全性はハッシュ関数の第二原像攻撃に対する安全性に依存しているが，XMSS, SLH-DSA については，秘密鍵入力を有するハッシュ関数が擬似ランダム関数であることにも依存する。また，偽造攻撃の計算量は，ハッシュ関数がランダムオラクルであることを仮定して見積もられている。なお，XMSS で用いられるビットマスクの生成についてもハッシュ関数がランダムオラクルであることが仮定される。

ハッシュ関数に基づく署名技術については，stateful であること，すなわち，各メッセージの署名に用いられる 1 回署名の秘密鍵を 2 回以上使用することのないよう管理しなければならないことが問題であった。XMSS は stateful な署名方式であり，それを推奨アルゴリズムとする NIST SP 800-208 [12] には，ハッシュ関数に基づく stateful な署名方式は一般的な使用には適するものでなく，近い将来に実装が必要であり，その実装が長期間の使用を予定されており，かつ，使用開始後に他の署名方式への移行が実用的でないような応用での使用が意図されていると述べられている。

SLH-DSA は XMSS の設計で得られた知見に基づいて設計されており，XMSS^{MT} と同様の構造を有するが，各メッセージの署名に一つの秘密鍵で数回署名可能な FORS を用いることによって署名可能な回数を増加させることにより，

stateless であることを達成している。

第 7 章の参照文献

[1] D. Amiet, L. Leuenberger, A. Curiger, P. Zbinden. FPGA-based SPHINCS$^+$ Implementations: Mind the Glitch. DSD. IEEE, 2020, pp. 229–237.

[2] J.-P. Aumasson et al. SPHINCS$^+$ Submission to the NIST post-quantum project, v.3.1. `https://sphincs.org/data/sphincs+-r3.1-specification.pdf`. 2022-06. （2024-03-08 閲覧）.

[3] M. Barbosa, F. Dupressoir, B. Grégoire, A. Hülsing, M. Meijers, P.-Y. Strub. Machine-Checked Security for XMSS as in RFC 8391 and SPHINCS$^+$. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 421–454.

[4] E. Barker, J. Kelsey. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. NIST SP 800-90A Rev. 1, `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf`. 2015-06.

[5] E. Barker, J. Kelsey, K. McKay, A. Roginsky, M. S. Turan. Recommendation for Random Bit Generator (RBG) Constructions. NIST SP 800-90C (4th public draft), `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90C.4pd.pdf`. 2024-07. (2025-02-17 閲覧).

[6] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, P. Schwabe. The SPHINCS$^+$ Signature Framework. CCS. ACM, 2019, pp. 2129–2146.

[7] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, P. Schwabe. The SPHINCS+ Signature Framework. Cryptology ePrint Archive, Paper 2019/1086. 2019. `https://eprint.iacr.org/2019/1086`.

[8] D. J. Bernstein et al. SPHINCS: Practical Stateless Hash-Based Signatures. EUROCRYPT (1). Vol. 9056. Lecture Notes in Computer Science. Springer, 2015, pp. 368–397.

[9] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche. Sponge functions. ECRYPT Hash Workshop. 2007.

[10] J. Buchmann, E. Dahmen, A. Hülsing. XMSS – A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions. PQCrypto. Vol. 7071. Lecture Notes in Computer Science. Springer, 2011, pp. 117–129.

[11] L. Castelnovi, A. Martinelli, T. Prest. Grafting Trees: A Fault Attack Against the SPHINCS Framework. PQCrypto. Vol. 10786. Lecture Notes in Computer Science. Springer, 2018, pp. 165–184.

[12] D. Cooper, D. Apon, Q. Dang, M. Davidson, M. Dworkin, C. Miller. Recommendation for Stateful Hash-Based Signature Schemes. NIST SP 800-208, `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-208.pdf`. 2020-10.

[13] E. Dahmen, K. Okeya, T. Takagi, C. Vuillaume. Digital Signatures Out of Second-Preimage Resistant Hash Functions. PQCrypto. Vol. 5299. Lecture Notes in Computer Science. Springer, 2008, pp. 109–123.

[14] I. Damgård. A Design Principle for Hash Functions. CRYPTO. Vol. 435. Lecture Notes in Computer Science. Springer, 1989, pp. 416–427.

[15] W. Diffie, M. E. Hellman. New directions in cryptography. IEEE Trans. Inf. Theory. Vol. 22, Num. 6 (1976), pp. 644–654.

[16] C. Dods, N. P. Smart, M. Stam. Hash Based Digital Signature Schemes. IMACC. Vol. 3796. Lecture Notes in Computer Science. Springer, 2005, pp. 96–115.

[17] A. Genêt. On Protecting SPHINCS+ Against Fault Attacks. IACR Trans. Cryptogr. Hardw. Embed. Syst. Vol. 2023, Num. 2 (2023), pp. 80–114.

[18] L. K. Grover. A fast quantum mechanical algorithm for database search. STOC. ACM, 1996, pp. 212–219.

[19] A. Huelsing, D. Butin, S.-L. Gazdag, J. Rijneveld, A. Mohaisen. XMSS: eXtended Merkle Signature Scheme. RFC 8391, `https://www.rfc-editor.org/info/rfc8391`. 2018-05.

[20] A. Hülsing. W-OTS$^+$ – Shorter Signatures for Hash-Based Signature Schemes. AFRICACRYPT. Vol. 7918. Lecture Notes in Computer Science. Springer, 2013, pp. 173–188.

[21] A. Hülsing, M. A. Kudinov. Recovering the Tight Security Proof of SPHINCS$^+$. ASIACRYPT (4). Vol. 13794. Lecture Notes in Computer Science. Springer, 2022, pp. 3–33.

[22] A. Hülsing, L. Rausch, J. Buchmann. Optimal Parameters for XMSS MT. CD-ARES Workshops. Vol. 8128. Lecture Notes in Computer Science. Springer, 2013, pp. 194–208.

[23] A. Hülsing, J. Rijneveld, F. Song. Mitigating Multi-target Attacks in Hash-Based Signatures. Public Key Cryptography (1). Vol. 9614. Lecture Notes in Computer Science. Springer, 2016, pp. 387–416.

[24] M. J. Kannwischer, A. Genêt, D. Butin, J. Krämer, J. Buchmann. Differential Power Analysis of XMSS and SPHINCS. COSADE. Vol. 10815. Lecture Notes in Computer Science. Springer, 2018, pp. 168–188.

[25] P. Lafrance, A. Menezes. On the security of the WOTS-PRF signature scheme. Adv. Math. Commun. Vol. 13, Num. 1 (2019), pp. 185–193.

[26] L. Lamport. Constructing digital signatures from a one-way function. SRI International Technical Report, CSL-98. 1979-10.

[27] F. T. Leighton, S. Micali. Large provably fast and secure digital signature schemes based on secure hash functions. 1995-07. US Patent 5,432,852.

[28] D. McGrew, M. Curcio, S. Fluhrer. Leighton-Micali Hash-Based Signatures. RFC 8554, `https://www.rfc-editor.org/info/rfc8554`. 2019-04.

[29] R. Merkle. Secrecy, Authentication, and Public Key Systems. PhD thesis. Stanford University, 1979. URL: `https://www.merkle.com/papers/Thesis1979.pdf`.

[30] R. C. Merkle. A Certified Digital Signature. CRYPTO. Vol. 435. Lecture Notes in Computer Science. Springer, 1989, pp. 218–238.

[31] R. C. Merkle. One Way Hash Functions and DES. CRYPTO. Vol. 435. Lecture Notes in Computer Science. Springer, 1989, pp. 428–446.

[32] K. Moriarty, B. Kaliski, J. Jonsson, A. Rusch. PKCS #1: RSA Cryptography Specifications Version 2.2. RFC 8017, `https://www.rfc-editor.org/info/rfc8017`. 2016-11.

[33] NIST. Secure Hash Standard (SHS). NIST FIPS 180-4, `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf`. 2015-08.

[34] NIST. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. NIST FIPS 202, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf. 2015-08.

[35] NIST. Stateless Hash-Based Digital Signature Standard. NIST FIPS 205, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf. 2024-08.

[36] NIST. The Keyed-Hash Message Authentication Code (HMAC). NIST FIPS 198-1, https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.198-1.pdf. 2008-07.

[37] R. A. Perlner, J. Kelsey, D. A. Cooper. Breaking Category Five SPHINCS$^+$ with SHA-256. PQCrypto. Vol. 13512. Lecture Notes in Computer Science. Springer, 2022, pp. 501–522.

[38] L. Reyzin, N. Reyzin. Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying. ACISP. Vol. 2384. Lecture Notes in Computer Science. Springer, 2002, pp. 144–153.

[39] M. S. Turan, E. Barker, J. Kelsey, K. McKay, M. Baish, M. Boyle. Recommendation for the Entropy Sources Used for Random Bit Generation. NIST SP 800-90B, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf. 2018-01.

[40] A. Wagner, V. Wesselkamp, F. Oberhansl, M. Schink, E. Strieder. Faulting Winternitz One-Time Signatures to Forge LMS, XMSS, or SPHINCS$^+$ Signatures. PQCrypto. Vol. 14154. Lecture Notes in Computer Science. Springer, 2023, pp. 658–687.

CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）
[CRYPTREC GL-2004-2022]

発行日：2023 年 3 月 31 日（第 1 版）
発行者

- 〒184-8795
  東京都小金井市貫井北町四丁目２番１号
  国立研究開発法人情報通信研究機構
  （サイバーセキュリティ研究所　セキュリティ基盤研究室）
  NATIONAL INSTITUTE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
  4-2-1 NUKUI-KITAMACHI, KOGANEI
  TOKYO, 184-8795 JAPAN
- 〒113-6591
  東京都文京区本駒込二丁目２８番８号
  独立行政法人情報処理推進機構
  （セキュリティセンター セキュリティ技術評価部 暗号グループ）
  INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN
  2-28-8 HONKOMAGOME, BUNKYO-KU
  TOKYO, 113-6591 JAPAN

CRYPTREC
耐量子計算機暗号の研究動向調査報告書

2025 年 3 月

CRYPTREC
暗号技術調査ワーキンググループ（耐量子計算機暗号）

目次

第 1 章

はじめに

暗号は情報を保護するための基礎的な手段である。基本的な暗号の分類として共通鍵暗号と公開鍵暗号があり，さらに公開鍵暗号の下位分類として通信相手の認証などを目的とした署名方式，情報の守秘を目的とした公開鍵暗号方式[*1]，秘密鍵の共有を目的とした鍵共有が存在する[*2] 。これらを含めた基本的な暗号方式を部品（プリミティブ）とした高機能暗号 [5] が数多く提案されている。2025 年現在，署名目的で DSA,ECDSA 等，情報の守秘目的で RSA-OAEP 等，秘密鍵共有の目的では DH,ECDH 等[*3] が国際的な標準暗号方式 [96] として用いられており，日本においても電子政府推奨暗号 [141] とされている。

これらの暗号方式の安全性と深く関わる計算問題として，素因数分解問題や楕円曲線上の離散対数問題があり，古典コンピュータ[*4]では効率的に解くことが困難であると信じられている。このことから，RSA 暗号や ECDSA 署名はある程度の大きさの鍵長を用いることで安全性が保てると考えられている [140]。一方で，Shor の量子アルゴリズム [121, 123] はこれらの計算問題を効率的に解くため，量子コンピュータの高性能化が情報セキュリティに影響を及ぼすとされている。以上の背景のもと，古典コンピュータ上での効率的な実装が可能であり，かつ古典・量子双方のコンピュータを用いた攻撃に対しても安全性を確保できる暗号方式が必要とされている。

本報告書で扱う耐量子計算機暗号の範囲 量子コンピュータによる攻撃への耐性を耐量子計算機性と呼び，耐量子計算機性を持つ暗号技術を耐量子計算機暗号（Post-Quantum Cryptography: PQC）と呼称する。しかしながら，耐量子計算機性の定式化はそれぞれの暗号技術の定式化を踏まえて行われており，一義的な意味で用いられる単語ではないことに注意が必要である。

例えば，公開鍵暗号方式は古典アルゴリズムの 3 つ組として定式化され，それらの古典安全性モデルの議論は IND-CCA 安全性をデファクトスタンダードとして収束している。それを踏まえ耐量子計算機性を持つ公開鍵暗号方式は，同じ定式化を持ちかつ耐量子計算機性の安全性モデルを満たす方式と捉えることができる。ただし 2025 年現在 IND-CCA 安全性の量子版は様々に提案されており，例えば Boneh と Zhandry による IND-qCCA2[24]，Chevalier らによる qIND-qCCA2[37] など，攻撃者が量子計算機をどのように用いるのかという点で定式化に細かい違いがある。
また，第 2 章冒頭にあるように，暗号技術のレイヤーを離れそれらを利用する暗号システムに関する耐量子計算機性を考える事も可能である。

PQC の考え方が出現した文脈は RSA，ECDSA の代替となる公開鍵暗号の開発であり [29]，PQC 候補とされるほとんどの暗号技術は古典コンピュータでの実装を前提として提案されている。この，古典コンピュータによる実装可能

[*1] 本報告書の中では公開鍵暗号を Public-Key Cryptography の意味で用い，その下位分類としての Public-Key Encryption を公開鍵暗号方式と表記する。
[*2] 基本的な暗号方式の定義と性質に関しては，例えば教科書 [148, 1.3 節] などを参照。
[*3] これらの方式には多くの解説記事があるが，DH，DSA に関しては例えば [179] を，ECDH,ECDSA に関しては [146] がある。
[*4] 理論的には決定的チューリングマシンを物理的に実装した計算機で，狭義においては CMOS 半導体を用いた論理回路による計算機を指す。現在普及しているコンピュータとほぼ同義である。

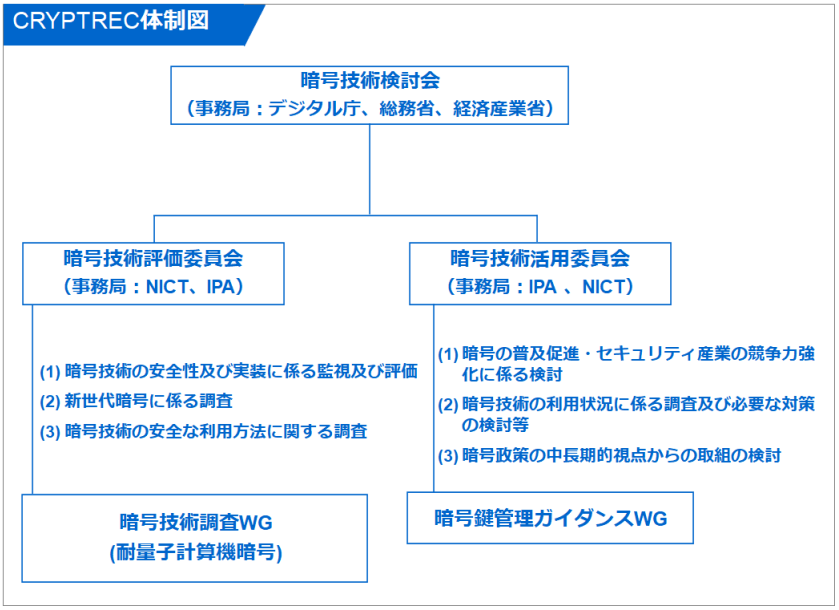


図 1.1: 2024 年度 CRYPTREC 体制図

性は PQC と量子暗号・量子鍵共有を区別する点とされることもある（例えば [85, p.3] を参照）。

以上の状況と本報告書の中で扱う暗号技術の種類（1.4 節も参照）を踏まえ，報告書内では特に断りのない場合，耐量子計算機暗号（PQC）の言葉を，古典アルゴリズムの組み合わせにより定式化され，かつ耐量子計算機性を持つことを技術的に判断できる暗号方式とする。例えば NIST PQC 標準化プロジェクトでの選定基準 [101, 4.A 節] では，公開鍵暗号方式，鍵共有（KEM），署名方式に対して IND-CCA2，EUF-CMA 等の古典の安全性ゲームから古典または量子による多項式時間帰着を行った先の計算問題の量子計算量がある値よりも大きいという基準が用いられ，暗号方式の提案者はその根拠を提案書に述べている。

本報告書の背景および調査内容 近年の世界的な量子コンピュータの開発と商用マシンの普及と並行して，PQC に関する研究及びその標準化に向けた活動も世界各国の組織で進んでおり，国内でも PQC の研究動向を把握する必要性がある。2020 年度第 2 回暗号技術検討会において，2021 年度から暗号技術評価委員会の活動計画として 2 年をかけて PQC の研究動向を調査し，ガイドラインを作成することが決定された。暗号技術評価委員会は暗号技術調査ワーキンググループ（耐量子計算機暗号）を設置し，ワーキンググループにおいて 2022 年 9 月 30 日までの調査結果をガイドライン [4] と調査報告書 [3] としてまとめ，出版した。その後，2022 年度第 2 回暗号技術検討会において，調査活動を継続しさらに，2 年間の研究動向調査を行い新たなガイドラインと調査報告書を作成することが決定され，暗号技術評価委員会は暗号技術調査ワーキンググループ（耐量子計算機暗号）を設置した（図 1.1）。

本ワーキンググループでは PQC の代表的な候補である 5 種類の分類（格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術，同種写像に基づく暗号技術，ハッシュ関数に基づく署名技術）について調査し，原則 2024 年 9 月 30 日までの調査結果をガイドラインと調査報告書にまとめた。本調査報告書の中で「現在」と表記する場合，特に断りがなければ上記 2024 年 9 月 30 日時点での情報を指すものとする。

ガイドラインは暗号初学者を対象としており，調査報告書は暗号についての知見のある技術者や専門家を対象としている。第 1 章ではガイドラインと調査報告書の概要，PQC を必要とする背景，研究及び標準化に関する動向，調査対象とした PQC の種類についてまとめている。第 2 章では PQC の活用方法と移行に関する内容，特に守秘・鍵共有・署名のための PQC の利用などについて記載している。第 3 章以降では暗号技術に携わる研究者及び技術者を読者として想定し，PQC の代表的な候補である 5 種類の分類をまとめた。ただし，これらの章ではガイドラインの記載内容

は調査報告書の簡略版となっており，ガイドラインでは専門的な内容を省略し，暗号初学者が代表的な PQC 方式を把握するために最小限の内容のみを記載した。

共通鍵暗号と暗号学的ハッシュ関数の耐量子計算機性　本報告書では詳しく述べていないが，共通鍵暗号や暗号学的ハッシュ関数に対しても古典的な定式化を踏まえた上で量子的な攻撃に対する安全性モデルが提案されている。代表的な量子攻撃モデルとして，復号オラクルに対して重ね合わせクエリを許さない Q1 モデルと許される Q2 モデル [73] がある。

Grover の量子検索アルゴリズム [61] による共通鍵暗号方式の安全性の低下 [26] や暗号学的ハッシュ関数の衝突発見の高速化 [27] が知られている。アルゴリズムの最適性 [138] から，攻撃の量子計算量は鍵長の指数関数であり，ほぼ全ての共通鍵暗号，暗号学的ハッシュ関数は耐量子計算機性を持つものと認識されている。ただし，[74, 16] のように共通鍵暗号の暗号利用モードによっては安全性が著しく低下する例は知られている。パラメータに関しても鍵長を数倍に伸ばすだけで量子攻撃計算量を古典攻撃計算量と同等に増やすことが可能であり，公開鍵暗号のように全く異なるアルゴリズムへの移行が必要とされない分影響は限定的と考えられている [147, 49]。共通鍵暗号方式の安全性への影響を調査した報告書として CRYPTREC による [147]，日本銀行金融研究所による [161] 等に詳しい記述がある。

1.1　暗号の安全性に影響のある量子コンピュータの開発状況

量子コンピュータは重ね合わせ，エンタングルメント等の量子的な物理現象を用いて計算を行うコンピュータの総称である（[151, 第 2 章]，[149] 等を参照）。本節では，暗号の安全性に影響を及ぼすと考えられる量子コンピュータの開発状況についてまとめる。

1.1.1　量子コンピュータの分類

量子コンピュータの開発は世界中で進められており，その形も多様であるが計算モデル，物理的実装，性能により分類できる。

計算モデルによる分類　量子計算の基本的な計算操作と物理的操作の対応関係を表すモデルにより，量子回路型計算，測定型量子計算，断熱型量子計算，アナログ量子シミュレーション，トポロジカル量子計算，ホロノミック量子計算等に分類できる[*5]。

量子回路型計算，測定型量子計算　特に量子回路型計算，測定型量子計算では多くの種類の物理実装が存在する。超伝導量子ビット，冷却原子（中性原子），イオントラップ，シリコン量子，光量子，カラーセンター量子コンピュータ等の開発が進められている。Shor のアルゴリズム等，暗号に大きな影響のあるアルゴリズムは量子回路を用いて記述されていることから，これらのコンピュータの大規模化が現代暗号に大きな影響を与えると考えられる。1.2.2 節に述べるように，多くの素因数分解実験が量子回路型計算のフレームワークで行われている。

断熱型量子計算と量子アニーリング　断熱型量子計算は基底状態が簡単に用意できる初期ハミルトニアンから，組み合わせ最適化問題の解が基底状態に対応するようなハミルトニアンへとゆっくりと変化させることで解を得る計算フレームワーク [14, Def. 1] である。断熱型量子計算の下位分類の中で特に量子アニーリング（Quantum Annealing: QA）はクラウドサービスを通じた商用コンピュータが提供されていることから注目を集めている。

量子アニーリングは元々，Apolloni ら [19] によりシミュレーテッドアニーリング（Simulated Annealing: SA）に類似したアルゴリズムを量子的に構成したことから名付けられたものであるが，現在では量子断熱計算のモデル [14, Def. 1] における条件を開放系，有限時間に緩め，ハミルトニアンをイジングモデルに制限した計算フレームワークを

[*5] 分類に関しては [149, 133, 51] および [70, Sect 1.6] を参照。

指すものと見なされている [168, § 3]。イジングモデルを用いた素因数分解実験も数多く行われている。(1.2.2 節を参照)

ハミルトニアンの形に制限のない量子断熱計算の計算能力は量子回路型計算と多項式時間等価であり，計算量クラス BQP に属する [9]。量子回路とハミルトニアンを互いに多項式サイズの差で変換する手法も上記論文内で与えられてはいるものの，2025 年現在でその変換を暗号方式への攻撃に応用した例は確認されていない。

3-XORSAT 問題のように古典多項式時間で計算可能であるにも関わらず，自然な形でインスタンスの変換を行ったイジングモデルによる量子アニーリングでは効率的に解くことが困難であることが示唆されている計算問題の存在も知られており [71]，暗号に関係する問題が同様の性質を持つかどうかが未解決問題となっている。

量子ゲート型と量子アニーリング型 量子回路型計算を超伝導量子ビット，イオントラップにより実現したコンピュータ，断熱型量子計算の中でも量子アニーリングを超伝導磁束量子ビットにより実現したコンピュータは物理的なハードウェアの進化とプログラミング環境の進化により商用利用が進んでいる。これらは量子ハードウェアを専門としない技術者でもクラウドを通じて容易に利用可能であることから注目を集めていることを踏まえ，上記の量子回路型コンピュータと量子アニーリング型コンピュータを指してそれぞれ量子ゲート型と量子アニーリング型という名称で分類し対比することもある [151, 第 2 章, p.11]。

アナログ量子シミュレータ 近年，中性原子や光格子を用いた様々な実装が急速に進展している計算フレームワークである。人工的な量子系を用いて別の量子系をシミュレーションするコンピュータの総称であり [172]，古典コンピュータを用いて量子回路や量子アニーリングの出力をシミュレーションする技術とは異なる。

規模と性能による分類 物理的な実現方法・計算モデルによる分類以外に，規模と性能による分類も提案されている。NISQ は Noisy Intermediate-Scale Quantum の略で，2018 年に Preskill[69] により提案された概念である。NISQ デバイスは搭載される物理量子 bit が数十から数百程度で，実行時のノイズが大きい量子デバイスを指す。量子誤り訂正や大規模な計算を行うには不十分な性能とされる。2025 年現在，全ての量子コンピュータは NISQ デバイスであると考えられる。

FTQC は Fault-Tolerant Quantum Computation の略で Shor[122] により提案された概念である。ノイズやデコヒーレンスのある量子デバイスでもその影響を量子誤り訂正等を用いて低減することで，大規模かつ長時間の計算が可能となる理論を指す。そのような計算を実現するデバイスは FTQC デバイスと呼ばれ，必然的に多くの論理量子 bit による非常に低いエラーレートでの量子計算を可能とする。実用的な暗号方式に用いられる大きさの素因数分解問題，離散対数問題の計算を行うためにはこの規模のコンピュータが必要と考えられている。

実際には FTQC デバイスが実現される前でもある程度の性能の量子エラー訂正を用いることで有用な計算が可能となると考えられており，NISQ と FTQC の中間的な性能のデバイスを指す様々な概念が提案されている。汎用的なものでは early-FTQC[174] という，数万物理量子 bit 程度の規模を持つ量子デバイスの概念があるが，特に暗号に関係する概念として CRQC（Cryptographically Relevant Quantum Computer）があり，古典コンピュータでは解くことが困難な暗号学的問題を解くことのできる量子コンピュータとして定義されている [8]。

1.1.2 ハードウェアの進展とロードマップ

前節の量子コンピュータの分類を踏まえ現在の量子コンピュータの開発状況と各組織のロードマップを概観する。量子コンピュータの開発は世界中で進められており，網羅的な記述を行うことは本報告書の目的ではない。近年開発された量子コンピュータの中で特筆すべき性能を持つもの，暗号に関係する応用に用いられたもの，日本国内で開発されたものに絞り紹介を行う。

以下では，物理量子 bit は搭載されている物理的な量子ビットの数を表し，論理量子 bit は量子誤り訂正などを行った後の論理レベルでの量子 bit 数を表すものとする。

量子回路型コンピュータの開発は米国の民間企業を中心に 2010 年代以降急速に発展しており，特に超伝導量子ビットによる実装と中性原子による実装が 1000 物理量子 bit を超えるプロセッサを実現している。しかしながら，調査の範囲で確認された量子コンピュータは総じて NISQ デバイスに留まっており，CRQC レベルのものは確認されていない。一方で，数年前までは誤り訂正処理を行うことで逆にノイズが蓄積しエラーレートが悪化する状態であったものが，2023 年には誤り訂正後のエラーレートの方が下がるという結果が報告されており [124, 95, 139, 38]，FTQC に向け安定な論理量子ビットの構築が進んでいる。また，近年では多くのコンピュータが実験室レベルではなく，商用として開発されクラウドサービスを通じて公開されている [64, 15, 129] ことも特徴である。

世界的に FTQC の開発を目指して研究が進められており，大きな枠組みでは例えば以下の目標が掲げられている。2020 年 1 月に決定された日本のムーンショット目標 6 では，様々な実装による量子コンピュータの開発を行い，2050 年までの FTQC 実現を目指している [153]。欧州の European Quantum Flagship が 2024 年 2 月に公表したロードマップでは，2020 年代後半に様々な実装での 1000 物理量子 bit，2030 年までに 99% 以上の忠実度[*6]をもつ 1000 論理量子 bit デバイスの実現を目標として掲げている [51]。

なお，量子コンピュータの性能を十分に引き出す強力なアルゴリズムを実現するためには量子 bit 数の増加のみではなく，ゲート操作の忠実度の向上，コヒーレント時間の向上などの課題を克服し，量子誤り訂正，量子ランダムアクセスメモリ等の 2025 年現在では完全には実用化されていない技術を用いる必要がある。それらの開発スピードの予測困難性が，量子コンピュータが暗号に与える影響の将来予測を困難なものとしている。

以下，各実装方式ごとの開発状況とロードマップを概観する。

超伝導量子ビット　超伝導量子ビットによる量子コンピュータは集積可能性と設計自由度が高いこと [155]，十分な品質の量子ビットが比較的安定に実現できること [173] から，大規模化に向けた開発が他の方式よりも先行した。特に 2010 年代から IBM のクラウドサービスによる一般公開があり。代表的な量子コンピュータ方式として認識されている。近年でも 2019 年 10 月に Google が 53 物理量子 bit の量子プロセッサ Sycamore を開発し量子超越性を宣言した事 [52]，2022 年 12 月に中国のチームが 110 物理量子 bit の量子プロセッサ [136] を用いて素因数分解アルゴリズムの実験を行った事 [137] 等多くの話題がある。

量子 bit 数の多いプロセッサでは, IBM が 2023 年 12 月に開発した 1121 物理量子 bit のプロセッサ IBM Condor[55]，中国科学院の量子情報・量子技術創新研究院が 2024 年 4 月に開発した 504 物理量子 bit のプロセッサ驍鴻（Xiaohong）[78] 等が代表的である。IBM が 2023 年 12 月に発表したロードマップ [65] によれば，2029 年までに実行可能ゲート数を 1 億に増やし，2033 年には実行可能ゲート数 10 億，論理量子 bit 数を 1000 まで上げるとしている。

量子誤り訂正に関する話題では 2022 年に訂正後のエラーレートが訂正前よりも下がるブレークイーブンポイントを達成したとの報告が米国，中国それぞれの研究チームから行われている [124, 95]. その後，2024 年 8 月には Google のチームが表面符号を用いた量子メモリの実装実験 [38] を行い，誤り訂正後のエラーレートが訂正前よりも下がり，構成された論理量子 bit の寿命が物理量子 bit の寿命よりも 2 倍程度長いと報告している。

日本国内では 2023 年 3 月に富士通と理研を中心としたチームが 64 物理量子 bit の量子コンピュータ叡を開発，現在までに 3 台がリリースされクラウドを通じて利用されている [165, 176]。理研の次の目標は 144 物理量子 bit デバイスの実現であるとしている [166]。富士通では 2024 年 5 月にロードマップ [154] を公開し，2025 年中に 256 物理量子 bit を実現し，2026 年度以降に 1000 物理量子 bit を達成するとしている。

また，ムーンショット目標 6 の中のプロジェクト「スケーラブルな高集積量子誤り訂正システムの開発」においても 2030 年までに超伝導量子ビットを用いた 100 万物理量子 bit の FTQC を，2025 年に 100 物理量子 bit を用いた 1 論理量子 bit のプロトタイプを作成することをマイルストーンとしている [150, 2:30]。

イオントラップ　イオントラップ方式の利点はコヒーレント時間の長さと量子操作時のエラー率の低さである [145]。

[*6] ここでは量子ゲート操作の忠実度（gate fidelity）を指す。厳密な定義は決まっていないものの，大まかに量子デバイスの出力が理想的な計算結果とどの程度一致しているかを測る指標である。

同じ量子 bit 数を持つ超電導量子ビットのコンピュータと比べると量子体積[*7] などの面で優位性がある [111]。2020 年 10 月に IonQ が 32 物理量子 bit[36]，2024 年 6 月には Quantinuum が 56 物理量子 bit[112] のデバイスを発表している。Quantinuum と Microsoft により，量子誤り訂正後のエラーレートが訂正前よりも 800 倍程度下がったという実験報告が行われている [139]。

中性原子 中性原子（冷却原子）を用いた量子コンピュータはコヒーレント時間が長く，スケーラビリティが超伝導量子ビットやイオントラップよりも良いとされる。また，動的光ピンセット技術により量子ビット間の全結合が可能であるともされている [178]。開発は Atom computing による 1180 物理量子 bit の量子プロセッサ [40] を筆頭にここ数年で急速に進展している。また，ハーバード大学らのチームが 280 物理量子 bit を用いて量子誤り訂正を行った 48 論理量子 bit の構築とベンチマーク [23] を行い，訂正後のゲート操作の忠実度が上がることを確認している。日本の自然科学研究機構分子科学研究所ではほぼ理論限界に近い 6.5 ns でのゲート操作速度を可能とする 400 物理量子 bit のシステムが構成されており [175, 8:45]，2030 年の事業化を目指している [159]。

シリコン量子ビット式 この方式はシリコン中の電子スピンを用いるため，既存の半導体製造技術を応用可能であると見込まれている [163]。また，動作温度が 10K 程度と超伝導量子ビットど比べて高いため冷却器の小型化が可能である点なども利点として挙げられている。

Intel は 2010 年代から研究を進め，2023 年 6 月に 12 物理量子 bit のプロセッサ Tunnel Falls を発表 [67] した。また，日立製作所 [156, 160] と理研 [164] が実用化に向けて研究を進めている。日本のムーンショット目標 6 では，2040 年までに 10 万〜100 万物理量子 bit による誤り訂正の実証を行い，2050 年までに 100 万以上の物理量子 bit からなる FTQC デバイスの実現を目指すとしている [162, p.2]。

光量子式 光子を用いることから室温や大気中での動作が可能で，装置の小型化が見込まれている [144]。2022 年 6 月にカナダの Xanadu 社が 216 量子 bit を搭載した Borealis を発表した [76]。2024 年 11 月には理研などが計算プラットフォームを開発し [158]，2030 年までに光ファイバ型の連続量光量子コンピュータの実現を目指すとしている [144]。

量子アニーリングマシン D-Wave が 2010 年代から超伝導磁束量子ビット型アニーリングマシンを商用に発表している。2020 年 9 月にリリースされた D-Wave Advantage は約 5000 物理量子 bit を搭載していた [82]。2024 年に発表された D-Wave Advantage 2 では量子ビット同士の結合数とノイズが改良されており，2024 年 6 月時点でのプロトタイプは約 1200 物理量子 bit を搭載 [45]，将来的に 7000 物理量子 bit 規模のアニーリングマシンを提供する予定であるとしている [44]。

日本国内では産総研が組み合わせ最適化問題に特化した超伝導磁束量子ビット型の 6 物理量子 bit 量子アニーリングマシンを開発 [167]。また，NEC と東北大学が中心となり，超伝導パラメトロンを用いた 8 物理量子 bit の量子アニーリングマシンが開発されている [152]。

1.2 耐量子計算機暗号（PQC） の必要性について

本節では，量子コンピュータによる現代暗号への影響と PQC の必要性についてまとめる。2024 年 9 月現在，以下に述べる素因数分解問題，離散対数問題を解く実験の他，検証用に構成した小規模な共通鍵暗号の鍵復元実験 [107]，耐量子計算機暗号の計算問題を解く実験 [114] が報告されている。調査の範囲では既存の量子コンピュータの性能が古典コンピュータの暗号解読性能を超えたという報告，および実社会で用いられている大きさのパラメータを持つ暗号方式が解かれたという報告は無く，現代暗号に対する量子コンピュータの直接的な脅威は現時点では生じていないと考えられる。

[*7] NISQ デバイスの性能を計る指標で，2018 年に IBM の研究者により提案された [43]。ベンチマーク用量子回路をデバイスで実行した測定結果と理想的な実行結果を比較し，量子ノイズの影響により結果が理想から大きくずれないような最大の量子回路サイズ（量子ビット数 × 回路深さ）を性能指標とする。

一方で，各機関が発表しているロードマップが予定通りに達成されると仮定すると，今後数十年で RSA，ECDSA をはじめとする素因数分解問題や離散対数問題の計算困難性に基づいた暗号の解読を可能とする規模の量子計算を実行可能な量子コンピュータが開発される。暗号方式の提案から社会的な普及までは RSA 暗号・楕円曲線暗号で 20 年ほどの期間が必要とされたことから，PQC の場合でも同程度の期間が必要と想定されるため，長期間の移行スケジュールを策定し，準備を行う必要がある。

1.2.1 量子コンピュータの影響による現代暗号の危殆化予測

Shor による素因数分解問題と離散対数問題の量子多項式時間アルゴリズム [123] が発表されて以降，数千 bits の RSA 暗号を危殆化させる量子コンピュータの規模の見積もり [56, 58, 171, 57]，実現時期の予測 [30, 18, 79, 90, 91] に関する研究が進められている。2025 年現在，直近の数年間で実用的な RSA 暗号，ECDSA 署名を攻撃可能な古典または量子コンピュータが開発される可能性は極めて低いと考えられる。一方で，各国の標準機関は長期的には新たな暗号方式に移行するプロジェクトを進めている。例えば NIST は 2024 年 11 月に公表した NIST IR 8547 （initial public draft）[88, Sec. 4.1] において 2048，3072bits RSA と，224，256bits ECDSA，EdDSA を 2035 年までに段階的に廃止，利用禁止にするとしている。

暗号解読のモデルと数値化方法 Shor のアルゴリズムが量子回路を用いて表現されていること，量子回路型計算を行う量子コンピュータの開発が他の方式よりも先行していることから，現代暗号の解読量子計算量の見積り [56, 58, 171, 57]，および量子コンピュータ性能の将来予測を通じた共通鍵暗号，耐量子計算機暗号のパラメータ設定は量子回路モデルを用いて数値化 [101, p.18] がされている。量子計算量理論の観点からは 1.1.1 節冒頭に挙げた量子計算モデルがいずれも多項式倍の差を除いて等価であることが示されており，量子回路型計算で困難であると予想される計算問題が断熱量子計算（量子アニーリング）のような他の既知のモデルを用いた量子計算により危殆化する可能性は小さいと考えられる。

なお，2048bits の合成数を公開鍵に用いた RSA 暗号（以下，RSA-2048 と表記）は古典で 112-bit 安全性を持つとされており [140]，暗号に影響のある量子コンピュータの開発が仮に実現しなかった場合でも，古典コンピュータの性能の伸びにより長期的には危殆化すると考えられている。このことから，将来的な鍵長の変更もしくは新たな暗号方式への移行は量子コンピュータの大規模化とは独立した課題として準備を進める必要があることは長年議論されてきた [140, 21, 120] ことを指摘しておく。

危殆化時期の予測 RSA-2048 に対する量子コンピュータの影響とその危殆化時期に関して，様々な予測が存在する。定量的な予測に基づいたものでは 2039 年以降 [30]，2050 年前後 [18] と少なくとも 20 年程度は実現に時間がかかるとされている。

セキュリティ・量子分野の専門家の予測では，Mariantoni が PQCrypto2014 の招待講演 [79] で調査に 5 年，開発に 10 年程度で 15 年後（2029 年前後）としたもの，Mosca が Workshop on Cybersecurity in a Post-Quantum World（2015 年）で 2026 から 2031 年 [90] と予測したものが有名である。近年では国際会議 RSA conference 2023 内で開かれた暗号専門家によるパネルディスカッションの中で，Shamir が RSA，DH，ECDH に影響を及ぼす量子コンピュータがあと 30 年から 40 年で開発される可能性があると発言している [128]。

個人ではなく，多くの専門家へのアンケートを集計した結果が 2019 年から毎年 Global Risk Institute により Quantum Threat Timeline として発行されている。2023 年に行われたアンケートを基にした予測レポート [91] では 24 時間で RSA-2048 を解読可能な量子コンピュータが 15 年以内に出現する可能性が 33% から 54% 程度であると分析している。この調査結果を引用するレポートは多く，例えば金融におけるサイバーリスクを取り扱う国際的なコンソーシアム Financial Services Information Sharing and Analysis Center の 2023 年報告書 [60] では危殆化時期をあと 10 年から 30 年，米国の金融サービス標準を決定する ASC X9 の 2022 年度版報告書 [59] では CRQC の登場時期

について “There is no consensus on this issue” としながらも，危殆化時期をあと 5 年から 30 年としている。

日本国内の専門家へのアンケート調査では，2019 年に行われた文部科学省科学技術・学術政策研究所（NISTEP）による科学技術予測調査 [169, p. （II-4）48, 52] がある。この中ではある程度コヒーレンス時間の長い数百物理量子 bit 規模の量子回路コンピュータの登場を 2033 年頃と予測しているため，現代暗号に対して脅威となる量子コンピュータが出現するのはそれ以降と解釈できる。ムーンショット目標 6 では 2050 年頃までに FTQC を実現するとしている [153] ことから，予測が実現されるのであれば現代暗号の量子コンピュータによる危殆化もその付近で起こると考えられる。

1.2.2 量子コンピュータによる素因数分解・離散対数問題計算の現状

将来的に RSA，ECDSA が危殆化すると考える専門家が多数存在する一方で，量子コンピュータ実機を用いた素因数分解問題及び離散対数計算の実験は小規模なものに留まっている。本節では，量子コンピュータを用いた Shor のアルゴリズムに関する実験，その他の素因数分解アルゴリズムに関する実験，および関連する理論的な成果についてまとめる。

なお，量子回路および量子アニーリングの古典計算機によるシミュレーションを用いた素因数分解の実験報告が多く存在する [135, 171] が，本報告書では省略し量子的な現象を用いた計算機による報告のみを取り上げる。

Shor のアルゴリズムの実機実験 量子回路型コンピュータ実機を用いた実験は，CRYPTREC 外部調査報告書「Shor のアルゴリズム実装動向調査」[157] に挙げられているもの及びその後の [127, 125, 134] を含めて 15，21，35 の素因数分解実験および離散対数問題 $2^z \equiv 1 \pmod 3$ の離散対数の計算実験を行ったもののみしか知られていない。

[131, 132] をはじめとする Shor のアルゴリズムを用いた初期の報告は $N = 15$ の素因数分解回路の量子フーリエ変換部分を除いた部分回路を実装する予備実験的なもの，位数や N の情報を用いて過度な簡略化を行ったものが多かった。2019 年には指数計算部分が簡略化されているものの，量子フーリエ変換部分と組み合わせた回路に対して IBM Quantum による $N = 15, 21, 35$ の実験報告 [83] が行われている。また，離散対数問題の実装実験報告 [18] が出版されるなど，実際に問題のインスタンスサイズには表れない量子回路規模の拡大は着実に続いていると考えられる。

Shor のアルゴリズムに関する理論の進展 1.1.2 節に紹介した量子コンピュータの性能進化がターゲットとなる数の目に見える伸びに繋がらない理由が量子コンピュータ実機の性能と Shor のアルゴリズムの性質双方の観点から検証され，明らかになりつつある。

Ichikawa らによる量子コンピュータ実機実験に関するサーベイ論文 [66] によると，2016 年から 2022 年の間に出版された 748 件の実験報告で用いられた実際の量子 bit 数の中央値が 5 から 6 に増えたのみでありほぼ横ばいである。これは 2022 年に 433 物理量子 bit を搭載した IBM Quantum Osprey [39] が発表されていたこととは対照的である。量子ノイズ，デコヒーレンス等の影響により，デバイスに搭載されている物理量子 bit 数と，実際に安定して動作し測定可能な物理量子 bit 数の間には大きな差がある。

また，2024 年には Cai により Shor のアルゴリズムが量子ノイズに弱い事の理論的な証明 [31] が与えられている。この現象は経験的には知られていたが [18, Sec. VI]，より大きな規模で Shor のアルゴリズムを実行するためには量子ビット数の増加だけではなく，量子ノイズの影響を下げる必要があることが理論的に示された形となる。

以上をまとめると，より大きな数の素因数分解を Shor のアルゴリズムを用いて行うためには十分にノイズが小さく，安定に動作する量子ビットを搭載した量子コンピュータが必要であると考えられる。

また，Shor のアルゴリズムを用いて現在より大きな数の素因数分解を行うためには，これまでの実験のように入力インスタンスに合わせ簡略化した量子回路ではなく，汎用の剰余加算・乗算回路による構成を行う必要がある。しかしながら，$N = 15$ の素因数分解を行うために 4 ビットの汎用剰余加算・乗算回路を用いて Shor のアルゴリズムを構成すると，ゲート数約 13,000，深さ約 10,500 という現在の量子コンピュータでは実行不可能な規模となるという山口ら

の評価 [171] があるため，簡略化されていない回路の実行は現在のところ困難であると考えられる。

一方，2023 年 8 月に Regev[115] により Shor のアルゴリズムよりも量子 bit 数を増やす代わりに量子ゲート数の少ないアルゴリズムが提案された。多くのフォロー論文 [113, 48] が発表されているものの，量子コンピュータ実機を用いた実験は確認されていない。

別の理論的な可能性として，量子アニーリングと一般的な断熱量子計算の中間にあるストカスティック断熱量子計算[*8]による Shor のアルゴリズムのシミュレーションがある。これはハミルトニアンを基底に関して非正定値の非対角実行列に制限した断熱量子計算のクラス [14, Sect. VI] であり，計算量理論ではクラス StoqAQC と名付けられている。このクラスの能力を持つコンピュータは，Shor のアルゴリズムを効率的にシミュレート可能であることが知られており [53, 54]，ストカスティック断熱量子計算を実行可能な量子コンピュータが出現するかどうかが安全性に関わると考えられる。

量子アニーリングによる実験 Shor のアルゴリズム以外の素因数分解の計算手法のうち代表的なものとして，2 進数乗算の筆算形式で式展開したものを，組み合わせ最適化問題（Quadratic Unconstrained Binary Optimization: QUBO）として定式化したものがある。QUBO とイジングモデルは自明な変換が知られていることから，量子アニーリングを中心とした断熱量子計算を用いた実験が多数報告されている。

2000 年代後半の初期の実験 [106] ではハミルトニアンに合わせて有機化合物を合成し，最適化問題の変数に対応する原子のスピンを核磁気共鳴（Nuclear Magnetic Resonance: NMR）を用いた分析により結果を取り出すという手法で計算を行っていたためスケーリングが困難であったが，D-Wave の量子アニーリングマシンがオンライン上で比較的手軽に利用可能になって以降は実験報告が相次いでいる [170, 135]。素因数分解のターゲットとなる数は着実に大型化しており，現時点での最大は 2023 年に D-wave Advantage 4.1 を用いた 23 ビットの 8219999=32749×251 [47] であり，最適化問題の表現方法，変数の省略など多くの技術を使い Pegasus トポロジーで接続された量子ビットの性能を引き出している。

その他の素因数分解手法 Shor のアルゴリズム，量子アニーリング以外の手法でも様々な素因数分解の実験が行われている。アニーリングと同様の QUBO を Quantum Approximate Optimization Algorithm （QAOA）を用いて解く実験 [110]（143，291311 を分解），Variational Quantum Eigensolver （VQE）を用いて解く実験 [126]（251 を分解），Digitized adiabatic quantum computation を用いて解く実験 [62]（2479 を分解）の報告がある。これらの実験はいずれも IBM Quantum を用いて行われた。

量子回路型コンピュータ上で QAOA を用いた素因数分解問題へのアプローチとして，Schnorr アルゴリズム [119] の部分的な量子化の研究が存在する。Schnorr アルゴリズムは数体篩法の関係探索を係数制限付きの近似最近ベクトル問題に変換して行うが，[137] ではこれをさらに最適化問題に落とし込み，QAOA を 10 量子 bit 回路上で実行することで 48bits の数の素因数分解実験結果を報告している。

また，中性原子による実装の一種として，リュードベリ原子によるアナログ量子シミュレータを用いたグラフの最大独立集合問題を解くための枠組みが整理されており，これを用いた素因数分解の実験も行われている。[105] では 6,15,35 の素因数分解のインスタンスを SAT を経由して最大独立集合問題に変換して実験を行っている。

1.3 PQC の研究及び標準化等に関する動向

現在 PQC として扱われている暗号のほとんどは 1994 年に Shor のアルゴリズムが発表される以前から効率性および理論的側面から研究が行われており [81, 75, 80]，2000 年代以降に耐量子計算機性が注目されたものである。なお，Post-Quantum Cryptography の用語自体は 2004 年の論文 Post-Quantum Signatures [29] が初出であり，p.1 に Bernstein の造語であることが明記されている。

[*8] ストカスティック（stoquastic）は stochastic と quantum の合成語で，2006 年に Bravyi らの論文 [28] で導入された単語される。

現在，PQC に関する研究成果は暗号の国際会議で主に発表されている。特に Crypto，Eurocrypt，Asiacrypt 等の暗号全般を扱う会議で取り扱われることも多いが，その他 PQC を専門に扱う国際会議として PQCrypto が 2006 年から開催され，2024 年までに 15 回が開催されている。

以下，各国における標準化の動向を述べる。米国 NIST は PQC の標準化活動を初期から大規模に行っており世界への影響力が大きいため，まず米国の状況について述べてその後に各国の状況について述べる。

1.3.1 米国 NIST における標準化の動向

2015 年 8 月国家安全保障局（NSA）が PQC への移行計画 [7] を発表したことを受け，標準化活動が国立標準技術研究所（NIST）により開始された。2016 年 2 月には福岡で開催された国際会議 PQCrypto 2016 において NIST の Moody により NIST PQC　標準化プロジェクトに関する講演 [86] が行われ，選定基準に関する意見募集を経て 12 月に Call for Proposals [101] が正式公開された。

2017 年 11 月 30 日の公募締め切りまでに世界中から耐量子計算機暗号の候補 82 方式が提案され，公募条件を満たした 69 方式が標準化プロジェクト第 1 ラウンド候補として公開されたが，5 方式は公開後に取り下げられている。締め切り直後からメーリングリスト pqc-forum [109] 上では世界中の暗号研究者，暗号方式の提案者らを交えて理論的な脆弱性から実装リファレンスコードの軽微なバグの指摘に至るまで多彩な議論が行われた。このときの議論の概要は [12, 108] 等にまとめられている。

2019 年 1 月 30 日には，第 2 ラウンドへ進む 26 方式が発表され，その後 2020 年 7 月 22 日には，第 3 ラウンドへ進む Finalists の 7 方式と Alternate Candidates の 8 方式が発表された [89]。両者の違いは Finalists が第 3 ラウンドの終了時に標準化方式となるかどうかが判断されるもの，Alternate Candidates が標準化方式の候補ではあるものの第 3 ラウンドの終了時点では判断が行われない可能性が高いものとされていた。しかし実際には Finalists であった Classic McEliece が第 4 ラウンド候補として判断を保留された一方で，Alternate Candidates であった SPHINCS$^+$ がそのまま標準化方式として選ばれている。

2022 年 7 月 5 日に NIST から標準化方式として公開鍵暗号 1 方式と署名 3 方式が発表された [13]。上記 4 方式のうち，格子に基づく公開鍵暗号方式 CRYSTALS-Kyber は FIPS 203 （ML-KEM）[98] として，格子に基づく署名方式 CRYSTALS-Dilithium は FIPS 204 （ML-DSA）[97] として，ハッシュ関数に基づく署名方式 SPHINCS$^+$ は FIPS 205 （SLH-DSA）[100] として 2024 年 8 月にそれぞれ標準化されている。また，格子に基づく署名方式 FALCON についてもアルゴリズムの微修正を経た後に FIPS 206 （FN-DSA）として標準化される予定である [25]。

標準化の 4 方式が決定されると同時に，第 3 ラウンド候補の中から第 4 ラウンドへと進む公開鍵暗号方式の 4 方式が発表され，さらに追加の電子署名方式が再公募されることが周知された [118]。第 4 ラウンドに進んだ 4 方式のうち，BIKE，Classic McEliece，HQC の 3 方式が符号に基づく公開鍵暗号方式，SIKE が同種写像に基づく公開鍵暗号方式であった。その後，2022 年 8 月に SIKE に対する古典多項式時間による鍵復元攻撃が発表され [32]，致命的であることが確認されたことから提案チームにより候補から取り下げられた。2024 年 4 月に開催された第 5 回 PQC Standardization Conference における Moody の講演によると，NIST は 2024 年末までに残った第 4 ラウンド候補の中から数件を標準化に選ぶとしている [85, p.12]。

署名方式の追加公募　上記第 4 ラウンドの発表と並行して，NIST は 2022 年 9 月から正式に追加の NIST PQC 標準化プロジェクト追加署名（Additional Digital Signature Schemes）の募集を開始した。締切の 2023 年 6 月 1 日までに 50 方式の応募があり，翌 7 月に公募条件を満たした 40 方式が発表された。公募の事前情報として，2022 年 7 月に pqc-forum に投稿された文書 [84] では NIST が署名長と検証時間の小さい方式を求めているとし，一例として多変数多項式に基づく署名方式の一種である UOV 方式が挙げられている。また，Module 格子のような構造化格子に基づく署名方式は既に CRYSTALS-Dilithium と FALCON が標準化に決まっていることから，構造化格子に基づく手法以外が望ましいとしており，後半の内容は募集要項にも明記された [99, p.2]。結果として格子に基づく署名は 7 方式，

UOV 型の多変数多項式に基づく署名では 7 方式の応募があった。

2022 年の署名方式公募後，NIST は選考の第 1 ラウンド候補を分類ごとに発表したが，その中に 2016 年の標準化には存在しないカテゴリ MPC-in-the-Head が新たに登場している。これは，マルチパーティ計算から構成したゼロ知識証明プロトコルに Fiat-Shamir 変換を適用することで署名方式を得る構成フレームワークであり，格子，符号のように安全性の根拠となる計算問題の種類を表すものではない。MPC-in-the-Head に分類されているそれぞれの方式の安全性は実際には符号問題，多変数方程式問題，共通鍵暗号方式の平文復元問題の困難性などに帰着されている。

2024 年 10 月には第 2 ラウンドの候補となる 14 方式 [117] が発表された。格子に基づく署名方式は格子同型性判定問題を安全性の根拠とした HAWK の 1 方式，多変数多項式に基づく UOV 型署名が 4 方式，MPC-in-the-Head 型の構成を行った署名が 5 方式であった。選定に関わるレポートは [11] で公開されている。

PQC への移行 2015 年に Mosca の提案した暗号の危殆化に関わる不等式 [90] では，X（情報を保護する期間）$+Y$（システム移行期間）と Z（CRQC 開発までの期間）の大小関係によりシステム移行の準備期間を設定する必要があるとしている。一方で，暗号化データを保存し，将来的にコンピュータの性能が上がってから解読するハーベスト攻撃（2.2.2 節も参照）を想定すると，CRQC 開発までの年数によらず，現在の暗号利用にはリスクがあるとも考えられている（例えば [88, Sec. 1] を参照。）以上の背景のもと，2022 年 5 月公表された国家安全保障覚書 NSM-10[92] では 2035 年を目処に暗号システムを PQC に移行することを目標としている。同様に，2022 年 9 月に発表された商用国家安全保障アルゴリズムのリスト 2.0[6] では 2035 年までにシステムに耐量子計算機性をもたせることを目標としたタイムラインを掲載している。

現在使われている暗号から PQC への移行を推進するため，NIST 内の NCCoE（National Cybersecurity Center of Excellence）を中心にコンソーシアムが設立された [50]。組織における暗号のユースケース，相互運用性やリスク評価を含めた移行計画の策定に関する包括的な技術文書が NIST SP 1800-38A から 38C として発行される予定であり，現在は Initial Preliminary Draft[94] が公開されている。

安全性レベル NIST PQC 標準化プロジェクトにおいて，暗号方式の安全性はレベル 1 から 5 で定義されており，提案者は応募時にパラメータと達成される安全性レベルを示す必要があった。レベル 1, 3, 5 はそれぞれ AES128, AES192, AES256 などの 128, 192, 256bits の秘密鍵を持つブロック暗号の鍵復元の困難性と同等かそれ以上の計算量であり，レベル 2 と 4 はそれぞれ SHA256/SHA3-256 と SHA384/SHA3-384 などの 256bits と 384bits の暗号学的ハッシュ関数の衝突探索の困難性と同等かそれ以上の古典もしくは量子計算量とされている。レベル 1 から 5 の具体的な計算量は表 1.1 で与えられる [99]。古典コンピュータによる攻撃者に対しては古典論理回路のゲート数が，量子コンピュータを利用可能な攻撃者に対しては量子回路のゲート数と最大深さの積が与えられている。計算量評価において，公開鍵暗号方式では，IND-CCA2 安全性を考える際には 2^{64} 個以下の選択暗号文を復号オラクルに古典的にクエリできるとし，署名方式では，EUF-CMA 安全性を考える際には 2^{64} 個以下のメッセージを署名オラクルに古典的にクエリできるとしている。

また，レベル 1,3,5 の量子回路計算量は 2022 年度版調査報告表の中で $2^{157}, 2^{221}, 2^{285}$ とされている部分は 2016 年版では $2^{170}, 2^{233}, 2^{298}$ であった。つまり，2016 年の PQC 候補でレベル 1,3,5 とされているものは 2022 年の定義でもレベル 1,3,5 の基準を満たすことになる。この更新は AES を解読する量子回路の改良により，量子計算量が改善されたことによる。

NIST 標準化に伴い，FIPS 203,204,205 の各文書ではセキュリティカテゴリの定義に関して NIST SP800-57 Part 1[21] を参照しているが，2024 年 9 月時点で最新の Rev. 5 において対応する定義が存在しないことが指摘されており，NIST の担当者から次期リビジョンでの修正が予告されている [87]。

表 1.1: 2022 年に公表された NIST PQC 標準化プロジェクト追加署名の Call for proposals [101] における安全性レベルと計算量の対応表。各レベルは古典，量子のどちらか一方の基準を満たすものとして定義されている。

レベル	量子回路の（最大深さ）×（ゲート数）	古典論理ゲート数
レベル 1	2^{157}	2^{143}
レベル 2	–	2^{146}
レベル 3	2^{221}	2^{207}
レベル 4	–	2^{210}
レベル 5	2^{285}	2^{272}

1.3.2 米国以外での動向

米国以外でも世界各国の機関が調査活動を行い，調査レポートの出版 [20, 93]，各国における PQC 標準方式，推奨暗号方式の選定 [6, 34, 17, 120, 33, 35, 102, 130] を進めている。国際的な機関では ISO/IEC[68]，IETF[63] 等で移行，標準化の議論が進められている。

各国の政府機関から PQC の標準暗号リスト，推奨暗号リストが公表されている。代表的なものを表 1.2 にまとめる。多くの国が NIST PQC 標準化プロジェクトに提案された暗号方式を採用しているが，FrodoKEM のように NIST PQC 標準化プロジェクトの第 3 ラウンド以降の選考に漏れた方式，Classic McEliece のように第 4 ラウンド選考中の状態で選ばれた例も存在する。また，多くの機関が NIST 標準方式の単独利用ではなく古典的安全性がよく知られている RSA や ECDSA とのハイブリッドを推奨していること，レベル 3 以上のパラメータ利用を推奨していることも特徴的である。国家による標準暗号以外でも Streamlined NTRU Prime[22] のように OpenSSH の実装 [104] を通じて実用化されている方式も存在する。

韓国では量子耐性暗号研究団の主催する KpqC プロジェクト [177] が耐量子計算機性を持つ公開鍵暗号方式と署名方式の公募を 2022 年に開始している。2022 年 10 月の締切までに公開鍵暗号方式・鍵共有が 8 方式，署名方式が 9 方式応募された。2022 年 11 月に第 1 ラウンドを開始，2023 年 12 月に第 2 ラウンドの選考が開始され，2025 年 1 月に共通鍵暗号に基づく MPC-in-the-Head パラダイムの署名方式 AIMer，および格子に基づく公開鍵暗号方式 NTRU+，署名方式 HAETAE，鍵交換方式 SMAUG-T の 4 方式が最終方式として選ばれたことがアナウンスされた。

中国では中国暗号学会（CACR）が中心となり PQC の公募を行っている [142]。2018 年 6 月の募集要項に従い 2019 年 2 月の締切までに公開鍵暗号 38 方式と共通鍵暗号 22 方式が応募されている。2019 年 9 月の第 2 ラウンドの時点で公開鍵暗号 14 方式と共通鍵暗号 10 方式に絞られ，最終的には 2020 年 1 月に一等，二等，三等としてランク付けが発表された [143]。一等として公開鍵暗号方式 LAC.PKE，Aigis-enc，署名方式 Aigis-sig，共通鍵暗号方式 uBlock，Ballet が挙げられている。

日本では CRYPTREC の暗号技術調査ワーキンググループにおいて 2014 年度に PQC の代表的な候補である格子に基づく暗号技術について調査を行い，報告書「格子問題等の困難性に関する調査」を公開している [1]。さらに 2017 年度から 2018 年度にかけて，PQC の代表的な候補である 4 種類の分類（格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術，同種写像に基づく暗号技術）について調査し，報告書にまとめた [2]。また，2021 年度から 2022 年度にかけて，上記 4 種類に加えてハッシュ関数に基づく署名技術を加えた 5 種類の技術について調査を行い，報告書 [3] およびガイドライン [4] としてまとめている。

表 1.2: 世界各国の標準暗号，推奨暗号リストの状況。表中の勧告，推奨，許容等はそれぞれのレポートからの翻訳であるため，厳密に同じ意味ではない。許容されているバージョン，安全性レベルなど，詳細は引用先のレポートを参照のこと。

方式の名称	NIST PQC（米）	CNSA 2.0（米）[6]	NCSC（英）[34]	ANSSI（仏）[17]	BSI（独）[120]	NCSC（蘭）[33, 35]	NÚKIB（チェコ）[102]	TRAFICOM（フィンランド）[130]
ML-KEM（CRYSTALS-Kyber）	標準化（FIPS 203 [98]）	勧告 [a]	推奨 [c]	許容 [d]	推奨 [ef]	推奨	推奨 [h]	暗号要件 [k]
FrodoKEM	Round 3	–	–	許容 [d]	推奨 [e]	許容	許容 [i]	–
Classic McEliece	Round 4	–	–	–	推奨 [e]	許容	許容 [i]	–
ML-DSA（CRYSTALS-Dilithium）	標準化（FIPS 204 [97]）	勧告 [a]	推奨 [c]	許容 [d]	推奨 [ef]	推奨/許容 [g]	推奨 [h]	暗号要件 [k]
FN-DSA（FALCON）	標準化中	–	–	許容 [d]	–	推奨/許容 [g]	推奨 [i]	–
XMSS/LMS	標準化（NIST SP 800-208）	勧告 [b]	推奨	許容 [d]	推奨	推奨/許容 [g]	推奨 [j]	–
SLH-DSA SPHINCS$^+$	標準化（FIPS 205 [100]）	–	推奨 [c]	許容 [d]	推奨 [ef]	推奨/許容 [g]	推奨 [i]	暗号要件 [k]

注釈一覧	
a	汎用量子アルゴリズムとしてレベル 5 パラメータの使用を勧告
b	ソフトウェア・ファームウェアに対する署名のための使用を勧告
c	標準化の最終文書を元に堅牢な実装がされたものの利用を推奨
d	メインストリームの PQC として適切だが，可能な限りパラメータを大きく取ること
e	古典的な安全性が確保された方式とのハイブリッドのみを推奨
f	NIST 標準化の安全性レベル 3,5 を推奨パラメータとする意向
g	緊急性のシナリオによって推奨と許容の方式が異なる
h	単独利用は安全性レベル 5 のみ，他は古典の推奨暗号とのハイブリッド利用とする
i	古典の推奨暗号とのハイブリッド利用とする
j	ファームウェア・ソフトウェアの保護目的での単独利用を推奨
k	古典の推奨暗号とのハイブリッド利用を推奨

1.4 本調査で対象とした PQC の種類

本調査報告書では PQC の調査を格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術，同種写像に基づく暗号技術，ハッシュ関数に基づく署名技術の 5 分類で行った。この分類は，安全性の根拠となる数学的な計算問題の種類に基づいて行われている。

例えば，教科書的な RSA 暗号では 2 つの異なる大きな素数 p, q と指数 d を秘密鍵，積 $N = pq$ と指数 e を公開鍵としている。鍵復元の困難性と素因数分解問題の困難性は多項式時間等価であるため [116, 41]，RSA 暗号の鍵復元の安全性は素因数分解問題の困難性に基づくものと考えることができ，素因数分解に基づく暗号に分類できる。同様に，楕円曲線暗号の場合も例えば楕円曲線上の ElGamal 暗号のように安全性が楕円曲線上の離散対数問題の困難性に基づくため，離散対数問題に基づく暗号に分類できる。

本ガイドライン・報告書で扱う代表的な 5 種類の PQC （格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術, 同種写像に基づく暗号技術，ハッシュ関数に基づく署名技術） も RSA 暗号等と同様に，暗号の安全性がそれぞれ格子問題の困難性，符号復号問題の困難性，多変数代数方程式の求解困難性，同種写像の計算困難

性，ハッシュ関数の衝突発見困難性に基づいている。そして，これらの問題を量子コンピュータを利用して効率よく解くアルゴリズムはまだ発見されていないことから，上記の暗号は PQC であると期待されている。暗号方式と数学的な計算問題の具体的な関係は各章の第 1 節に記載されている。

上記 5 種類を選んだ理由は主に研究期間の長さ，研究コミュニティの大きさによる。より細かい歴史的な背景は各章の第 4 節に記載されている。

格子に基づく暗号技術は 1997 年の Ajtai と Dwork による論文 [10] から 25 年以上の歴史を持ち，解読技術である格子アルゴリズムに関しても 50 年の歴史を持つ [46, 72, 77]。符号に基づく暗号技術は McEliece による 1978 年の論文 [81] から 40 年以上の歴史を持ち，解読技術は通信における符号の復号技術であり符号理論として 70 年以上研究が行われている。多変数多項式に基づく暗号技術は Ong と Schnorr による 1983 年の論文 [103] を源流[*9]とし，1988 年の松本-今井暗号 [80] を経て 40 年以上の研究が続けられている。同種写像の計算問題に基づく暗号技術も Couveignes による 1997 年の提案 [42] から 25 年以上研究が続けられている。ハッシュ関数に基づく署名方式は Lamport による 1979 年の論文 [75] から 40 年以上の研究が行われている。

1.5 耐量子計算機暗号調査報告書執筆者リスト

主査	國廣 昇	筑波大学
委員	青木 和麻呂	文教大学
委員	伊藤 忠彦	セコム株式会社
委員	下山 武司	国立情報学研究所
委員	高木 剛	東京大学
委員	高島 克幸	早稲田大学
委員	成定　真太郎	KDDI 総合研究所
委員	廣瀬 勝一	福井大学
委員	安田 貴徳	岡山理科大学
委員	安田 雅哉	立教大学
事務局	青野 良範	情報通信研究機構
事務局	五十部 孝典	情報通信研究機構
事務局	伊藤 竜馬	情報通信研究機構
事務局	大久保 美也子	情報通信研究機構
事務局	大東 俊博	情報通信研究機構
事務局	小川 一人	情報通信研究機構
事務局	金森 祥子	情報通信研究機構
事務局	黒川 貴司	情報通信研究機構
事務局	高安 敦	情報通信研究機構
事務局	横山 和弘	情報通信研究機構
事務局	吉田 真紀	情報通信研究機構
事務局	篠原 直行	情報通信研究機構

[*9] ただし Ong と Schnorr による方式の安全性は素因数分解問題に基づくため耐量子計算機性を持たないことに注意。

第 1 章の参照文献

[1] CRYPTREC 暗号技術調査 WG (暗号解析評価). 格子問題等の困難性に関する調査. CRYPTREC EX-2404-2014, https://www.cryptrec.go.jp/exreport/cryptrec-ex-2404-2014.pdf. 2015-03.

[2] CRYPTREC 暗号技術調査 WG (暗号解析評価). 耐量子計算機暗号の研究動向調査報告書. CRYPTREC TR-2001-2018, https://www.cryptrec.go.jp/report/cryptrec-tr-2001-2018.pdf. 2019-04.

[3] CRYPTREC 暗号技術調査 WG (耐量子計算機暗号). CRYPTREC 耐量子計算機暗号の研究動向調査報告書. CRYPTREC GL-2004-2022, https://www.cryptrec.go.jp/report/cryptrec-tr-2001-2022.pdf. 2023-03.

[4] CRYPTREC 暗号技術調査 WG (耐量子計算機暗号). CRYPTREC 暗号技術ガイドライン (耐量子計算機暗号). CRYPTREC GL-2004-2022, https://www.cryptrec.go.jp/report/cryptrec-gl-2004-2022.pdf. 2023-03.

[5] CRYPTREC 暗号技術調査 WG (高機能暗号). CRYPTREC 暗号技術ガイドライン (高機能暗号). CRYPTREC GL-2005-2022, https://www.cryptrec.go.jp/report/cryptrec-gl-2005-2022.pdf. 2023-03.

[6] National Security Agency. Announcing the Commercial National Security Algorithm Suite 2.0. https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF. 2022-09. (2024-12-06 閲覧).

[7] National Security Agency. Cryptography Today. https://web.archive.org/web/20150815072948/https://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml. 2015-08. (2024-12-05 Internet Archive 版を確認).

[8] National Security Agency. Frequently Asked Questions, Quantum Computing and Post-Quantum Cryptography. https://media.defense.gov/2021/Aug/04/2002821837/-1/-1/1/Quantum_FAQs_20210804.PDF. 2021-08. (2024-12-01 閲覧).

[9] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, O. Regev. Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation. SIAM J. Comput. Vol. 37, Num. 1 (2007), pp. 166–194.

[10] M. Ajtai, Cynthia Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. STOC. ACM, 1997, pp. 284–293.

[11] G. Alagic et al. Status Report on the First Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8528, https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8528.pdf. 2024-10.

[12] G. Alagic et al. Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8204, https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf. 2019-01.

[13] G. Alagic et al. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8413, https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413-upd1.pdf. 2022-07.

[14] T. Albash, D. A. Lidar. Adiabatic quantum computation. Rev. Mod. Phys. Vol. 90, Iss. 1 (2018), p. 015002.

[15] Amazon Braket 量子コンピュータ. https://aws.amazon.com/jp/braket/quantum-computers/.

[16] M. V. Anand, E. E. Targhi, G. N. Tabia, D. Unruh. Post-Quantum Security of the CBC, CFB, OFB, CTR, and XTS Modes of Operation. PQCrypto. Vol. 9606. Lecture Notes in Computer Science. Springer, 2016, pp. 44–63.

[17] ANSSI. ANSSI views on the Post-Quantum Cryptography transition (2023 follow up). https://cyber.gouv.fr/sites/default/files/document/follow_up_position_paper_on_post_quantum_cryptography.pdf. 2023-12. (2024-12-06 閲覧).

[18] Y. Aono, S. Liu, T. Tanaka, S. Uno, R. Van Meter, N. Shinohara, R. Nojima. The Present and Future of Discrete Logarithm Problems on Noisy Quantum Computers. IEEE Transactions on Quantum Engineering. Vol. 3 (2022), pp. 1–21.

[19] B. Apolloni, N. Cesa-Bianchi, D. De Falco. A numerical implementation of "quantum annealing". Proceedings of the Ascona-Locarno conference. 1988, pp. 97–111.

[20] GSM Association. Post Quantum Cryptography – Guidelines for Telecom Use Cases. https://www.gsma.com/newsroom/wp-content/uploads/PQ.03-Post-Quantum-Cryptography-Guidelines-for-Telecom-Use-v1.0.pdf. 2024-02. (2024-12-06 閲覧).

[21] E. Barker. Recommendation for Key Management: Part 1 – General. NIST SP 800-57 Part 1 Rev. 5, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf. 2020-05.

[22] D. J. Bernstein et al. NTRU Prime. https://ntruprime.cr.yp.to/. (2024-12-06 閲覧).

[23] D. Bluvstein et al. Logical quantum processor based on reconfigurable atom arrays. Nature. Vol. 626, Num. 7997 (2023), pp. 58–65.

[24] D. Boneh, M. Zhandry. Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World. CRYPTO (2). Vol. 8043. Lecture Notes in Computer Science. Springer, 2013, pp. 361–379.

[25] C. Boutin. NIST Releases First 3 Finalized Post-Quantum Encryption Standards. https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards. 2023-08. (2024-12-06 閲覧).

[26] G. Brassard. Searching a Quantum Phone Book. Science. Vol. 275, Num. 5300 (1997), pp. 627–628.

[27] G. Brassard, P. Høyer and A. Tapp. Quantum Cryptanalysis of Hash and Claw-Free Functions. LATIN. Vol. 1380. Lecture Notes in Computer Science. Springer, 1998, pp. 163–169.

[28] S. Bravyi, D. P. DiVincenzo, R. Oliveira, B. M. Terhal. The complexity of stoquastic local Hamiltonian problems. Quantum Inf. Comput. Vol. 8, Num. 5 (2008), pp. 361–385.

[29] J. Buchmann et al. Post-Quantum Signatures. (2004). https://eprint.iacr.org/2004/297.

[30] J. Sevilla and C. J. Riedel. Forecasting timelines of quantum computing. (2020). arXiv: 2009.05045.

[31] J.-Y. Cai. Shor's algorithm does not factor large integers in the presence of noise. Science China Information Sciences. Vol. 67, Num. 7 (2024).

[32] W. Castryck, T. Decru. An Efficient Key Recovery Attack on SIDH. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 423–447.

[33] National Cyber Security Centre. Guidelines for quantum-safe transport-layer encryption. https://www.ncsc.nl/documenten/publicaties/2022/juli/guidelines-for-quantum-safe-transport-layer-encryption/guidelines-for-quantum-safe-transport-layer-encryption. 2022-07. (2024-12-06 閲覧).

[34] National Cyber Security Centre. Next steps in preparing for post-quantum cryptography. https://www.ncsc.gov.uk/pdfs/whitepaper/next-steps-preparing-for-post-quantum-cryptography.pdf. 2024-08. (2024-12-06 閲覧).

[35] National Cyber Security Centre. The PQC Migration Handbook, Guidelines for migrating to post-quantum cryptography (Version 2). https://publications.tno.nl/publication/34641918/oicFLj/attema-2023-pqc.pdf. 2023-12. (2024-12-06 閲覧).

[36] P. Chapman. Introducing the World's Most Powerful Quantum Computer. https://ionq.com/posts/october-01-2020-introducing-most-powerful-quantum-computer. 2020-10. (2024-12-01 閲覧).

[37] C. Chevalier, E. Ebrahimi, QH Vu. On Security Notions for Encryption in a Quantum World. INDOCRYPT. Vol. 13774. Lecture Notes in Computer Science. Springer, 2022, pp. 592–613.

[38] Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. Nature. Vol. 616, Num. 7955 (2024).

[39] H. Collins, C. Nay. IBM Unveils 400 Qubit-Plus Quantum Processor and Next-Generation IBM Quantum System Two. https://newsroom.ibm.com/2022-11-09-IBM-Unveils-400-Qubit-Plus-Quantum-Processor-and-Next-Generation-IBM-Quantum-System-Two. 2022-11. (2024-12-06 閲覧).

[40] Atom Computing. Quantum startup Atom Computing first to exceed 1,000 qubits. https://atom-computing.com/quantum-startup-atom-computing-first-to-exceed-1000-qubits/. 2023-10. (2024-12-01 閲覧).

[41] J.-S. Coron, A. May. Deterministic Polynomial-Time Equivalence of Computing the RSA Secret Key and Factoring. J. Cryptol. Vol. 20, Num. 1 (2007), pp. 39–50.

[42] J.-M. Couveignes. Hard Homogeneous Spaces. Cryptology ePrint Archive, Paper 2006/291. 2006. https://eprint.iacr.org/2006/291.

[43] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, J. M. Gambetta. Validating quantum computers using randomized model circuits. Phys. Rev. A. Vol. 100, Iss. 3 (2019), p. 032328.

[44] D-Wave. Ahead of the Game: D-Wave Delivers Prototype of Next-Generation Advantage2 Annealing Quantum Computer. https://www.dwavesys.com/company/newsroom/press-release/ahead-of-the-game-d-wave-delivers-prototype-of-next-generation-advantage2-annealing-quantum-computer/. 2022-06. (2024-12-01 閲覧).

[45] D-Wave. The Most Connected and Powerful Quantum Computer Built for Business. https://www.dwavesys.com/solutions-and-products/systems/. (2024-12-01 閲覧).

[46] U. Dieter. How to calculate shortest vectors in a lattice. Mathematics of Computation. Vol. 29 (1975), pp. 827–833.

[47] J. Ding, G. Spallitta, R. Sebastiani. Experimenting with D-Wave quantum annealers on prime factorization problems. Frontiers Comput. Sci. Vol. 6 (2024).

[48] M. Ekerå, J. Gärtner. Extending Regev's Factoring Algorithm to Compute Discrete Logarithms. PQCrypto (2). Vol. 14772. Lecture Notes in Computer Science. Springer, 2024, pp. 211–242.

[49] ETSI. Quantum-Safe Cryptography (QSC); Limits to quantum computing applied to symmetric key sizes. https://www.etsi.org/deliver/etsi_gr/QSC/001_099/006/01.01.01_60/gr_QSC006v010101p.pdf. 2017-02. (2024-12-01 閲覧).

[50] National Cybersecurity Center of Excellence. Migration to Post-Quantum Cryptography. https://www.nccoe.nist.gov/crypto-agility-considerations-migrating-post-quantum-cryptographic-algorithms.

[51] European Quantum Flagship. Strategic Research and Industry Agenda. https://qt.eu/media/pdf/Strategic-Reseach-and-Industry-Agenda-2030.pdf. 2024-02.

[52] A. Frank et al. Quantum supremacy using a programmable superconducting processor. Nature. Vol. 574, Num. 7779 (2019), pp. 505–510.

[53] K. Fujii. Quantum speedup in stoquastic adiabatic quantum computation. (2018). arXiv: 1803.09954.

[54] K. Fujii. Quantum speedup in stoquastic adiabatic quantum computation. 2019-01. QIP 2019 Poster session https://jila.colorado.edu/qip2019/qip2019_posters_monday.pdf.

[55] J. Gambetta. The hardware and software for the era of quantum utility is here. https://jila.colorado.edu/qip2019/qip2019_posters_monday.pdf. 2023-12. (2024-12-01 閲覧).

[56] C. Gidney, M. Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum. Vol. 5 (2021), p. 433.

[57] É. Gouzien, D. Ruiz, F.-M. Le Régent, J. Guillaud, N. Sangouard. Performance Analysis of a Repetition Cat Code Architecture: Computing 256-bit Elliptic Curve Logarithm in 9 Hours with 126 133 Cat Qubits. Phys. Rev. Lett. Vol. 131, Iss. 4 (2023), p. 040602.

[58] É. Gouzien, N. Sangouard. Factoring 2048-bit RSA Integers in 177 Days with 13 436 Qubits and a Multimode Memory. Phys. Rev. Lett. Vol. 127, Iss. 14 (2021), p. 140503.

[59] ASC X9 Quantum Computing Risk Study Group. Quantum Computing Risks to the Financial Services Industry. https://x9.org/download-qc-ir/. 2022-11. (2024-12-05 閲覧).

[60] FS-ISAC's post-quantum cryptography working group. Preparing for a post-quantum world by managing cryptographic risk. https://www.fsisac.com/hubfs/Knowledge/PQC/Preparing/ForAPostQuantumWorldByManagingCryptographicRisk.pdf?hsLang=en. 2023-03. (2024-12-05 閲覧).

[61] L. K. Grover. A fast quantum mechanical algorithm for database search. STOC. ACM, 1996, pp. 212–219.

[62] N. N. Hegade, K. Paul, F. Albarrán-Arriagada, X. Chen, E. Solano. Digitized adiabatic quantum factorization. Phys. Rev. A. Vol. 104, Iss. 5 (2021), p. L050403.

[63] P. E. Hoffman, S. Celi. Post-Quantum Use In Protocols (pquip). https://datatracker.ietf.org/wg/pquip/about/. (2024-12-06 閲覧).

[64] IBM Quantum Platform. https://quantum.ibm.com/.

[65] IBM、次世代量子プロセッサーおよび IBM Quantum System Two を発表するとともに、 実用的な量子コンピューティングの時代の前進に向けロードマップを拡張. https://jp.newsroom.ibm.com/2023-12-05-IBM-Debuts-Next-Generation-Quantum-Processor-IBM-Quantum-System-Two,-Extends-Roadmap-to-Advance-Era-of-Quantum-Utility. 2023-12. (2024-12-01 閲覧).

[66] T. Ichikawa et al. Current numbers of qubits and their uses. Nature Reviews Physics. Vol. 6, Num. 6 (2024), pp. 345–347.

[67] Intel's New Chip to Advance Silicon Spin Qubit Research for Quantum Computing. https://www.intel.com/content/www/us/en/newsroom/news/quantum-computing-chip-to-advance-research.html. 2023-06. (2024-12-01 閲覧).

[68] ISO. PQCRYPTO Post-quantum cryptography for long-term security. https://www.iso.org/organization/5984715.html. (2024-12-06 閲覧).

[69] P. John. Quantum Computing in the NISQ era and beyond. Quantum. Vol. 2 (2018), p. 79.

[70] S. P. Jordan. Quantum Computation Beyond the Circuit Model. 2008. arXiv: 0809.2307.

[71] T. Jörg, F. Krzakala, G. Semerjian, F. Zamponi. First-Order Transitions and the Performance of Quantum Algorithms in Random Optimization Problems. Phys. Rev. Lett. Vol. 104, Iss. 20 (2010), p. 207206.

[72] R. Kannan. Improved Algorithms for Integer Programming and Related Lattice Problems. STOC. ACM, 1983, pp. 193–206.

[73] M. Kaplan, G. Leurent, A. Leverrier, M. Naya-Plasencia. Quantum Differential and Linear Cryptanalysis. IACR Trans. Symmetric Cryptol. Vol. 2016, Num. 1 (2016), pp. 71–94.

[74] H. Kuwakado, M. Morii. Security on the quantum-type Even-Mansour cipher. ISITA. IEEE, 2012, pp. 312–316.

[75] L. Lamport. Constructing digital signatures from a one-way function. SRI International Technical Report, CSL-98. 1979-10.

[76] J. Lavoie, Z. Vernon. Beating classical computers with Borealis. https://www.xanadu.ai/blog/beating-classical-computers-with-Borealis. 2022-06. (2024-12-01 閲覧).

[77] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen. Vol. 261, Num. 4 (1982), pp. 515–534.

[78] Linwen. Quantum Leap: China's 504-Qubit Chip Narrows the US Gap. https://thechinaacademy.org/quantum-leap-chinas-504-qubit-chip-narrows-the-us-gap/. 2024-04. (2024-12-01 閲覧).

[79] M. Mariantoni. Building a superconducting quantum computer (Invited Talk). PQCrypto 2014. 2024-10. (2024-12-01 閲覧) 暗号危殆化の予測については動画 https://www.youtube.com/watch?v=wWHAs--HA1c の 49:30 で述べられている.

[80] T. Matsumoto, H. Imai. Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. EUROCRYPT. Vol. 330. Lecture Notes in Computer Science. Springer, 1988, pp. 419–453.

[81] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report. Vol. 44 (1978), pp. 114–116.

[82] C. McGeoch, P. Farre. D-Wave Advantage システム: 概要. https://dwavejapan.com/app/uploads/2020/12/14-1049A-A_J-The_D-Wave_Advantage_System_An_Overview_0-.pdf. 2020-12. (2024-12-01 閲覧).

[83] A. Mirko, Z. H. Saleem, K. Muir. Experimental study of Shor's factoring algorithm using the IBM Q Experience. Physical Review A. Vol. 100, Iss. 1 (2019), p. 012305.

[84] D. Moody. Announcement: The End of the 3rd Round – the First PQC Algorithms to be Standardized. https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/G0DoD7lkGPk/m/f3Hl0sh3AgAJ. 2022-07. (2024-12-06 閲覧).

[85] D. Moody. Are we there yet? An Update on the NIST PQC Standardization Project. https://csrc.nist.gov/csrc/media/Presentations/2024/update-on-the-nist-pqc-standardization-project/images-media/moody-are-we-there-yet-pqc-pqc2024.pdf. 2024-04. (2024-12-01 閲覧).

[86] D. Moody. Post-Quantum Cryptography: NIST's Plan for the Future. https://pqcrypto2016.jp/data/pqc2016_nist_announcement.pdf. 2016-02. (2024-12-06 閲覧).

[87] D. Moody. security category reference. https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/OmLRb2rQyN4/m/_7y82chdAQAJ. 2022-09. (2024-12-06 閲覧).

[88] D. Moody, R. Perlner, A. Regenscheid, A. Robinson, D. Cooper. Transition to Post-Quantum Cryptography Standards. NIST IR 8547 (initial public draft), https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf. 2024-11. (2025-02-17 閲覧).

[89] D. Moody et al. Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8309, https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf. 2020-07.

[90] M. Mosca. Cybersecurity in a quantum world: will we be ready? Workshop on Cybersecurity in a Post-Quantum World. Session 8. 2015-04. (2024-02-29 閲覧).

[91] M. Mosca, M. Piani. 2023 Quantum Threat Timeline Report. https://globalriskinstitute.org/publication/2023-quantum-threat-timeline-report/. 2023-12. (2024-12-02 閲覧).

[92] National Security Memorandum on Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems. https://www.whitehouse.gov/briefing-room/statements-releases/2022/05/04/national-security-memorandum-on-promoting-united-states-leadership-in-quantum-computing-while-mitigating-risks-to-vulnerable-cryptographic-systems/. 2022-05. (2025-01-11 閲覧).

[93] NCSA. Guidelines for Post – Quantum Readiness. https://www.navy.mi.th/storage/frontend/article/23852/file/th/Quantum%20Readiness.pdf. 2023-12. (2024-12-06 閲覧).

[94] W. Newhouse et al. Migration to Post-Quantum Cryptography: Preparation for Considering the Implementation and Adoption of Quantum Safe Cryptography. NIST SP 1800-38 (initial preliminary draft), https://csrc.nist.gov/pubs/sp/1800/38/iprd-(1). 2023-12. (2025-02-17 閲覧).

[95] Z. Ni et al. Beating the break-even point with a discrete-variable-encoded logical qubit. Nature. Vol. 616, Num. 7955 (2023), pp. 56–60.

[96] NIST. Digital Signature Standard (DSS). NIST FIPS 186-5, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf. 2023-02.

[97] NIST. Module-Lattice-Based Digital Signature Standard. NIST FIPS 204, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf. 2024-08.

[98] NIST. Module-Lattice-Based Key-Encapsulation Mechanism Standard. NIST FIPS 203, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf. 2024-08.

[99] NIST. Standardization of additional digital signature schemes, call for proposals. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf. 2022-10. (2024-03-05 閲覧).

[100] NIST. Stateless Hash-Based Digital Signature Standard. NIST FIPS 205, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf. 2024-08.

[101] NIST. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf. 2016-12. (2024-03-05 閲覧).

[102] NÚKIB. Minimum requirements for cryptographic algorithms – Cryptographic security recommendations. https://nukib.gov.cz/download/publications_en/Minimum_Requirements_for_Cryptographic_Algorithms_final.pdf. 2023-11. (2024-12-06 閲覧).

[103] H. Ong, C. P. Schnorr. Signatures through Approximate Representation by Quadratic Forms. CRYPTO. Plenum Press, New York, 1983, pp. 117–131.

[104] OpenSSH 9.0 was released. https://www.openssh.com/txt/release-9.0. 2022-04. (2024-12-06 閲覧) Streamlined NTRU Prime と X25519 を組み合わせたハイブリッド鍵交換は 8.5 で試験的に実装され，9.0 からはデフォルトで利用される仕様となっている.

[105] J. Park et al. Rydberg-atom experiment for the integer factorization problem. Physical Review Research. Vol. 6, Iss. 2 (2024), p. 023241.

[106] X. Peng, Z. Liao, N. Xu, G. Qin, X. Zhou, D. Suter, J. Du. Quantum Adiabatic Algorithm for Factorization and Its Experimental Implementation. Physical Review Letters. Vol. 101, Iss. 22 (2008), p. 220405.

[107] L. Phab, S. Louise, R. Sirdey. First Attempts at Cryptanalyzing a (Toy) Block Cipher by Means of Quantum Optimization Approaches. J. Comput. Sci. Vol. 69 (2023), p. 102004.

[108] Post-Quantum Cryptography Lounge. https://www.safecrypto.eu/pqclounge/.

[109] pqc-forum. https://groups.google.com/a/list.nist.gov/g/pqc-forum. 2016/08/01 開始.

[110] L. Qiu, M. Alam, A. Ash-Saki, S. Ghosh. Resiliency analysis and improvement of variational quantum factoring in superconducting qubit. ISLPED. ACM, 2020, pp. 229–234.

[111] Quantinuum H-Series quantum computer accelerates through 3 more performance records for quantum volume. https://www.quantinuum.com/blog/quantinuum-h-series-quantum-computer-accelerates-through-3-more-performance-records-for-quantum-volume. 2023-06. (2024-12-01 閲覧).

[112] Quantinuum's H-Series hits 56 physical qubits that are all-to-all connected, and departs the era of classical simulation. https://www.quantinuum.com/blog/quantinuums-h-series-hits-56-physical-qubits-that-are-all-to-all-connected-and-departs-the-era-of-classical-simulation. 2024-06. (2024-12-01 閲覧).

[113] S. Ragavan, V. Vaikuntanathan. Space-Efficient and Noise-Robust Quantum Factoring. CRYPTO (6). Vol. 14925. Lecture Notes in Computer Science. Springer, 2024, pp. 107–140.

[114] S. Ramos-Calderer, C. Bravo-Prieto, R. Lin, E. Bellini, M. Manzano, N. Aaraj, J. I. Latorre. Solving systems of Boolean multivariate equations with quantum annealing. Phys. Rev. Res. Vol. 4, Iss. 1 (2022), p. 013096.

[115] O. Regev. An Efficient Quantum Factoring Algorithm. arXiv: 2308.06572.

[116] R. L. Rivest, A. Shamir, L. M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM. Vol. 21, Num. 2 (1978), pp. 120–126.

[117] Round 2 Additional Signatures. https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures. 2024-10. (2024-12-06 閲覧).

[118] Round 4 Submissions. https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions. 2022-07. (2024-12-06 閲覧).

[119] C. P. Schnorr. Fast Factoring Integers by SVP Algorithms, corrected. Cryptology ePrint Archive, Paper 2021/933. 2021. https://eprint.iacr.org/2021/933.

[120] Federal office for information security. Cryptographic mechanisms: recommendations and key lengths version: 2024-1. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.html. 2024-02. (2024-12-05 閲覧).

[121] P. W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. FOCS. IEEE Computer Society, 1994, pp. 124–134.

[122] P. W. Shor. Fault-Tolerant Quantum Computation. FOCS. IEEE Computer Society, 1996, pp. 56–65.

[123] P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. Vol. 26, Num. 5 (1997), pp. 1484–1509.

[124] V. V. Sivak et al. Real-time quantum error correction beyond break-even. Nature. Vol. 616, Num. 7955 (2023), pp. 50–55.

[125] U. Skosana, M. Tame. Demonstration of Shor's factoring algorithm for $N = 21$ on IBM quantum processors. Scientific Reports. Vol. 11, Num. 16599 (2021).

[126] M. Sobhani, Y. Chai, T. Hartung, K. Jansen. Variational Quantum Eigensolver Approach to Prime Factorization on IBM's Noisy Intermediate Scale Quantum Computer. arXiv: 2410.01935.

[127] E. G. Johansen and T. Simula. Prime number factorization using a spinor Bose-Einstein condensate-inspired topological quantum computer. Quantum Inf. Process. Vol. 21, Num. 1 (2022), p. 31.

[128] The Cryptographers' Panel. https://www.rsaconference.com/library/presentation/usa/2023/the%20cryptographers%20panel. 2023-04. RSA Conference 2023 (2024-12-05 閲覧).

[129] The Leap quantum cloud service. https://www.dwavesys.com/solutions-and-products/cloud-platform/.

[130] TRAFICOM. Kryptografiset vahvuusvaatimukset luottamuksellisuuden suojaamiseen – kansalliset turvallisuusluokat. https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/file/Kryptografiset_vahvuusvaatimukset_-_kansalliset_turvallisuusluokat_0.pdf. 2024-09. (2024-12-06 閲覧).

[131] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, R. Cleve, I. L. Chuang. Experimental Realization of an Order-Finding Algorithm with an NMR Quantum Computer. Phys. Rev. Lett. Vol. 85, Iss. 25 (2000), pp. 5452–5455.

[132] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature. Vol. 414, Num. 6866 (2001), pp. 883–887.

[133] D.-S. Wang. A comparative study of universal quantum computing models: Toward a physical unification. Quantum Eng. Vol. 3, Num. 4 (2021).

[134] W. Wang, Z. You, S. Wang, Z. Tang, H. Ian. Computing Shor's algorithmic steps with classical light beams. Scientific Reports. Vol. 12, Num. 21157 (2022).

[135] D. Willsch, P. Hanussek, G. Hoever, M. Willsch, F. Jin, H. De Raedt, K. Michielsen. The State of Factoring on Quantum Computers. 2024. arXiv: 2410.14397.

[136] S. Xu et al. Digital Simulation of Projective Non-Abelian Anyons with 68 Superconducting Qubits. Chinese Physics Letters. Vol. 40, Num. 6 (2023), p. 060301.

[137] B. Yan et al. Factoring integers with sublinear resources on a superconducting quantum processor. arXiv: 2212.12372.

[138] C. Zalka. Grover's quantum searching algorithm is optimal. Phys. Rev. A. Vol. 60, Iss. 4 (1999), pp. 2746–2751.

[139] J. Zander. Advancing science: Microsoft and Quantinuum demonstrate the most reliable logical qubits on record with an error rate 800x better than physical qubits. https://blogs.microsoft.com/blog/2024/04/03/advancing-science-microsoft-and-quantinuum-demonstrate-the-most-reliable-logical-qubits-on-record-with-an-error-rate-800x-better-than-physical-qubits/. 2024-04. (2024-12-01 閲覧).

[140] デジタル庁, 総務省, 経済産業省. 暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準. CRYPTREC LS-0003-2022r1, https://www.cryptrec.go.jp/list/cryptrec-ls-0003-2022r1.pdf. 2022-03.

[141] デジタル庁, 総務省, 経済産業省. 電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）. CRYPTREC LS-0001-2022r1, https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2022r1.pdf. 2024-05.

[142] 中国密码学会. 全国密码算法设计竞赛通知. https://sfjs.cacrnet.org.cn/site/content/309.html. 2018-06. (2025-01-11 閲覧).

[143] 中国密码学会. 关于全国密码算法设计竞赛算法评选结果的公示. https://sfjs.cacrnet.org.cn/site/content/854.html. 2020-01. (2025-01-11 閲覧).

[144] 橋本 俊和, 梅木 毅伺, 柏﨑 貴大, 井上 飛鳥. 連続量光量子コンピュータに向けた光技術. https://www.rd.ntt/research/JN202304_21560.html. 2023-04. (2024-12-01 閲覧).

[145] 豊田 健二. 究極のコンピュータへ「もう一つの道」 イオンで可視化する量子情報. https://resou.osaka-u.ac.jp/ja/story/2021/specialite_002_6. 2021. (2024-12-01 閲覧).

[146] 宮地 充子. 楕円曲線の理論的及び実用的可能性. IEICE FUNDAMENTALS REVIEW. Vol. 14, Num. 4 (2021), pp. 329–336.

[147] 細山田 光倫. 量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価. CRYPTREC EX-2901-2019, https://www.cryptrec.go.jp/exreport/cryptrec-ex-2901-2019.pdf. 2020-01.

[148] 縫田 光司. 耐量子計算機暗号. 森北出版, 2020.

[149] 伊藤 公平. 量子計算. 2010-02. https://www.ieice-hbkb.org/files/ad_base/view_pdf.html?p=/files/S2/S2gun_05hen_03.pdf. 電子情報通信学会 知識ベース 知識の森 S2 群（ナノ・量子・バイオ）5 編（量子通信と量子計算）3 章.

[150] 小林 和淑. ムーンショット目標 6 公開シンポジウム 2023 〜誤り耐性型汎用量子コンピュータの実現を目指して〜. https://www.youtube.com/watch?v=ebkT0LyKIKk. 2023-03.

[151] 国立国会図書館調査及び立法考査局. 量子情報技術：科学技術に関する調査プロジェクト報告書. 2022-03. https://www.ndl.go.jp/jp/diet/publication/document/2022/index.html.

[152] 国立大学法人東北大学, 日本電気株式会社. 新開発の 8 量子ビット量子アニーリングマシンを利用して東北大学と NEC が将来のコンピュータシステムに関する共同研究を開始. https://jpn.nec.com/press/202306/20230628_01.html. 2023-06. (2024-12-01 閲覧).

[153] 国立研究開発法人科学技術振興機構. 目標 6 2050 年までに、経済・産業・安全保障を飛躍的に発展させる誤り耐性型汎用量子コンピュータを実現. https://www.jst.go.jp/moonshot/program/goal6/index.html. (2024-12-01 閲覧).

[154] 富士通. 量子コンピュータの誤り耐性量子計算を解説！エラー訂正とエラー緩和の最新トレンドを紐解く. https://activate.fujitsu/ja/key-technologies-article/ta-fault-tolerant-quantum-computation-20240515. 2024-05. (2024-12-01 閲覧).

[155] 向井 寛人, 朝永 顕成, 蔡 兆申. 超伝導量子コンピュータの基礎と最先端. 低温工学. Vol. 53, Num. 5 (2018), pp. 278–286.

[156] 鈴木 教洋. 日立の量子コンピュータ研究開発戦略. https://www8.cao.go.jp/cstp/ryoshigijutsu/jitsuyo_wg/3kai/siryo2-2.pdf. 2022-12. (2024-12-01 閲覧).

[157] 高安 敦. Shor のアルゴリズム実装動向調査. CRYPTREC EX-2005-2020, https://www.cryptrec.go.jp/exreport/cryptrec-ex-3005-2020.pdf. 2021-06.

[158] 新方式の量子コンピュータを実現 ー世界に先駆けて汎用型光量子計算プラットフォームが始動ー. https://group.ntt/jp/newsrelease/2024/11/08/241108a.html. 2024-11. (2024-12-01 閲覧).

[159] 日本経済新聞. 量子計算機で新会社　富士通・日立など 10 社 産学で商用化. https://ohmori.ims.ac.jp/news/2024/02/27/2607/. 2024-02. (2024-12-01 閲覧).

[160] 日立、量子コンピュータの実用化に向けて 量子ビットの寿命を 100 倍以上長く安定化させる操作技術を開発. https://www.hitachi.co.jp/New/cnews/month/2024/06/0617.html. 2024-06. (2024-12-01 閲覧).

[161] 清藤 武暢, 四方 順司. 量子コンピュータが共通鍵暗号の安全性に与える影響. 金融研究. Vol. 38, Num. 1 (2019), pp. 45–72. https://cir.nii.ac.jp/crid/1523106604811659392.

[162] 樽茶 清悟. 拡張性のあるシリコン 量子コンピュータ技術の開発. https://www.jst.go.jp/moonshot/sympo/20230328/pdf/01_20230328_tarucha.pdf. 2023-03. (2024-12-01 閲覧).

[163] 理化学研究所. シリコン量子ビットの高温動作に成功 ー大型冷却装置が不要に、センサーなど幅広い量子ビット応用へー. https://www.riken.jp/press/2019/20190124_3/. 2019-01. (2024-12-01 閲覧).

[164] 理化学研究所. シリコン量子ビットの高精度読み出しを実現 ー半導体系の誤り耐性量子コンピュータの実現に前進ー. https://www.riken.jp/press/2024/20240213_2/index.html. 2024-02. (2024-12-01 閲覧).

[165] 理化学研究所. 量子コンピュータを利用できる「量子計算クラウドサービス」開始 ー国産超伝導量子コンピュータ初号機の公開ー. https://www.riken.jp/pr/news/2023/20230324_1/. 2023-03. (2024-12-01 閲覧).

[166] 理化学研究所. 量子コンピュータ開発に挑む若手研究者たち. https://www.riken.jp/pr/closeup/2023/20230904_1/index.html. 2023-09. (2024-12-01 閲覧).

[167] 産業技術総合研究所. 独自のアーキテクチャを用いた超伝導量子アニーリングマシンを実現. https://www.aist.go.jp/aist_j/new_research/2021/nr20210706/nr20210706.html. 2021-07. (2024-12-01 閲覧).

[168] 大関 真之. 量子アニーリングが拓く機械学習と計算技術の新時代 (量子システム推定の数理). 数理解析研究所講究録. Vol. 2059 (2017), pp. 13–23. https://cir.nii.ac.jp/crid/1050564288163922560.

[169] 文部科学省 科学技術・学術政策研究所科学技術予測センター. 第 11 回科学技術予測調査 デルファイ調査. https://nistep.repo.nii.ac.jp/?action=repository_uri&item_id=6692&file_id=13&file_no=3. 2020-06. (2024-12-05 閲覧).

[170] 山口 純平, 伊豆 哲也. イジング計算を用いた暗号解析について. オペレーションズ・リサーチ：経営の科学. Vol. 67, Num. 6 (2022), pp. 290–296. https://cir.nii.ac.jp/crid/1520011030559130112.

[171] 山口 純平, 伊豆 哲也, 國廣 昇. 素因数分解問題に対する Shor アルゴリズムの実装と量子計算機シミュレータを用いた実験. 暗号と情報セキュリティシンポジウム (SCIS 2023). 2023-01, 4A2–3.

[172] 大塩 耕平. アナログ量子シミュレータの開発動向と応用. https://www.mizuho-rt.co.jp/publication/others/pdf/mhrt003_01.pdf. 2024-03. (2024-12-06 閲覧).

[173] 塩見 英久. マイクロ波技術者から学ぶ超伝導量子コンピュータ入門. MWE2023 マイクロウェーブ ワークショップ プログラム FR6A 基礎講座. (2023). https://apmc-mwe.org/mwe2024/pdf/tut23/FR6A-1.pdf.

[174] 徳永 裕己. 誤り耐性量子コンピュータの早期実現に向けた取り組み. NTT 技術ジャーナル. Vol. 35, Num. 9 (2023), pp. 26–29.

[175] 大森 賢治. 大規模・高コヒーレンスな動的原子アレー型・誤り耐性量子コンピュータ. https://www.youtube.com/watch?v=OIVQ5ZmdCEo. 2024-03. (2024-12-01 閲覧).

[176] 大阪大学 量子情報・量子生命研究センター. 【プレスリリース】大阪大学に設置した超伝導量子コンピュータ国産 3 号機の クラウドサービスを開始. https://qiqb.osaka-u.ac.jp/20231220pr/. 2023-12. (2024-12-01 閲覧).

[177] 量子耐性暗号研究団. KpqC. https://kpqc.or.kr/. (2024-12-06 閲覧).

[178] 富田 隆文. 冷却原子型量子コンピュータの急速な発展とその展望について. https://www-conf.kek.jp/joint-colloquium/slides/Tomita.pdf. 2024-03. (2024-12-01 閲覧).

[179] 満保 雅浩. 公開鍵暗号. 映像情報メディア学会誌. Vol. 69, Num. 9 (2015), pp. 714–720.

第 2 章

PQC の活用方法

将来，一定以上の能力を持つ量子コンピュータが登場した場合には，既存の公開鍵暗号が解読される（破られる）という脅威が指摘されている [1, 20]。本章では，現在，標準的に用いられている公開鍵暗号の解読が可能となる水準の量子コンピュータを Cryptographically Relevant Quantum Computer（CRQC）と記載し，CRQC を用いた攻撃に対しても安全な性質を「耐量子計算機性」と記載する。また，耐量子計算機暗号（Post-Quantum Cryptography: PQC）とは，耐量子計算機性を持つ暗号アルゴリズムを意味し，本稿の対象である公開鍵暗号アルゴリズム以外にも，共通鍵暗号やハッシュ関数も含まれるものとする [1]。加えて，「耐量子計算機性を持つ情報システム」とは，CRQC を用いた攻撃に対しても安全な情報システムを示すものとする。

ある情報システムが，既存の公開鍵暗号を利用していた場合，その情報システムは，将来における CRQC を用いた攻撃の脅威に晒されることになる。そのような脅威への対応方法としては，情報システム内の（耐量子計算機性を持たない）既存の公開鍵暗号方式部分を，耐量子計算機性を持つ公開鍵暗号方式に置き換えることで，その情報システムに耐量子計算機性を持たせることが考えられる。

なお，耐量子計算機性を持たせるためには異なるアプローチも考えられる。例えば，今まで公開鍵暗号を利用していた情報システムを，公開鍵暗号を利用しない仕組みに置き換えるアプローチである。具体的には，信頼できる特使等の別の情報共有手段を利用し，通信相手と共通鍵の事前共有を行う方法である。しかし，このアプローチでは，情報システムの「スケーラビリティ」[*1]が損なわれることが予想され，場合によっては実現不可能なコストが発生する。

現在，普及している情報システムの中には，公開鍵暗号を利用することにより，そのサービスのスケーラビリティを維持しているものも多い。インターネットはその代表例であり，通信相手を認証する用途等で公開鍵暗号を利用することにより，大規模な通信ネットワークの構築及び維持を実現している [4, 11, 14, 18]。このような大規模情報システムにおいて，仮に，耐量子計算機性を持たせるために公開鍵暗号の利用を取りやめた場合，スケーラビリティが損なわれ，その結果，維持・運用コストが大きく上昇してシステムの維持も困難となる。このため，公開鍵暗号を利用した情報システムの現在及び将来においてスケーラビリティ上の懸念が発生しないという見通しがない限り，耐量子計算機性の実現のためのアプローチとしては，耐量子計算機性を持つ公開鍵暗号を利用することが望ましい。

以下では，より具体的に，耐量子計算機性を持たせるためのアプローチについて紹介する。公開鍵暗号によって暗号化（守秘・鍵共有）を行う情報システムに対して，耐量子計算機性を持たせるアプローチには，表 2.1 に示す手法及びその組み合わせが存在するが，一般に下段のアプローチになるほどスケーラビリティが低下する。ここで，最もスケーラビリティが期待できるデータ削除や匿名化といった手法は，そのデータが削除や匿名化が可能であるか否かを検討した後に実施する必要があり，運用上のスケーラビリティは高いものの，導入前の検討のために時間を必要とし，情報シ

[*1] スケーラビリティとは，要求される処理量等の変化に応じてそのシステムの対処能力を柔軟に増減させることができる能力である。
https://www.gartner.com/en/information-technology/glossary/scalability
本章では，情報システムの規模（ステークホルダ数，利用者数，処理量等）が増減した場合でも，その情報システムが消費するリソース（計算量，通信量，人の手間等）が極端に増加しない，又は，減少させることができる能力の意味で利用する。

表 2.1: 公開鍵暗号による暗号化（守秘・鍵共有）を行う情報システムに対して耐量子計算機性を持たせるためのアプローチ

	アプローチ	概要
1.	削除・匿名化	情報システムが，漏洩しても問題ない情報以外は保管しない/扱わないようにする。又は，保管する/扱う情報を加工することによって，漏洩しても問題ないように変形する。この方式は，スケーラビリティが最も高いが，可用性が大きく低下することが考えられ，選択できないことも多い。
2.	耐量子計算機性を持つ公開鍵暗号の採用	最も一般的な解決策であり，スケーラビリティを確保できる。現代暗号の利点を維持するアプローチである。
3.	公開鍵暗号を用いない鍵共有手段の導入	公開鍵暗号を利用している情報システムを，公開鍵暗号を利用しない仕組み（例えば，物理的に通信相手全員に IC カードを配布することで，共通鍵の事前共有を行うなど）に置き換えることで，耐量子計算機性を持たせる。暗号技術の観点からは，公開鍵暗号が登場する以前の思想で再設計することになる。スケーラビリティが低く，不特定多数が利用するシステムでは採用が困難と考えられる。また，通信当事者の捕捉が容易となることも考えられ，匿名性の確保やプライバシ保護に関する再設計も併せてが必要になる可能性がある。
4.	物理アクセス制御	上記 1〜3 のアプローチが採用できない場合にも採用可能である。暗号技術の観点からは，暗号技術が発展する以前の思想で再設計することになる。実装コスト及び運用コストが非常に高くなることが予想される。

ステムの可用性が低下するおそれもある。また，法令やポリシー等で削除・匿名化が許容されていない場合には，実施できないおそれもある。

これらの事情より，耐量子計算機性を持たせるための最も汎用的かつ根本的な対応は，既存の公開鍵暗号方式を耐量子計算機性を持つ公開鍵暗号方式に置き換えることであると考えられる。

ただし，情報システムで利用されている公開鍵暗号方式を，耐量子計算機性を持つ公開鍵暗号方式に置き換えることは容易ではない。公開鍵暗号方式を，耐量子計算機性を持つものへとすることは，実装をシンプルに切り替えただけでは完了せず，公開鍵暗号がどのように利用されているのかについて認識した上で，運用やデータ管理に係る様々な処理も併せて実施することが要求される（以降，暗号方式の置き換えに加えて，これらの処理を行うことを「暗号移行」と呼ぶ)。そこで本章では，公開鍵暗号のいくつかの利用形態を念頭に，耐量子計算機性を持つ公開鍵暗号方式への暗号移行について紹介する。まず，現行の公開鍵暗号の利用形態を紹介した上で，各利用形態における CRQC による脅威及びその対策について，システム運用やデータ管理処理の観点を踏まえて概説する。また，脅威を評価する上で重要となる，保護対象となるデータの保護期間について記載した上で，利用形態や保護対象を踏まえた対応についても概説する。

2.1 公開鍵暗号の利用形態

既存の公開鍵暗号方式を，耐量子計算機性を持つものへと暗号移行するに際しては，その公開鍵暗号方式の利用形態ごとに，暗号移行のプロセスが大きく異なることが予想される。そこで，本節で公開鍵暗号の利用形態について概説した上で，次節以降で各利用形態における暗号移行のプロセスについて述べる。公開鍵暗号にはいくつかの利用形態が存在するが，本章では「電子政府における調達のために参照すべき暗号のリスト」[24] （以下「CRYPTREC 暗号リス

ト」と呼ぶ。）に合わせて，公開鍵暗号を署名・守秘・鍵共有に分類し，以降その分類に沿って概説する。また，本節では，署名用途／守秘用途／鍵共有用途の耐量子計算機性を持つ公開鍵暗号方式を，それぞれ署名用途／守秘用途／鍵共有用途の PQC と表記する。

2.1.1 署名用途での公開鍵暗号の利用

本節では，署名を付与する行為を「デジタル署名処理」と呼び，付与される署名データを「デジタル署名」と呼ぶ。デジタル署名が付与されたコンテンツを改竄すると，その改竄を検知することができる。このため，署名用途の公開鍵暗号を用い，コンテンツにデジタル署名を付与することで，コンテンツの改竄によりもたらされる被害を防止することができる。コンテンツは，人が読む文章（ドキュメントデータ），動画等の情報であることもあれば，暗号鍵の鍵情報[*2]であることもある。また，デジタル署名処理に用いられる秘密鍵が，対応する公開鍵を含む電子証明書によって所定の人物／組織／装置等と紐づいている場合では，コンテンツの生成人物／組織／装置を確認（認証）することもできる。このように署名用途の公開鍵暗号は，コンテンツの改竄防止，署名者の認証，データ元の認証等に利用される。

具体的な署名用途の公開鍵暗号の利用例としては，TLS 通信 [18] におけるクライアント認証（利用者の認証）やサーバ認証（サービス提供者の認証），OS のコードサイン（バイナリデータが改竄されていないことの確認）等に広く利用されている。また，公開鍵の配布手段の一種である公開鍵暗号基盤（PKI）の構成においても，公開鍵暗号は広く利用されており [4]，コンテンツに対して署名が付与された時刻を確認可能なタイムスタンプ署名方式 [23] 等も存在する。CRYPTREC 暗号リストには，DSA，ECDSA，EdDSA，RSA-PSS，及び RSASSA-PKCS1-v1_5 が署名用途の公開鍵暗号として記載されている。

2.1.2 守秘用途での公開鍵暗号の利用

守秘用途の公開鍵暗号によって暗号化された暗号文は，対応する秘密鍵なしに復号することは困難となる。このため，守秘用途の公開鍵暗号は，意図した相手だけにデータを提示するために利用することができる。暗号化処理による保護は，ドキュメントデータ，動画等の情報に対して行われることもあれば，暗号鍵の鍵情報[*3]に対して行われることもある。保護が鍵情報に対して行われるユースケースとしては，鍵情報を通信当事者間で共有する場合や，暗号鍵所有者がその鍵情報をバックアップする場合等が該当する。

守秘用途及び鍵共有用途の公開鍵暗号の一般的な実装形態として，公開鍵暗号方式により別の暗号鍵を保護し，その暗号鍵を利用した共通鍵暗号方式によりコンテンツの秘匿性や完全性を保護するというアプローチが存在する。このアプローチでは，共通鍵暗号方式の暗号鍵（以下，単に共通鍵と呼ぶ）は送信者により作成され，配送される。したがって，ある時点で共通鍵が漏洩した場合には，過去にその秘密鍵を持つ利用者に対して配送された共通鍵が漏洩するおそれがある。また，受信者は共通鍵の生成に関わることがないため，送信者が別の通信相手と共通鍵を使い回していても察知することができない。このため，昨今の TLS 通信等における共通鍵の共有においては，守秘用途の公開鍵暗号でなく，次節で概説する鍵共有用途での公開鍵暗号を一時的な鍵と組み合わせて利用することが望ましいと考えられている [5, 19]。なお，「TLS 暗号設定ガイドライン」[5] においても，鍵交換（鍵共有・守秘）においては，Perfect Forward Security（PFS）[*4]の特性を持つ DHE（又は ECDHE）を選択することがセキュリティ上望ましいと記載されている。CRYPTREC 暗号リストには，RSA-OAEP 及び RSAES-PKCS1-v1_5[*5]が守秘用途の公開鍵暗号として記載されている。また，RFC7525 においても [19]，4.1 節において守秘用途で使用される RSA 暗号方式による鍵の転送（RSA key transport）は利用すべきでないと記載されており，4.2 節において一時的（Ephemeral）な鍵を用いる

[*2] 鍵情報には暗号鍵やメタデータが含まれ [5]，公開鍵暗号の鍵のみではなく共通鍵暗号の鍵に関する情報も含む概念となる。
[*3] 秘密鍵，共通鍵，鍵導出鍵及びそれらの鍵のメタデータを含む。
[*4] ある時点における鍵が漏洩した場合でも，漏洩した鍵とは異なる鍵を使用していた過去の暗号文の復号はできない性質。
[*5] 守秘用途の RSAES-PKCS1-v1_5 は，運用監視暗号リストに記載されており，互換性維持以外での利用は推奨されていない。

暗号スイート[*6]が推奨されている。

2.1.3 鍵共有用途での公開鍵暗号の利用

鍵共有用途での公開鍵暗号は，鍵共有に参加する二者が，同一の鍵情報[*7]を共有するために使用される。近年利用されている二者間鍵共有を目的とした多くの公開鍵暗号プロトコルにおいては，鍵共有に参加する双方が何らかの値を生成し，その値に対して秘密鍵を使用した計算を行う。結果として，共有される鍵には双方の生成した値が影響することとなり，一方のみの計算で暗号鍵を導出することはできない。このため，守秘用途でのデータ送付と異なり，送信者があらかじめ意図した特定の鍵を，共有鍵として利用することはできない。CRYPTREC 暗号リストには，DH，ECDH，及び PSEC-KEM が鍵共有用途の公開鍵暗号として記載されている。

2.2 PQC の導入における課題

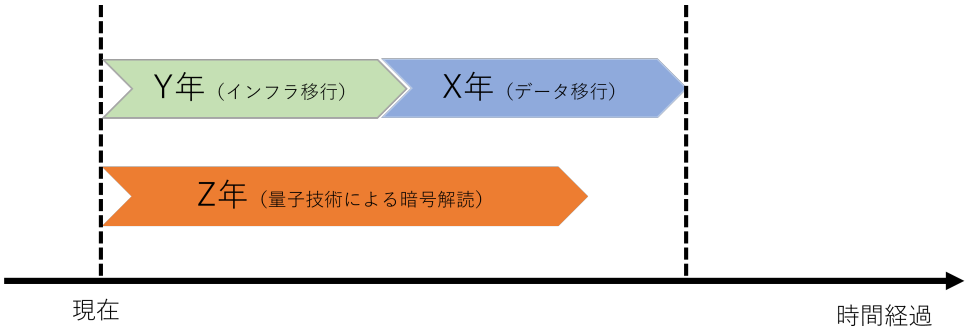


図 2.1: Mosca の発表 [15] より

現在広く利用されている公開鍵暗号が，量子コンピュータを利用した攻撃に起因して，“近い将来”に危殆化する可能性は低い [7] と考えられている。他方で，Michel Mosca [15] が指摘するように，その情報システムで生成されるデータに対して暗号方式による保護が期待される期間（図 2.1 における X）に，暗号処理の実装の置き換えに要する期間（同図における Y）を加えたものが，CRQC による攻撃が実現するまでの期間（同図における Z）よりも長い場合（X + Y > Z の場合）は，当該情報システムで生成されるデータは CRQC による攻撃の脅威にさらされることになる。すなわち，CRQC 実現までの期間（Z）が非常に長く，遠い将来であったとしても，その情報システムの X や Y の値が大きければ，何らかの対応が求められる。なお本章において，特記しない限り以降では，X，Y，Z は図 2.1 における X，Y，Z を示す。

もっとも，CRQC の実現時期は未だ不透明であり，Z を予想することは困難である。また，X は，暗号方式のみならず，保護対象となるデータの性質等によっても大きく異なる。特に，保護対象となるデータに対して，保護期間が設定されていない場合などは，X を導出すること自体が新たな課題となる。同様に，Y も，暗号方式の実装形態によって大きく変化する。さらに，X 及び Y は，情報システムの運用を通して，将来において変動することもありうる。

このように，ある公開鍵暗号アプリケーションが利用されている際に，CRQC による脅威について備える必要性があるか否かを判断しようとした場合，Z は不確定であり，X や Y も変動しうるため，判断が難しいという課題がある [25]。ここで，保護対象となるデータに保護期間が設定されていない場合においては，判断に先駆けて（X 導出のために）データの保護期間を決定することとなり，場合によってはその判断を行うための情報収集に相当の期間を必要と

[*6] 複数の暗号アルゴリズムの組合せ
[*7] 共通鍵暗号の共通鍵，鍵導出機能の鍵やパラメータ等

する。

PQC の導入においては，その情報システムに耐量子計算機性を持たせることが必要なのか，また，いつまでにそれを行う必要があるのか，を判断すること自体が課題となる。

2.2.1 署名用途での課題

署名用途の公開鍵暗号は，コンテンツの改竄防止，認証等に利用されるが，ユースケースによって脅威の性質は大きく異なる。例えば，TLS 通信 [18] におけるクライアント認証やサーバ認証においては，認証用に付与されたデジタル署名の検証を行うのは基本的にその場限りとなるため，X の値は小さくなる。また，Web ブラウザが信用するサーバ認証用の証明書の有効期間は，ごく一部の例外を除いて 1 年程度であり，それほど長い期間利用されることはない。そのため，X の値は，守秘用途や他の認証用途に比べて非常に小さくなる [25]。さらに，ブラウザのアップデートやルート認証局の入れ替えを，より迅速に実施できる体制を整備しており，Y の値も守秘用途や他の認証用途に比べて小さい。

他方で，電子データに対するドキュメント署名や，バイナリデータに対するコードサインであれば，署名対象のデータを利用する人が存在する限り（数十年に渡り）デジタル署名が検証されることもある。特に，コードサインにおいては，仮に電子証明書に有効期間が記載されていたとしても，その有効期間満了後にも検証されることが十分に考えられる。そのため，X の値は，守秘用途や他の認証用途に比べて非常に大きくなる。

このように，署名用途においては，X の値は大きく異なりうるものであり，個々のアプリケーションごとに判断する必要がある。また，公開鍵の配布のために PKI を利用した場合，トラストアンカーの置き換え等に時間を要するため，Y が 10 年以上となることも珍しくない。

ここで，アプリケーションごとの判断の一例として，S/MIME プロトコルを利用するメールクライアントソフトウェアにおいて保護が必要な期間について概説する。S/MIME 用に発行された証明書（に対応する秘密鍵）は，通信相手の認証及び通信データの暗号化に利用可能であり。以下の用途での使用が可能である。

1. 通信時に通信相手を認証する
2. 通信時に通信データの暗号化（復号）に利用する
3. 過去に受け取ったメールの通信相手を後から認証する
4. 過去に受け取った通信データを復号する

3 及び 4 は，S/MIME プロトコルを通信プロトコルと捉えると，所管範囲外とも整理できるが，エンドユーザが利用するメールクライアントソフトウェアの中には，通信終了後の保管されたメールに対しても暗号処理を行うものも存在する。このような処理におけるセキュリティは，保存する／されているデータ（data at rest）のセキュリティとなり，データ保護の対象期間は非常に長くなる。

他方で，メールクライアントソフトウェアが 1 のみをサポートする場合[*8]では，S/MIME 証明書（に対応する秘密鍵）は受信者が送信者を認証した後は利用されることはない。このような，通信路上の転送されているデータ（data in transit）の認証におけるセキュリティでは，保護の対象期間は短くなる。このように，メールクライアントソフトウェア間でも，そのソフトウェアがサポートする機能の違いよって X の値は大きく変化する。

[*8] 通信時のメールコンテンツの暗号化は，（S/MIME 以外の）メールサーバ間の通信プロトコルにて実施することも可能である。そのため，送受信者間の E2E が必要でない場合や，送受信者間の E2E 暗号化が許容されない場合においては，S/MIME による暗号化が行われないこともある。送受信者間の E2E 暗号化が許容されない例としては，メールサーバを運用する組織が，自社ポリシーにて（メールクライアントではなく）メールサーバでのウィルス検知を必須とするケース等が挙げられる。

2.2.2 守秘用途での課題

守秘用途の公開鍵暗号においては，攻撃者が事前に暗号技術で保護されたデータを収集して保存しておき，後からそのデータに対して攻撃を行う攻撃である，Harvest Now Decrypt Later 攻撃（以下，「ハーベスト攻撃」と呼ぶ）[*9]の脅威が指摘されている。

ハーベスト攻撃においては，保護対象となるデータの保護期間，すなわち X の値が大きくなるほど，攻撃者が攻撃可能な期間が長くなる。これは，攻撃者が CRQC の開発を待たずに攻撃（保護された情報の収集）を開始できるためである。一方，防御側は，攻撃者に情報が収集される前に，情報システムに耐量子計算機性を付与することが求められる。保護対象となる情報の保護期間が長くなるほど，この不均衡は大きくなり，攻撃者の攻撃可能期間が長くなる。

守秘用途の公開鍵暗号では，保護対象となるコンテンツや鍵情報の保護期間が非常に長期となることが想定されている場合や，無期限で保護することが想定されている場合も存在する。例えば，患者を特定又は推測可能な形態で保管された遺伝性疾患に関する医療情報や，外交関係の機微な情報，さらには，それらの情報の暗号化に利用される鍵などは長い保護期間を持つ傾向にある。また，ドキュメントの生成時において，無期限に守秘することを前提としており，公開することを想定していない情報も存在する。

これらの情報においては，X の値は非常に大きくなるため，おそらく $X + Y > Z$ が成立することになる。そのため，速やかに CRQC の脅威に対する何らかの対応を行うことで，被害を軽減することが望ましい [25]。

2.2.3 鍵共有用途での課題

鍵共有用途での課題は，守秘用途における課題と同種の課題を含んでいる。例えば，鍵共有で共有された共通鍵が，非常に長い保管期間を持つデータの暗号化に利用されていた場合，X の値は非常に大きくなり，$X + Y > Z$ が成立すると考えられ，速やかに CRQC の脅威に対する何らかの対策が必要となる。

さらに，守秘用途では存在しない新たな懸念も存在する。例えば，一時的（Ephemeral）な鍵情報を用いた DH 鍵共有方式を採用することにより PFS を達成している情報システムが存在し，その情報システムは，PFS であることを前提とした運用ポリシーを策定していたとする。この情報システムの DH 鍵共有処理部分を，耐量子計算機性を持つ標準化された公開鍵暗号方式に置き換える場合，以下の 2 つの方針が考えられる。

1）鍵共有用途の PQC に置き換える
2）守秘用途の PQC に置き換える

標準化された鍵共有用途の PQC が存在するのであれば，1）が選択可能であり，比較的容易に実現可能だと考えられる。しかし，そのような鍵共有用途の PQC が存在せず，守秘用途の PQC しか標準化されていない場合には，2）を選択することとなり，守秘用途の PQC を用いて鍵共有部分を構成することとなる。

2）の選択において，守秘用途の PQC を単純に導入した場合，PFS の性質を持たなくなるおそれがあり，それによりデータ保護及び運用ポリシー策定時に想定していなかった経路からの情報漏洩等が発生する懸念が生じる。また，守秘用途の公開鍵暗号方式に対して何らかの手を加えて，PFS の性質を持つ暗号プロトコルを構成したとしても，その暗号プロトコルが標準化されていない場合は，運用ポリシー上，利用できないこともある。

他方で，2）の選択において，既存の鍵交換及び守秘用途の PQC の両方のハイブリッド構成を用いることによって対応するアプローチも存在する[*10]。ハイブリッド構成を用いることで，既存のアルゴリズムでしか防げない攻撃に対しても，新たなアルゴリズムでしか防げない攻撃に対しても，安全な構成とすることができる [22]。

[*9] Record Now Decrypt Later 攻撃，Store Now Decrypt Later 攻撃等とも呼ばれる。
[*10] TLS における [13, 22]，CMS における [16] 等が当該アプローチとして挙げられる。

例えば，既存の鍵共有方式（ephemeral ECDH 等）を用いた鍵交換で導出された秘密情報と，守秘用途の PQC（ML-KEM 等）を用いて導出された秘密情報の，両方を入力とし，所定のハッシュ計算の出力を暗号鍵とすることにより，PFS の性質を持つアルゴリズムでしか防げない攻撃に対しても，耐量子計算機性を持つアルゴリズムでしか防げない攻撃に対しても安全な構成とすることができる。

もっとも，鍵共有処理を複数回行うことに起因し，処理量及びデータ転送量が増加するため，その増加に対応できるように情報システムや通信プロトコルの修正が必要となりうることには注意が必要である。

2.3 PQC 導入へのアプローチ

2.2 節でも記載したように，CRQC の実現時期（又は実現までの期間 Z）は不透明ながら，X や Y の値が大きな情報システムにおいては，何らかの対応を取ることが望ましい。また，本章冒頭で記載したように，情報システムに耐量子計算機性を持たせる手段は，耐量子計算機性を持つ公開鍵暗号方式の導入だけではないものの，スケーラビリティを考慮すると耐量子計算機性を持つ公開鍵暗号方式の利用が有望である。本節では，耐量子計算機性を持つ公開鍵暗号方式への暗号移行を念頭に，その暗号移行を円滑に行う上での考慮事項について概説する。

2.3.1 プライオリティ設定の重要性

公開鍵暗号は様々な用途において普及している。それらの全ての公開鍵暗号方式を耐量子計算機性を持つものへ暗号移行するためには，長い期間及び労力を要する。また，情報システムの中には，そのシステムの利用期間及び生成されるデータの保護期間が短い等の理由により，耐量子計算機性を持たせる必要がないものも存在するかもしれない。

そこで，暗号移行を検討する上では，X,Y,Z を意識して対応することが重要と考えられる。もっとも，X や Y は暗号方式の利用局面ごとに異なることも想定され，またそれらの値は将来において変動する可能性がある。さらに，Z は不確定であり，予想すること自体も困難である。このような状況の下で，全ての暗号モジュールに対して X,Y,Z を分析するアプローチを取ることは，作業量の観点で大きな困難が伴うことが予想され，結果として本当に保護が必要なデータに対する対応に手が回らないおそれがある。そこで，暗号移行を行う担当者は，優先度の高いものを洗い出し，その優先度に応じて対応を行うことが適切である [9, 26, 25]。

PQC への暗号移行を検討するにあたり，あらかじめ優先順位付けを行うことの重要性は，金融庁の報告書 [27] でも触れられており，基本事項は以下のように整理されている。

- 暗号解読可能な量子コンピュータによる既存の暗号危殆化に関連するリスクに基づいて，移行対象の優先順位付けを行う。
- 移行対象の詳細な把握のため，クリプト・インベントリを構築する。
- 暗号危殆化状況に応じて安全かつ迅速に対応できるアーキテクチャを検討する。
- 優先順位の高いものを中心に移行期限を設定し，期限超過の可能性も踏まえたリスク低減策も検討する。

ここで，クリプト・インベントリとは利用している暗号モジュールや暗号方式のリストのことであり，その作成においては，既に管理簿や仕様書等が存在する場合はそれを利用することができる。また，管理簿や仕様書等が存在しない場合は，何らかの自動化ツールを使うことが，効率の観点からもミスを減らす面からも望ましい。そのような自動化ツールの利用を検討する上では，NIST NCCoE の検討 [9] が参考になる。

CRQC による攻撃リスクの評価においては，CRQC による攻撃が成功した場合の影響，暗号方式によって保護される情報の保護期間（X の把握のために必要），情報システムで利用する各暗号モジュールの移行に要する時間（Y），CRQC を利用する攻撃を行うための前提条件の難易度（攻撃対象である暗号化データ取得の難易度や，そのデータを利用した攻撃の難易度）等の把握が有用である。

この優先順位付けに先駆けて，過剰な保護期間が設定されている情報の保管期間短縮，不要な情報の消去，公開可能な情報の公開等を併せて実施する事も望ましい。このような処理により，X の短縮が期待でき，暗号移行の対象となるシステムを削減する効果が期待される。

暗号移行に際しては，速やかに PQC に暗号移行するというアプローチと，あらかじめクリプトグラフィック・アジリティ [12][*11]を向上させつつ，ある程度以上のクリプトグラフィック・アジリティを達成した上で暗号移行するというアプローチが存在する。

クリプトグラフィック・アジリティが向上すると，Y や X の値が小さくなる。このため，例えば，PQC の評価が十分にされておらず，暗号移行開始の妨げとなっている期間においては，当面の間はクリプトグラフィック・アジリティ向上に努めるというアプローチも一定の合理性があるものと考えられる [26]。なお，クリプトグラフィック・アジリティ向上に伴う Y や X の短縮により，該当する情報システムのプライオリティが下がれば，他の相対的にプライオリティが高くなった情報システムへリソースを集中させることも可能となる。

本節では，インベントリ管理，不要な情報の消去，クリプトグラフィック・アジリティの確保等について概説したが，こららのアプローチは，一般的な情報セキュリティの文脈でも有効である。そのため，これらのアプローチの検討においては，PQC への暗号移行の文脈における効果のみを念頭に検討するのではなく，他のセキュリティ上の恩恵も併せて視野に入れて検討を行うべきであろう [25]。

2.3.2 クリプトグラフィック・アジリティの重要性

クリプトグラフィック・アジリティは，文脈によって捕捉範囲が異なり，それに伴って異なる意味合いを持つことがある [2]。しかしながら，それらに通底しているのは，暗号アルゴリズムや暗号プロトコルをより迅速に変更できる性質が挙げられる。

暗号移行においては，暗号移行の対象となる情報システムの暗号部分が，情報システムにハードコードされている場合には，暗号アルゴリズムの変更が困難である。このような状態は「クリプトグラフィック・アジリティを持たない」と表現することができる。

他方で，標準プロトコルを採用する情報システム，暗号モジュールにも標準プロトコルを利用している情報システム，その API が適切に定義されている情報システム，相互運用性が確保されている情報システム，及び暗号回路を含むファームウェアアップデートをオンラインで実施できるように設計されている情報システム等では，その暗号移行に要する時間は比較的短くなり，$X + Y > Z$ となる可能性も低くなる。X 及び Y の値が十分に低く，所定の目標期間以内に暗号移行が可能なシステムは，「クリプトグラフィック・アジリティを持つ」と表現することができる [2, 25]。

クリプトグラフィック・アジリティを持たせるための対応は，PQC の実装とは独立して実施することが可能である [3]。また，より短い期間での暗号移行を行うことが可能となれば，移行プロセスを開始するまでの猶予期間（$Z-X-Y$）をより長くすることが期待される。より長い猶予期間での暗号移行が可能となれば，該当猶予期間を CRQC の開発動向調査等に充てることが可能となり，より適切なタイミングでの移行が期待できる。

なお，クリプトグラフィック・アジリティが十分向上した情報システムにおいては，暗号機能以外を変更する場合の対応速度も高くなることが期待できる。すなわち，セキュリティ上の脆弱性が発覚した場合の対応速度や，ビジネス環境に合わせたサービス変更の対応速度も高いことも期待される [25]。

以上を踏まえ，PQC への暗号移行を実施するにあたっては，まずは暗号移行を長期化する要素を排除することを試み，情報システムにおける暗号プロトコルの変更をより迅速にできるようにシフトさせていく対策，すなわちクリプトグラフィック・アジリティを確保する対策を併せて実施することが効果的である [2, 3, 25]。

[*11] 暗号方式を変更可能とする性質。2.3.2 節参照。

2.3.3 既存暗号方式とのハイブリッド構成

暗号移行においては，ハイブリッド構成を採用することができる。PQC への暗号移行の文脈におけるハイブリッド構成とは，既存の公開鍵暗号と，PQC の両方を利用することによって何らかの目標の達成を目指すものであるが，厳密な定義は見当たらない [8]。ハイブリッド構成の目標は，暗号アルゴリズムの切替期間中における相互運用性の確保や，既存暗号方式しか利用できない機器に対する後方互換性の確保であることもあれば，両方のアルゴリズムのうち片方が危殆化した場合の安全性の維持であることもある。

また，ハイブリッドという用語は，単一の暗号モジュールを構成するコンポジット方式 [13, 16] の文脈で使用されることもあれば，複数の暗号モジュールの出力を入力として受け取り，新たな出力を生成するコンバイナー構造に対して使用されることもある [8]。

なお，IETF の標準化活動において，ハイブリッド構成による鍵共有方式に関しては一定の合意が見受けられるが [13, 16]，ハイブリッド構成によるデジタル署名方式 [17] に関しては合意に時間を要している。

2.3.4 署名用途固有の対策

署名用途の公開鍵暗号は様々なユースケースで利用されるが，PKI 等のインフラの移行に要する時間（Y）やコードサイン証明書が利用される期間（X）が比較的長いことから，速やかな PQC への暗号移行が困難である。この場合においても，以下の対応を取ることが望ましい。

PKI においては，一般に Y が長くなる傾向にあるが，電子証明書の有効期間の短縮や，1 枚の電子証明書に対して（既存暗号方式と署名用途の PQC の）2 つの公開鍵及びデジタル署名を付与する方式などを採用することで，Y の短縮が期待できる [21]。なお，後者の 2 つの公開鍵暗及びデジタル署名を利用する方式においては，実装やポリシー管理の複雑さが大きく増加することから，注意を必要とする。

X を実質的に短縮する技術として，タイムスタンプ更新技術が存在する。例えば，ERS[10] 等を利用することで，タイムスタンプの更新や，暗号方式の更新が可能となる。Z が経過する前に，既存の公開鍵暗号を PQC に更新することが可能であれば，X,Y,Z の関係によらず，データは保護される。ただし，このアプローチでは，データ構造の複雑さが増加する傾向があり，（PQC への即時の暗号移行に比べては小さいものの）情報システムの運用費用が増加する点には注意を必要とする。

2.3.5 守秘及び鍵共有用途固有の対策

既に述べたように，耐量子計算機性を持たせるための一般的な対策は，既存の暗号方式を耐量子計算機性を持つ公開鍵暗号方式に移行することである。ここで，2.2.2 節及び 2.2.3 節で述べたとおり，守秘及び鍵共有用途で保護されたコンテンツや鍵情報は，保護期間が非常に長いことや，場合によっては無期限で保護されることも考えられる。このような情報に対するハーベスト攻撃の脅威を考慮すると，当該情報は，将来における CRQC による解読リスクに既に晒されていることから，一刻も早く耐量子計算機性を持たせる対応を始めることが望ましい。ただし，全ての守秘及び鍵共有用途の公開鍵暗号を移行するためには非常に大きなリソースが要求され，現実的なコストでは実現が困難であるおそれがある。

このような状況においても，守秘及び鍵共有用途固有の対策を効率的に行う方法 [26] として，以下のアプローチがある。

Z に対して X + Y の値が非常に小さく，X + Y $\ll$ Z と予測される暗号文に対しては，CRYPTREC による注意喚起情報 [6] に注意を払いつつ，現在用いている暗号の使用を継続する。また，X + Y $>$ Z となることが十分予想される暗号文に対しては，2.3 節前段で述べた，PQC への暗号移行や，暗号文の保護期間である X の短縮，情報システムの

暗号処理の実装の置き換えに要する期間 Y の短縮を行う。その結果，X や Y の値を十分に小さくすることができるのであれば，現在用いている暗号方式の使用を継続する。

一方で，X + Y > Z と予想される，又は，X + Y > Z となることが避けられない暗号文に対してしては，暗号システムの PQC への暗号移行を進めつつも，既存の公開鍵暗号によって保護されている暗号文は公開ネットワーク等に保管せず，適切にアクセスコントロールを行う。

なお，現在 DH を利用している場合は，2.2.3 節で述べたような検討を行い，DH 固有の性質が必要か否かをあらかじめ検討することが望ましい。

2.4 PQC の活用にむけて

PQC への暗号移行においては，どのようなデータに対して，どのような暗号技術を利用しているのかを把握することが第一歩となる。また，保護対象となるデータの保護期間等をあらかじめ把握しておくことで，より効率的な対応ができる [26]。その上で，公開できるデータは公開し，破棄可能なデータは破棄することも検討すべきである。この検討を進めることで，クリプトグラフィック・アジリティ [12] の確保も見込まれ，より効果的な PQC への移行が期待できる。CRQC の脅威への対策を検討するにあたっては，保護されている情報の価値，CRQC による攻撃が成功した場合の影響，図 2.1 における X,Y,Z の関係等を踏まえ，プライオリティを付けて，そのプライオリティ順に対策を実施することが望ましい [27, 25]。

第 2 章の参照文献

[1] National Security Agency. The Commercial National Security Algorithm Suite 2.0 and Quantum Computing FAQ. 2024-04. https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/1/CSI_CNSA_2.0_FAQ_.PDF. (2025-01-06 閲覧).

[2] N. Alnahawi, N. Schmitt, A. Wiesmaier, A. Heinemann, T. Grasmeyer. On the State of Crypto-Agility. Cryptology ePrint Archive, Paper 2023/487. 2023. https://eprint.iacr.org/2023/487.

[3] A. Amadori et al. The PQC Migration Handbook. https://publications.tno.nl/publication/34643386/fXcPVHsX/TNO-2024-pqc-en.pdf. 2024-12. (2025-01-06 閲覧).

[4] S. Boeyen, S. Santesson, T. Polk, R. Housley, S. Farrell, D. Cooper. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, https://www.rfc-editor.org/info/rfc5280. 2008-05. (2023-04-12 閲覧).

[5] CRYPTREC. TLS 暗号設定ガイドライン. CRYPTREC GL-3001-3.0.1, https://www.cryptrec.go.jp/report/cryptrec-gl-3001-3.0.1.pdf. 2020-07.

[6] CRYPTREC. 注意喚起一覧. https://www.cryptrec.go.jp/er.html. (2024-03-05 閲覧).

[7] CRYPTREC 暗号技術評価委員会. 注意喚起情報 “現在の量子コンピュータによる暗号技術の安全性への影響”. https://www.cryptrec.go.jp/topics/cryptrec-er-0001-2019.html.

[8] F. Driscoll, M. Parsons, B. Hale. Terminology for Post-Quantum Traditional Hybrid Schemes. Internet-Draft. 2024-12. https://datatracker.ietf.org/doc/draft-ietf-pquip-pqt-hybrid-terminology/05/. (2025-02-20 閲覧).

[9] NIST National Cyersecurity Center of Excellence. Migration to Post-Quantum Cryptography Quantum Readiness: Cryptographic Discovery. NIST SP 1800-38B (initial preliminary draft), https://www.nccoe.nist.gov/sites/default/files/2023-12/pqc-migration-nist-sp-1800-38b-preliminary-draft.pdf. 2023-12. (2025-02-17 閲覧).

[10] T. Gondrom, R. Brandner, U. Pordesch. Evidence Record Syntax (ERS). RFC 4998, https://www.rfc-editor.org/info/rfc4998. 2007-08. (2023-04-12 閲覧).

[11] P. E. Hoffman. DNS Security Extensions (DNSSEC). RFC 9364, https://www.rfc-editor.org/info/rfc9364. 2023-02.

[12] R. Housley. Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms. RFC 7696, https://www.rfc-editor.org/info/rfc7696. 2015-11. (2023-04-12 閲覧).

[13] K. Kwiatkowski, P. Kampanakis, B. Westerbaan, D. Stebila. Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLSv1.3. Internet-Draft. 2024-12. https://datatracker.ietf.org/doc/draft-kwiatkowski-tls-ecdhe-mlkem/03/. (2025-02-20 閲覧).

[14] M. Lepinski, S. Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480, https://www.rfc-editor.org/info/rfc6480. 2012-02. (2025-01-15 閲覧).

[15] M. Mosca. Cybersecurity in a quantum world: will we be ready? Workshop on Cybersecurity in a Post-Quantum World. Session 8. 2015-04. (2024-02-29 閲覧).

[16] M. Ounsworth, J. Gray. Composite KEM For Use In Internet PKI. Internet-Draft. 2024-10. https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-kem/. (2025-01-06 閲覧).

[17] M. Ounsworth, J. Gray, M. Pala, J. Klaußner, S. Fluhrer. Composite ML-DSA For use in X.509 Public Key Infrastructure and CMS. Internet-Draft. 2024-10. https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-sigs/03/. (2025-01-15 閲覧).

[18] E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, https://www.rfc-editor.org/info/rfc8446. 2018-08. (2023-04-12 閲覧).

[19] Y. Sheffer, R. Holz, P. Saint-Andre. Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). RFC 7525, https://www.rfc-editor.org/info/rfc7525. 2015-05. (2023-04-12 閲覧).

[20] P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. Vol. 26, Num. 5 (1997), pp. 1484–1509.

[21] D. Stebila, S. Fluhrer, S. Gueron. Hybrid key exchange in TLS 1.3. Internet-Draft. 2024-10. https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/11/. (2025-02-20 閲覧).

[22] D. Stebila, S. Fluhrer, S. Gueron. Hybrid key exchange in TLS 1.3. Internet-Draft. 2025-01. https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/12/. (2025-02-20 閲覧).

[23] R. Zuccherato, P. Cain, Dr. C. Adams, D. Pinkas. Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161, https://www.rfc-editor.org/info/rfc3161. 2001-08. (2023-04-12 閲覧).

[24] デジタル庁, 総務省, 経済産業省. 電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）. CRYPTREC LS-0001-2022r1, https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2022r1.pdf. 2024-05.

[25] 伊藤 忠彦. 耐量子計算機暗号への移行へ向けた課題と社会実装への論点整理. 電子情報通信学会誌. Vol. 106, Num. 11 (2023), pp. 1026–1030.

[26] 伊藤 忠彦, 宇根 正志, 清藤 武暢. 量子コンピュータによる脅威を見据えた暗号の移行対応. 2019-08. https://www.imes.boj.or.jp/research/papers/japanese/19-J-15.pdf. (2025-01-06 閲覧).

[27] 預金取扱金融機関の耐量子計算機暗号への対応に関する検討会. 預金取扱金融機関の耐量子計算機暗号への対応に関する検討会報告書. 2024-11. https://www.fsa.go.jp/singi/pqc/houkokusyo.pdf. (2025-01-06 閲覧).

第 3 章

格子に基づく暗号技術

本章では格子に基づく暗号技術についてまとめる。格子に基づく暗号技術の安全性は，LWE (Learning with Errors) 問題，LWR (Learning with Rounding) 問題，NTRU 問題，およびそれらの変種等を含む格子理論に関係する問題を解く計算の困難性に依存している。

3.1 格子に基づく暗号技術の安全性の根拠となる問題

3.1.1 LWE 問題と代表的な求解法

本節では，2005 年 Regev が提案した LWE 問題 [131] を紹介すると共に，格子を利用した LWE 問題に対する求解法を紹介する。また，LWE 問題のいくつかの変種についても言及する。

3.1.1.1 LWE 問題の紹介

LWE 問題は機械学習理論から派生した求解困難な問題で，整数剰余類環 $\mathbb{Z}_q$ 上の秘密ベクトル $\mathbf{s} \in \mathbb{Z}_q^n$ に関するランダムな連立線形「近似」方程式が与えられたとき，その秘密ベクトルを復元する問題である。具体的な数値例として $n = 4$, $q = 17$ に対して，秘密ベクトル $\mathbf{s} = (s_1, s_2, s_3, s_4) \in \mathbb{Z}_{17}^4$ に関する連立線形近似方程式

$$\begin{cases} 14s_1 + 15s_2 + 5s_3 + 2s_4 \approx 8 \pmod{17} \\ 13s_1 + 14s_2 + 14s_3 + 6s_4 \approx 16 \pmod{17} \\ 6s_1 + 10s_2 + 13s_3 + s_4 \approx 12 \pmod{17} \\ \qquad\qquad\qquad\qquad \vdots \\ 6s_1 + 7s_2 + 16s_3 + 2s_4 \approx 3 \pmod{17} \end{cases}$$

が与えられたとする。(この数値例は [133] から引用した。) ただし，各線形方程式の値は近似値であり，その誤差はこの例では ± 1 以内と仮定する。このとき，この連立線形近似方程式の解 $\mathbf{s}$ を求めるのが LWE 問題である。ここに示した数値例では $\mathbf{s} = (0, 13, 9, 11) \in \mathbb{Z}_{17}^4$ が解となる。LWE 問題で注意すべきことは，連立線形近似方程式に誤差がない場合は，Gauss の消去法により効率的に解を求めることができる点である。逆に言うと，連立線形近似方程式で与えられる誤差の大きさが LWE 問題の求解を困難にする。

■ 離散 Gauss 分布 一般に，LWE 問題における連立線形近似方程式の誤差は，平均 0，パラメータ $\sigma > 0$ の $\mathbb{Z}$ 上の離散 Gauss 分布 $\chi = D_{\mathbb{Z},\sigma}$ から生成される[*1]。より正確には，χ は各整数 x がサンプルされる確率が $\exp\left(-\dfrac{\pi x^2}{\sigma^2}\right)$ に比例する $\mathbb{Z}$ 上の離散確率分布である。この分布は，数学的な正規分布[*2] とは異なるが，絶対値の大きな値が生成され

[*1] 本章では，記号 σ を Gauss 分布のパラメータ (標準偏差とは異なる) の意味で使い，署名を表すときには sig を用いる。

[*2] 分散 t^2 に対して数学的な正規分布 $N(0, t^2)$ は，確率密度関数が $\dfrac{1}{\sqrt{2\pi}t} e^{-z^2/(2t^2)}$ により定義されるため，$\sqrt{2\pi}$ 倍のずれがある。暗号の安全性を議論する際に格子上のフーリエ変換が用いられることが多く [131]，本文中の定義を用いることで，数式の表現が簡潔となる。

る確率が非常に小さいという性質は共通している。例えば，絶対値が 3σ より大きな整数がサンプルされる確率は非常に小さい。離散 Gauss 分布の詳細については [100] などを参照。

離散 Gauss 分布を厳密に実装するのは容易ではなく，timing attack などの脆弱性 [39] が生まれてしまう。現実の方式 (3.3 節参照) においては，誤差 (ノイズ) として離散 Gauss 分布との統計距離が小さい分布を用いている。それらと区別するため，方式 Scheme 内で用いられるノイズの分布を $D^{\mathsf{Scheme}}_{\mathbb{Z},s}$ と表現する。ここで，s はパラメータである。また，記号 $D^{\mathsf{Scheme}}_{\mathbb{Z}^n,s}, D^{\mathsf{Scheme}}_{\mathbb{Z}^{n\times m},s}$ を，それぞれ成分を $D^{\mathsf{Scheme}}_{\mathbb{Z},s}$ から独立に生成した n 次元ベクトル，$n\times m$ 行列とする。

■ **LWE 問題の定式化**　以下は，定式化された LWE 問題である：

定義 3.1 (LWE 問題 [131]) n を正の整数とし，q を奇素数とする。平均 0，パラメータ σ の $\mathbb{Z}$ 上の離散 Gauss 分布を $\chi = D_{\mathbb{Z},\sigma}$ とする。秘密ベクトル $\mathbf{s}\in\mathbb{Z}_q^n$ を固定する。一様ランダムに選ばれた $\mathbf{a}\in\mathbb{Z}_q^n$ と離散 Gauss 分布 χ からサンプルされた $e\in\mathbb{Z}$ に対して，$(\mathbf{a},b)\in\mathbb{Z}_q^n\times\mathbb{Z}_q$ の組を出力する確率分布を $L_{\mathbf{s},\chi}$ とする。ただし，$b\equiv\langle\mathbf{a},\mathbf{s}\rangle+e \pmod q$ とする。(2 つのベクトル $\mathbf{v}$ と $\mathbf{w}$ の内積を $\langle\mathbf{v},\mathbf{w}\rangle$ で表す。) このとき，次の 2 つの問題を考える：

1. **判定** LWE (Decision-LWE)　与えられた組 $(\mathbf{a},b)\in\mathbb{Z}_q^n\times\mathbb{Z}_q$ が，確率分布 $L_{\mathbf{s},\chi}$ からサンプルされた元か，$\mathbb{Z}_q^n\times\mathbb{Z}_q$ 上一様ランダムに生成された元かを決定する問題。
2. **探索** LWE (Search-LWE)　確率分布 $L_{\mathbf{s},\chi}$ からサンプルされた組 $(\mathbf{a},b)$ から秘密ベクトル $\mathbf{s}$ を復元する問題。

一般に，ここに示した 2 つの LWE 問題において確率分布 $L_{\mathbf{s},\chi}$ は任意個の組 $(\mathbf{a},b)$ をサンプルするオラクルとしてみなす。具体的には，ある固定したサンプル数 $m>0$ に対して，確率分布 $L_{\mathbf{s},\chi}$ からサンプルされた異なる m 個の組

$$\begin{cases}(\mathbf{a}_1,b_1), & b_1\equiv\langle\mathbf{a}_1,\mathbf{s}\rangle+e_1 \pmod q\\ (\mathbf{a}_2,b_2), & b_2\equiv\langle\mathbf{a}_2,\mathbf{s}\rangle+e_2 \pmod q\\ \quad\vdots & \\ (\mathbf{a}_m,b_m), & b_m\equiv\langle\mathbf{a}_m,\mathbf{s}\rangle+e_m \pmod q\end{cases}$$

から LWE 問題を解くことを考える。(解読に要する計算時間が最も短くなるような m を攻撃者が選べることを想定する。) 第 i 行ベクトルを $\mathbf{a}_i$ とする $m\times n$ 行列を $\mathbf{A}$ とし，$\mathbf{b}=(b_1,b_2,\ldots,b_m)$ とおく。このとき，ここに示した m 個の LWE サンプルの組は $(\mathbf{A},\mathbf{b})\in\mathbb{Z}_q^{m\times n}\times\mathbb{Z}_q^n$ と簡潔に表せて，関係式 $\mathbf{b}\equiv\mathbf{s}\mathbf{A}^\top+\mathbf{e} \pmod q$ を満たす。ただし，$\mathbf{e}=(e_1,e_2,\ldots,e_m)\in\mathbb{Z}^m$ をノイズベクトルとする。(各 e_i は χ からサンプルされた元であることに注意する。)

■ **LWE 問題の変種**　LWE 問題の変種として，多項式環 $R_q=\mathbb{Z}_q[x]/(\phi)$ 上の LWE である Ring-LWE [147, 106][*3] や Module-LWE [96] がある。Ring-LWE では，3 つの多項式 $s,a_i,e_i\in R_q$ に対する Ring-LWE サンプルとして $\{(a_i,a_i\cdot s+e_i)\}_{i=1}^m$ を考える。(特に，通常の LWE 問題と同じように，ランダムな s と，係数が小さい多項式の集合からサンプリングされた e_i が用いられる。) Ring-LWE の基礎環 R_q を定める多項式として，2 のべき乗の形をした整数 n に対し $\phi=x^n+1$ がよく用いられる。また，Module-LWE では，多項式ベクトル $\mathbf{s},\mathbf{a}_i\in R_q^k$ と多項式 $e_i\in R_q$ に対する Module-LWE サンプルとして $\{(\mathbf{a}_i,\langle\mathbf{a}_i,\mathbf{s}\rangle+e_i)\}_{i=1}^m$ を考える。Module-LWE の基礎環 R_q を定める多項式としては $\phi=x^{n/k}+1$ がよく用いられる。さらに，環上の LWE 以外の LWE 問題の変種として，丸め込み（rounding）でノイズベクトルを生成する LWR[25] や middle-product と呼ばれる多項式演算を用いる Middle-product LWE [134] など数多くの変種が提案されている。

[*3] 文献 [106] ではより一般的に整数環とイデアルを用いて定義されているが，後の文献 [38] ではそれの簡略化として，多項式環 R_q を用いた表現である “polynomial-LWE assumption” が提案された。現在では，後者の表現の方が Ring-LWE と呼ばれている。

3.1.1.2 格子の基本事項と q-ary 格子の紹介

■ 格子の基本事項 m 次元実ベクトル空間 $\mathbb{R}^m$ の一次独立な m 個のベクトル $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$ の整数係数の線形結合全体 $L = \{\sum_{i=1}^{m} x_i \mathbf{b}_i : x_i \in \mathbb{Z}, 1 \leq i \leq m\}$ を (完全階数の) m 次元**格子**と呼ぶ。特に，格子 L はベクトル空間 $\mathbb{R}^m$ の (離散) 加法部分群である。また，格子 L を生成する一次独立な m 個のベクトルの組 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ を**基底**と呼び，各 $\mathbf{b}_i$ を**基底ベクトル**と呼ぶ。さらに，行ベクトルで表した基底ベクトル $\mathbf{b}_i \in \mathbb{R}^m$ を行として持つ $m \times m$ 行列 $\mathbf{B} = (\mathbf{b}_i)_{i=1}^{m}$ を格子 L の**基底行列**と呼ぶ。2 次元以上の格子を生成する異なる基底は無限に存在し，同じ格子を生成する 2 つの基底行列 $\mathbf{B}_1$ と $\mathbf{B}_2$ に対し $\mathbf{B}_2 = \mathbf{V}\mathbf{B}_1$ を満たす $m \times m$ のユニモジュラ行列 $\mathbf{V}$ が存在する。また，基底行列 $\mathbf{B}$ を用いて，格子 L の**体積**を $\mathrm{vol}(L) = |\det(\mathbf{B})|$ と定める。(体積は基底の取り方に依存しない。) 格子 L の**第 1 逐次最小**は L 上の最短な非零ベクトルの Euclid ノルムを指し，$\lambda_1(L)$ と表す。ベクトル空間 $\mathbb{R}^m$ の完全階数の格子 L に対し，集合 $\widehat{L} = \{\mathbf{x} \in \mathbb{R}^m : \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z} \quad (\forall \mathbf{y} \in L)\}$ を格子 L の**双対格子**と呼ぶ。また，格子 L の基底行列 $\mathbf{B}$ に対して，$\widehat{\mathbf{B}} = \left(\mathbf{B}^{-1}\right)^{\top}$ は双対格子 $\widehat{L}$ の基底行列となり，この $\widehat{\mathbf{B}}$ を**双対基底行列**と呼ぶ。単位行列 $\mathbf{I}_m$ に対し $\mathbf{B}\widehat{\mathbf{B}}^{\top} = \mathbf{I}_m$ を満たすので，$\mathrm{vol}(L) \times \mathrm{vol}(\widehat{L}) = 1$ が成り立つ。

■ q-ary 格子 ここでは，LWE 問題の求解で利用する特殊な格子を紹介する。正の整数 q に対して，$q\mathbb{Z}^m \subseteq L \subseteq \mathbb{Z}^m$ を満たす完全階数の m 次元格子 L を **q-ary 格子**と呼ぶ。2 つの自然数 $m > n$ に対し，任意の正の整数 q と $n \times m$ 整数行列 $\mathbf{X}$ に対する 2 つの m 次元 q-ary 格子を

$$\Lambda_q(\mathbf{X}) = \{\mathbf{y} \in \mathbb{Z}^m : \exists \mathbf{s} \in \mathbb{Z}^n \text{ s.t. } \mathbf{y} \equiv \mathbf{s}\mathbf{X} \pmod{q}\}, \quad \Lambda_q^{\perp}(\mathbf{X}) = \left\{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y}\mathbf{X}^{\top} \equiv \mathbf{0} \pmod{q}\right\}$$

と定義する。(これらの集合は $\mathbb{R}^m$ の離散加法部分群なので格子である。) 正規化の差を除き，これら 2 つの q-ary 格子は互いに双対の関係にある。正確には $\Lambda_q^{\perp}(\mathbf{X}) = q\widehat{\Lambda_q(\mathbf{X})}$ と $\Lambda_q(\mathbf{X}) = q\widehat{\Lambda_q^{\perp}(\mathbf{X})}$ が成り立つ。また，群準同型写像 $f : \mathbb{Z}^m \longrightarrow (\mathbb{Z}_q)^n, \quad \mathbf{y} \longmapsto \mathbf{y}\mathbf{X}^{\top} \pmod{q}$ の核は q-ary 格子 $\Lambda_q^{\perp}(\mathbf{X})$ なので，群の準同型定理から $\mathrm{vol}(\Lambda_q^{\perp}(\mathbf{X})) = [\mathbb{Z}^m : \Lambda_q^{\perp}(\mathbf{X})] = \#\mathrm{Im}(f)$ が成り立つ。(群の指数 $[\mathbb{Z}^m : \Lambda_q^{\perp}(\mathbf{X})]$ は格子の体積の比 $\dfrac{\mathrm{vol}(\Lambda_q^{\perp}(\mathbf{X}))}{\mathrm{vol}(\mathbb{Z}^m)}$ に一致することに注意する。) これより，体積 $\mathrm{vol}\left(\Lambda_q^{\perp}(\mathbf{X})\right)$ は q^n を割る。さらに，元の格子と双対格子の体積の関係から，q^{m-n} は体積 $\mathrm{vol}\left(\Lambda_q(\mathbf{X})\right)$ を割ることが分かる。(ただし，ほとんどの行列 $\mathbf{X}$ に対して写像 f は全射で，その時 $\mathrm{vol}\left(\Lambda_q^{\perp}(\mathbf{X})\right) = q^n$ と $\mathrm{vol}\left(\Lambda_q(\mathbf{X})\right) = q^{m-n}$ が成り立つ。) q-ary 格子 $\Lambda_q(\mathbf{X})$ 上のベクトルは $\mathbf{y} = \mathbf{s}\mathbf{X} + q\mathbf{z}$ $(\mathbf{s} \in \mathbb{Z}^n, \mathbf{z} \in \mathbb{Z}^m)$ とかけるので，その格子は $(n+m) \times m$ 整数行列 $\begin{pmatrix} \mathbf{X} \\ q\mathbf{I}_m \end{pmatrix}$ の一次従属な $(n+m)$ 個の行ベクトルで生成される。この生成行列の Hermite Normal Form を計算することで，m 次元 q-ary 格子 $\Lambda_q(\mathbf{X})$ の基底行列 $\mathbf{B} \in \mathbb{Z}^{m \times m}$ が得られる。また，双対基底の性質から，もう片方の q-ary 格子 $\Lambda_q^{\perp}(\mathbf{X})$ の基底行列は $(q\mathbf{B}^{-1})^{\top} \in \mathbb{Z}^{m \times m}$ で得られる。

3.1.1.3 LWE 問題の代表的な求解法

格子上の計算問題である格子問題として，最短ベクトル問題（Shortest Vector Problem，SVP）や最近ベクトル問題（Closest Vector Problem，CVP）などが代表的である。ここでは，LWE 問題の格子問題への帰着を述べる。

■ 判定 LWE 問題に対する求解 判定 LWE 問題を SIS (Short Integer Solution) 問題に帰着して解く方法を紹介する：正の整数 q と，$0 < \beta < q$ を満たす実数 β を固定する。各成分が剰余環 $\mathbb{Z}/q\mathbb{Z}$ 上一様ランダムに選ばれた $n \times m$ 整数行列 $\mathbf{X}$ に対して，$\|\mathbf{v}\| \leq \beta$ かつ $\mathbf{v}\mathbf{X}^{\top} \equiv \mathbf{0} \pmod{q}$ を満たす非零ベクトル $\mathbf{v} \in \mathbb{Z}^m$ を見つける問題を **SIS 問題**と呼ぶ。つまり，これは q-ary 格子 $\Lambda_q^{\perp}(\mathbf{X})$ 上の短い非零ベクトルを見つける問題である。剰余パラメータ q における LWE 問題のサンプル数を m とし，m 個の LWE サンプルの組を $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ とする。ここで，$n \times m$ の転置行列 $\mathbf{A}^{\top}$ に対する SIS 問題の短い解ベクトル $\mathbf{v} \in \Lambda_q^{\perp}(\mathbf{A}^{\top})$ が得られたとする $(0 < \|\mathbf{v}\| \leq \beta$ と仮定)。このとき，LWE サンプルの組 $(\mathbf{A}, \mathbf{b})$ は関係式 $\mathbf{b} \equiv \mathbf{s}\mathbf{A}^{\top} + \mathbf{e} \pmod{q}$ を満たすので，$\langle \mathbf{v}, \mathbf{b} \rangle \equiv \langle \mathbf{v}, \mathbf{s}\mathbf{A}^{\top} + \mathbf{e} \rangle \equiv \langle \mathbf{v}\mathbf{A}, \mathbf{s} \rangle + \langle \mathbf{v}, \mathbf{e} \rangle \equiv \langle \mathbf{v}, \mathbf{e} \rangle$

$(\mathrm{mod}\ q)$ が成り立つ ($\mathbf{vA} \equiv \mathbf{0}\ (\mathrm{mod}\ q)$ に注意) 。さらに，ノイズベクトル $\mathbf{e}$ のすべての成分 e_i は離散 Gauss 分布 χ からサンプルされた元なので，$|\langle \mathbf{v}, \mathbf{e} \rangle| \lesssim \sigma\sqrt{m}\|\mathbf{v}\| \leq \sigma\beta\sqrt{m}$ が期待できる。(離散 Gauss 分布 $\chi = D_{\mathbb{Z},\sigma}$ のサンプル元 e_i の絶対値はおおよそ σ 未満で，多めに見積もって $\|\mathbf{e}\| \lesssim \sigma\sqrt{m}$ とした。) ゆえに，$\sigma\beta\sqrt{m} \ll q$ ならば，$|\langle \mathbf{v}, \mathbf{b} \rangle|$ $(\mathrm{mod}\ q)$ の値の大きさから LWE サンプルの組 $(\mathbf{A}, \mathbf{b})$ は確率分布 $L_{\mathbf{s},\chi}$ からサンプルされたものか判定できる。

■ **探索 LWE 問題に対する求解法**　探索 LWE 問題を BDD (Bounded Distance Decoding) 問題に帰着して解く方法を紹介する：格子 L と目標ベクトル $\mathbf{w}$ に対し，ある $0 < \mu \leq \frac{1}{2}$ が存在し $\mathrm{dist}(\mathbf{w}, L) = \min_{\mathbf{v} \in L} \|\mathbf{w} - \mathbf{v}\| < \mu\lambda_1(L)$ を満たすと仮定する。格子 L の基底が与えられたとき，目標ベクトル $\mathbf{w}$ に最も近い格子ベクトル $\mathbf{v} \in L$ を見つける問題を **BDD 問題**と呼ぶ。m 個の LWE サンプルの組 $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ は関係式 $\mathbf{b} \equiv \mathbf{sA}^\top + \mathbf{e}\ (\mathrm{mod}\ q)$ を満たすので，探索 LWE 問題は $\mathbf{b}$ を目標ベクトルとする q-ary 格子 $\Lambda_q(\mathbf{A}^\top)$ 上の BDD 問題とみなせる。実際，目標ベクトル $\mathbf{b} = \mathbf{sA}^\top + \mathbf{e} + q\mathbf{z}\ (\exists \mathbf{z} \in \mathbb{Z}^m)$ に対して，格子ベクトルを $\mathbf{v} = \mathbf{sA}^\top + q\mathbf{z} \in \Lambda_q(\mathbf{A}^\top)$ とおくと，$\mathbf{b} - \mathbf{v} = \mathbf{e}$ が成り立つ。ノイズベクトル $\mathbf{e}$ のすべての成分 e_i は離散 Gauss 分布 χ からサンプルされた元であるため，分散と次元が大きい場合にはおおよそスケーリングされたカイ二乗分布に従い，高い確率で $\|\mathbf{e}\| \approx \frac{\sigma}{\sqrt{2\pi}} \cdot \sqrt{m}$ となる。ゆえに，目標ベクトル $\mathbf{b}$ との距離が $\sigma\sqrt{m}$ 以下となる q-ary 格子 $\Lambda_q(\mathbf{A}^\top)$ 上の格子ベクトル $\mathbf{v}$ を見つけることで，ノイズベクトル $\mathbf{e}$ を復元することができる。実用的には，Kannan や Bai-Galbraith らの埋め込み法 [93, 22] により，BDD 問題を unique-SVP 問題に帰着してからノイズベクトル $\mathbf{e}$ を復元する。

注意 3.2 (LWE 問題の変種に対する求解) LWE 問題の代表的な変種である Ring-LWE や Module-LWE では，上述したように多項式環 $R_q = \mathbb{Z}_q[x]/(\phi)$ を基礎環として利用する。n 次多項式 ϕ に対して，基礎環 $R_q = \mathbb{Z}_q[x]/(\phi)$ の任意の元は $n-1$ 次以下の多項式 $f = f_0 + f_1 x + \cdots + f_{n-1}x^{n-1}\ (f_i \in \mathbb{Z}_q)$ と表せ，その係数ベクトル $\mathbf{f} = (f_0, f_1, \ldots, f_{n-1}) \in \mathbb{Z}_q^n$ と一対一に対応する。このように，基礎環 R_q の元をその係数ベクトルに対応させることで，Ring-LWE や Module-LWE 問題は通常の LWE 問題と同じようにベクトル・行列の形で表現できる。(詳細は [6] を参照。また，ベクトル・行列の形の表現については，次で説明する NTRU 問題も参照。) ベクトル・行列の形で表現した Ring-LWE や Module-LWE 問題に対して，上述で説明した通常の LWE 問題の求解法が適用できる。

3.1.2　NTRU 問題と代表的な求解法

ここでは，NTRU 問題とその代表的な求解法を紹介する。まず以下で，NTRU 問題について述べる：

定義 3.3 (NTRU 問題 [86]) 2 つの正の整数 n と q に対し，$\phi \in \mathbb{Z}[x]$ を次数 n の多項式とし，$R_q = \mathbb{Z}_q[x]/(\phi)$ とする。係数が小さい 2 つの多項式 $f \in R_q^\times, g \in R_q$ に対して，$h = g \cdot f^{-1} \in R_q$ とする。(特に，f は環 R_q の可逆元に注意。) このとき，与えられた多項式 h から，f または g の多項式を復元する問題を (探索) NTRU 問題という。

NTRU 問題における多項式 ϕ の選び方として，$\phi = x^n \pm 1, x^n - x - 1, x^n - x^{n/2} + 1, \sum_{i=0}^{n-1} x^i$ などがある [8, Table 1]。(最後の ϕ のみ次数は $n-1$ である。) また，多項式 f (または g) の選び方として，$\{-1, 0, 1\}$ などの小さい係数を持つ多項式や，小さい素数 p と係数が小さい多項式 F に対し $f = pF$ または $f = pF + 1$ と選ぶことが多い。

次に，NTRU 問題の代表的な求解法を紹介する。まず，与えられた多項式 $h \in R_q$ に対して，h の回転行列を $\mathbf{H} \in \mathbb{Z}^{n \times n}$ とする。(具体的には，$n \times n$ 整数行列 $\mathbf{H}$ の i 行ベクトルを多項式 $x^{i-1}h \in R_q$ の係数ベクトルとする。) このとき

$$\mathbf{H} \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix} = \begin{pmatrix} h \\ xh \\ \vdots \\ x^{n-1}h \end{pmatrix} \in R_q^n$$

が成り立つ。ここで，$2n \times 2n$ 行列 $\mathbf{B} = \begin{pmatrix} \mathbf{I}_n & \mathbf{H} \\ \mathbf{0} & q\mathbf{I}_n \end{pmatrix}$ の行ベクトルで生成される **NTRU 格子**を L とする。このとき，$2n$ 次元の NTRU 格子 L は短いベクトル $(\mathbf{f} \mid \mathbf{g}) \in \mathbb{Z}^{2n}$ を含む。(ただし，$\mathbf{f}, \mathbf{g} \in \mathbb{Z}^n$ を多項式 $f, g \in R_q$ の係数ベクトルとする。) 実際，$hf = g \pmod q$ より，$g = hf + qr$ を満たす多項式 $r \in R_q$ が存在する。また，多項式 r の係数ベクトルを $\mathbf{r} \in \mathbb{Z}^n$ とすると，

$$\mathbf{g} \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix} = g = hf + qr = \mathbf{f} \begin{pmatrix} h \\ xh \\ \vdots \\ x^{n-1}h \end{pmatrix} + q\mathbf{r} \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix} = (\mathbf{fH} + q\mathbf{r}) \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix} \in R_q$$

となるので，$\mathbf{g} = \mathbf{fH} + q\mathbf{r}$ が成り立つ。これより，$(\mathbf{f} \mid \mathbf{g}) = (\mathbf{f} \mid \mathbf{fH} + q\mathbf{r}) = (\mathbf{f} \mid \mathbf{r})\mathbf{B} \in L$ が成り立つ。(つまり，ベクトル $(\mathbf{f} \mid \mathbf{g})$ が NTRU 格子 L に含まれる。) ベクトル $(\mathbf{f} \mid \mathbf{g}) \in \mathbb{Z}^{2n}$ が十分小さく NTRU 格子 L 上の最短ベクトルと仮定すると，これは NTRU 問題を SVP に帰着できることを示している。(最短ベクトル $(\mathbf{f} \mid \mathbf{g})$ の各ブロックにおける回転で得られるベクトルも NTRU 格子 L に含まれるので，一般に NTRU 格子 L は複数の最短ベクトルを含む。)

3.1.3 格子問題を解くアルゴリズムとその計算量について

SVP・CVP の格子問題やここまでに紹介した LWE 問題・NTRU 問題などの格子問題を解くのに有用な技術として**格子基底簡約**がある。格子基底簡約は，与えられた格子 L の基底から，各ベクトル $\mathbf{b}_i$ が短く・互いのベクトルが直交に近い格子 L の新しい基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ を見つける操作である。(明確な定義はないが，このような基底を「簡約基底」または「良い基底」と呼ぶ。)

3.1.3.1 代表的な格子基底簡約アルゴリズムの紹介

基底簡約アルゴリズムを紹介するために，Gram-Schmidt の直交化を説明する：基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ の Gram-Schmidt ベクトル $\mathbf{b}_i^*$ は次のように再帰的に定まる：$\mathbf{b}_1^* = \mathbf{b}_1, \mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*$, $\mu_{i,j} = \dfrac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\|\mathbf{b}_j^*\|^2}$。また，各 $2 \le \ell \le m$ に対し $\mathbb{R}^m$ から $\mathbb{R}$-ベクトル空間 $\langle \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{\ell-1} \rangle_{\mathbb{R}}$ の直交補空間への直交射影を π_ℓ とかく。(便宜上，π_1 を恒等写像とする。) 以下で，代表的な 2 つの格子基底簡約アルゴリズムを紹介する。

■ LLL [98]　簡約パラメータ $\dfrac{1}{4} < \delta < 1$ に対し，LLL 基底簡約は次の 2 条件を満たす基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ を見つける （次元 m に関する）多項式時間アルゴリズムである： (i) 基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ はサイズ簡約されている：Gram-Schmidt 係数が $|\mu_{i,j}| \le \dfrac{1}{2}$ $(\forall i > \forall j)$ を満たす。 (ii) 基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ は Lovász 条件を満たす：$\delta\|\mathbf{b}_{k-1}^*\|^2 \le \|\pi_{k-1}(\mathbf{b}_k)\|^2$ $(2 \le \forall k \le m)$ を満たす。入力基底に対して，Lovász 条件が成り立たないとき LLL 基底簡約アルゴリズム内で隣り合う基底ベクトル $\mathbf{b}_{k-1}$ と $\mathbf{b}_k$ の交換を行い，(i) と (ii) の 2 条件を満たす基底を見つける。

■ BKZ [140]　BKZ 基底簡約アルゴリズムは，ブロックサイズ β による LLL 基底簡約アルゴリズムの一般化である。LLL に比べ，BKZ 基底簡約アルゴリズムでより良い簡約基底を見つけることが可能であるが，その計算量は β に関して指数時間である。特に，BKZ 基底簡約アルゴリズムに入力するブロックサイズ β を増やすごとに，実行時間が非常に遅くなるが，より短い基底ベクトルを出力する。具体的には，ブロックサイズ $2 \le \beta \le m$ に対して，BKZ 基底簡約アルゴリズムは次の 2 つの条件を満たす格子 L の基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ を見つける： (i) 基底はサイズ簡約されている。 (ii) すべての $1 \le j \le m$ に対し $\|\mathbf{b}_j^*\| = \lambda_1(L_{[j,k]})$ を満たす。ただし，$k = \min(j + \beta - 1, m)$ とし，射影ベクトル $\pi_j(\mathbf{b}_j), \ldots, \pi_j(\mathbf{b}_k)$ で生成されるブロック射影格子を $L_{[j,k]}$ とする。入力基底に対して，BKZ 基底簡約アルゴリズム内ではブロック射影格子 $L_{[j:k]}$ 上の SVP オラクルを繰り返し呼びだし，(i) と (ii) の 2 条件を満たす基底を見つける。

3.1.3.2 BKZ 基底簡約アルゴリズムの出力基底と計算量

これまで BKZ2.0 [45] などの効率的な BKZ の改良アルゴリズムが提案され，格子に基づく暗号技術の安全性評価で頻繁に利用されている。以下で，BKZ の出力基底と計算量評価の見積もりについて紹介する。 (詳細は [8] を参照。)

■ BKZ の出力基底の見積もり 格子基底簡約アルゴリズムが出力する簡約基底の「良さ」を測る指標として Hermite 因子がある。m 次元格子 L の基底が与えられたとき，アルゴリズムが出力する最短な基底ベクトルを $\mathbf{b} \in L$ とする。このとき，その基底簡約アルゴリズムの Hermite 因子は $\gamma = \dfrac{\|\mathbf{b}\|}{\mathrm{vol}(L)^{1/m}}$ で定義される。(つまり，Hermite 因子が小さいほど，より短い基底ベクトルの出力を意味する。) 100 以上の高次元のランダム格子に対し，LLL や BKZ などの基底簡約アルゴリズムの Hermite 因子の m 乗根 $\gamma^{1/m}$ は定数に収束することが実験的に知られている。高い次元 m のランダム格子において，ブロックサイズ $\beta \geq 50$ に対する BKZ 基底簡約アルゴリズムの root Hermite 因子はおおよそ

$$\gamma^{\frac{1}{m}} \approx \left(\nu_\beta^{-\frac{1}{\beta}}\right)^{\frac{1}{\beta-1}} \approx \left(\frac{\beta}{2\pi e}(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}}$$

に従うことが実験的に知られている [45, 156]。ただし，ν_β は β-次元の単位超球の体積とする。(例えば，$\beta = 85$ で $\gamma^{1/m} \approx 1.01$ となる。) この root Hermite 因子の見積もりを用いて，格子に基づく暗号技術の安全性評価対象の格子問題の求解で必要となる BKZ のブロックサイズ β を求めることができる。

■ BKZ の計算量の見積もり BKZ 基底簡約アルゴリズムの計算量は，β 次元格子上の「SVP オラクルの計算量」と「呼び出し回数」の積で見積もることができる。β 次元格子上の SVP オラクルに適したアルゴリズムとして篩 (sieving) と数え上げ (enumeration) があり，篩の方が漸近計算量が小さい。(ただし，数え上げの空間計算量が β に関して多項式的であるのに対し，篩の空間計算量は β に関して指数関数的である。) 具体的には，β 次元格子上の篩の時間計算量は $2^{c\beta+o(\beta)}$ で，古典計算機上では $c = 0.292$ で，Grover アルゴリズムによって量子計算機上で $c = 0.265$ と見積もられている。一方，数え上げの時間計算量は古典計算機上で $2^{c_1\beta\log\beta+c_2\beta+c_3}$ または $2^{c_1\beta^2+c_2\beta+c_3}$ で，Grover アルゴリズムにより量子計算機上ではその指数部分が半分になると見積もられている。(定数 c_1, c_2, c_3 に関しては様々な評価値があり，具体的な値については [8, Table 4] を参照。) また，BKZ 内の SVP オラクルの呼び出し回数については，β または $8m$ と見積もることが多い。 (β は BKZ のブロックサイズで，m は格子の次元とする。)

■ Core-SVP による安全性レベルの見積もり 上述の LWE や NTRU の探索問題の求解において，秘密情報に対応するベクトル $\mathbf{v}$ を帰着先の格子 L の最短ベクトルとして埋め込み，L の基底に BKZ 基底簡約アルゴリズムを適用することで目的の $\mathbf{v}$ を見つけることを考える。(具体的には，目的の $\mathbf{v}$ は，探索 LWE 問題では q-ary 格子上のノイズベクトル $\mathbf{e}$ で，NTRU 問題では NTRU 格子上のベクトル $(\mathbf{f} \mid \mathbf{g})$ を想定する。) ここで，$\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$ を格子 L の β-BKZ 簡約基底とし，$\{\mathbf{b}_1^*, \ldots, \mathbf{b}_m^*\}$ をその Gram-Schmidt ベクトルとする。Gaussian Heuristic と Geometric Series Assumption (GSA) の仮定の下で，目的ベクトル $\mathbf{v}$ の $m-\beta$ の位置における射影ベクトル $\pi_{m-\beta}(\mathbf{v}) \in \pi_{m-\beta}(L)$ の長さが

$$\|\pi_{m-\beta}(\mathbf{v})\| < \|\mathbf{b}_{m-\beta}^*\| \approx \delta_\beta^{2\beta-m-1}\mathrm{vol}(L)^{\frac{1}{m}}, \quad \delta_\beta = \left(\frac{\beta}{2\pi e}(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}}$$

を満たせば，BKZ 基底簡約の基底ベクトルとして目的の $\mathbf{v}$ を見つけることができる。(探索 LWE 問題に対する BKZ による求解実験については，[11, 127] を参照。) この不等式を満たす BKZ のブロックサイズ β に対して，BKZ 基底簡約アルゴリズムのサブルーチンである β-次元 SVP アルゴリズムの 1 回の計算困難性を Core-SVP 困難性と呼ぶ [13]。格子に基づく暗号方式の具体的な安全性レベルは，Core-SVP 困難性で評価・比較されることが多い。

3.1.3.3 格子問題の公開チャレンジの求解状況

SVP や LWE に対する求解アルゴリズムをテストする目的で，ドイツ・ダルムシュタット大学によって「SVP チャレンジ」・「LWE チャレンジ」と呼ばれる求解コンテストがインターネット上で開催されている [55]。2018 年に，篩（ふるい）をベースとした高速な格子アルゴリズム群である General Sieve Kernel (G6K)[10] が提案され，SVP チャレンジ・LWE チャレンジの求解記録が飛躍的に更新された。具体的には，SVP チャレンジにおいては，G6K 内の篩アルゴリズムを GPU 実装することで，180 次元の SVP インスタンスが 4 NVIDIA Turing GPUs の計算機（1.5TB RAM）を用いて 51.6 日で求解されたと 2021 年 2 月に報告されている [68]。（ただし，本報告では Gaussian Heuristic で期待される最短ベクトル長に対する近似因子が 1.04002 なので，今回見つかった格子ベクトルは 180 次元 SVP インスタンスの厳密解ではなく近似解である。）また 2023 年 7 月に，186 次元の SVP インスタンスに対して，次のスペックを持つ計算機システムで約 50 日程度で近似因子が 1.01405 の非常に短い格子ベクトルを見つけることに成功している（計算時間の内訳は，Progressive pnj-BKZ による基底簡約に 12.3 日，Sieving に短い格子ベクトルの探索に 38 日かかったと報告されている）[56]。

- CPU: 1 ∗ Intel Xeon Gold 6330, 56 threads @ 2.00GHz
- GPU: 4 ∗ NVIDIA A100 80GB PCIe
- Max RAM used: 1441.6685 GB

さらに，2024 年 7 月に，190 次元の SVP インスタンスに対し 1.5TBytes（4 NVIDIA 4090）と 2.0TBytes（3 NVIDIA 4090 D）の RAM を持つ 2 つの計算機で G6K ライブラリの $\beta = 155$〜158 の篩次元を用いて，約 4 か月で近似因子が 1.04237 の短い格子ベクトルを見つけたと報告されている。

LWE チャレンジにおいては，$(n, \alpha) = (40, 0.040), (45, 0.030), (50, 0.025), (55, 0.020), (90, 0.005)$ の数多くの LWE インスタンスが G6K 内の progressive-BKZ の改良により求解されたと 2022 年 6〜10 月に報告されている。（ただし，n は LWE の秘密ベクトル長で，α はノイズの大きさに関するパラメータで，組 (n, α) のバランスで LWE インスタンスの難しさが大きく変化する。）例えば，$(n, \alpha) = (50, 0.025)$ と $(40, 0.040)$ の 2 つの LWE インスタンスに対して，次のスペックを持つ計算システムでそれぞれ約 592 時間と約 683 時間で求解されている [155]：

- HardwareCPU：AMD EPYC 7002 Series 128@2.6GHz
- RAM：1.5TBytes
- GPU：8 ∗ NVIDIA GeForce RTX 3090
- VRAM：8 ∗ 24GB（936.2 GB/s）

2024 年 9 月には，$(n, \alpha) = (95, 0.005)$ の LWE インスタンスに対して，最大 144 の篩次元を用いて約 46 日で求解したと報告されている（計算機は 8∗Nvidia RTX 4090, 2∗ Intel Xeon Platinum 8480+ Processor, 32∗ 64GB DDR5-4800MHz）。

3.2 格子に基づく代表的な暗号方式

本節では，格子に基づく代表的な暗号方式として，LWE 問題に基づく Regev による暗号化方式 [131] および Lindner，Peikert らによる暗号化方式 [100]，Ring-LWE 問題に基づく Brakerski らによる暗号化方式 [38]，NTRU 問題に基づく Hoffstein らによる暗号化方式 [86]，Hash-and-Sign に基づく署名方式の格子問題への拡張，ならびに Fiat-Shamir 署名方式の格子問題への拡張について述べる。

3.2.1 LWE に基づく Regev による暗号化方式

Regev による暗号化方式 [131] の構成には，以下の 4 つのパラメータが必要である。

- n: 安全性パラメータ
- m: LWE サンプルの個数 ($m \geq 1.1 \cdot n\log q$ となる最小の整数を選ぶ)
- q: 剰余パラメータ (q として $n^2 \leq q \leq 2n^2$ を満たす素数を選ぶ)
- $\alpha > 0$: ノイズパラメータ ($\alpha = 1/(\sqrt{n} \cdot \log^2 n)$)

以下に具体的な暗号方式の構成を示す。

秘密鍵の生成 一様ランダムに $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ を選ぶ。

公開鍵の生成 秘密鍵 $\mathbf{s}$，剰余パラメータ q ノイズパラメータ α を持つ LWE 分布から生成した m 個のサンプル $(\mathbf{a}_i, b_i)_{i=1}^m \leftarrow A_{\mathbf{s},\chi}^m$ を公開鍵とする。ただし各 i について，$a \leftarrow \mathbb{Z}_q^n, e_i \leftarrow \chi = D_{\mathbb{Z},\alpha q}$ とした時，$b_i = \langle \mathbf{a}_i, s\rangle + e_i \in \mathbb{Z}_q$ とする。

暗号化 集合 S を $\{1, 2, \ldots, m\}$ の中から一様ランダムに選んだ部分集合とする。このとき，平文ビットが 0 の暗号文を $\left(\sum_{i\in S} \mathbf{a}_i, \sum_{i\in S} b_i\right)$ とし，平文ビットが 1 の暗号文を $\left(\sum_{i\in S} \mathbf{a}_i, \left\lfloor \frac{q}{2} \right\rfloor + \sum_{i\in S} b_i\right)$ とする。

復号 暗号文 $(\mathbf{a}, b)$ に対し，$b - \langle \mathbf{a}, \mathbf{s}\rangle \in \mathbb{Z}_q$ が $\left\lfloor \frac{q}{2} \right\rfloor$ より 0 に近い場合，復号結果として 0 を出力し，それ以外の場合は 1 を出力する。

復号の正当性について，平文 0 を暗号化した暗号文 $(\mathbf{a}, b) = \left(\sum_{i\in S} \mathbf{a}, \sum_{i\in S} b_i\right)$ の場合，$b - \langle \mathbf{a}, \mathbf{s}\rangle \in \mathbb{Z}_q = \sum_{i\in S}(b_i - \langle \mathbf{a}_i, \mathbf{s}\rangle) = \sum_{i\in S} e_i$ なので，$-\frac{q}{4} < \sum_{i\in S} e_i < \frac{q}{4}$ であれば復号に成功し，0 が出力される。

各ノイズ e_i は Gauss 分布 $\chi = D_{\mathbb{Z},\alpha q}$ から選ばれているので，$\sum_{i\in S} e_i$ の標準偏差は高々 $\sqrt{m}\alpha q$ となる。

ここで，各パラメータの選択方法から $\sqrt{m}\alpha q < q/\log n$ であり，非常に高い確率で復号に成功することが分かる。また平文 1 を暗号化した暗号文に対しても同様の議論が成り立つ。この暗号方式の安全性については，LWE 仮定の下で CPA 安全であることが証明されている [132]。

ここで紹介した [131] による暗号方式は，公開鍵のサイズが $O(mn\log q) = \widetilde{O}(n^2)$ で，暗号文サイズも平文サイズの $O(n\log q) = \widetilde{O}(n)$ 倍に増加するため，決して効率的ではない。より効率的な方式としては [122] などを参照。

3.2.2 LWE に基づく Lindner，Peikert らによる暗号化方式

Lindner，Peikert らによる暗号化方式 [100] の構成には，以下のパラメータが必要である。

- n_1, n_2 LWE 問題の安全性パラメータ
- s_k, s_e：鍵生成時，暗号化時のノイズ付与のための Gauss 分布パラメータ
- q：q-ary 格子を構成する剰余パラメータ ($q > 2$)
- l：平文空間の次元，$\{0,1\}^l$ を平文の対象空間とする

以下に具体的な暗号方式の構成を示す。

鍵生成 $\mathbb{Z}_q^{n_1\times n_2}$ からランダムな行列 A を選択する。$n_2 \times l$ 次元 Gauss 分布 $D_{\mathbb{Z},s_k}^{n_2\times l}$ から要素 S を選択し秘密鍵とする。$n_1 \times l$ 次元 Gauss 分布 $D_{\mathbb{Z},s_k}^{n_1\times l}$ から要素 E を選択し，$P = E - AS \in \mathbb{Z}_q^{n_1\times l}$ を求め，(A, P) を公開鍵とする。

暗号化 メッセージ $m \in \{0,1\}^l$ に対し，成分が小さいベクトル $(e_1, e_2, e_3) \in (\mathbb{Z}^{n_1}, \mathbb{Z}^{n_2}, \mathbb{Z}^l)$ の各要素をそれぞれ $D_{\mathbb{Z},s_e}$

から選択する。$c = (e_1A + e_2, e_1P + e_3 + m\left\lfloor \frac{q}{2} \right\rfloor)$ とし c を暗号文とする。

復号 $v = c_1S + c_2$ を求め，各要素毎に m_i を $|v_i| < \frac{q}{4}$ であれば 0，それ以外であれば 1 として $m = \{m_1, ..., m_l\}$ を復号文とする。

復号の正当性について，平文の 0 に対応する暗号文 $c = (e_1A + e_2, e_1P + e_3)$ に対し，$v = c_1S + c_2 = e_1AS + e_2S + e_1P + e_3 = e_2S + e_1E + e_3$ となる。秘密鍵 S ならびに E の各成分は，Gauss 分布 $D_{\mathbb{Z},s_k}$ から，各ノイズ e_i の成分は Gauss 分布 $D_{\mathbb{Z},s_e}$ から選ばれており，また s_k, s_e の値が一定以下に抑えられることから，高い確率で $|v| < s_e s_k(n_1 + n_2) < \frac{q}{4}$ を満たし，復号に成功することが分かる。また平文ビットが 1 の暗号文に対しても同様の議論が成り立つ。この暗号方式の安全性については LWE 仮定の下で CPA 安全であることが証明されている [100]。

3.2.3 Ring-LWE に基づく Brakerski らによる暗号化方式

Brakerski らによる Ring-LWE 問題にもとづく暗号化方式は，暗号化したまま限定回の加算と乗算が可能な somewhat 準同型暗号として提案されているものである。この暗号方式には，以下の 4 つのパラメータが必要である。

- n: 2 のべき乗の整数で，暗号方式を構成する基礎的な環 $R = \mathbb{Z}[x]/(x^n + 1)$ を定義する (n が 2 べきであることから，多項式 $x^n + 1$ が $\mathbb{Z}$ 上既約となることに注意)。
- q: $q \equiv 1 \pmod{2n}$ を満たす素数で，暗号文空間の基礎環 $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ を定義する。
- t: 条件 $t < q$ を満たす整数で，暗号方式の平文空間 $R_t = \mathbb{Z}_t[x]/(x^n + 1)$ を定義する。
- $\sigma > 0$: ノイズを与えるための離散 Gauss 分布のパラメータ。

以下に具体的な暗号方式の構成を示す。なお，$a_0 + a_1x + \cdots + a_{n-1}x^{n-1} \to (a_0, a_1, \ldots, a_{n-1})$ によって，環 R を $\mathbb{Z}^n$ と同一視する。また同様に R_q と $\mathbb{Z}_q^n$ を同一視する。

鍵生成 $s \in R \leftarrow D_{\mathbb{Z}^n,\sigma}$ を選び，一様ランダムに $p_i \in R_q$ を取り，小さなエラー $e \leftarrow \chi$ を固定する。([38] では $s \leftarrow \chi$ を一様ランダムに選択するのに対し，[114] では一様ランダムには選択しない点だけが異なる)。そこで，公開鍵を $\mathsf{pk} = (p_0, p_1)$ とし (ただし，$p_0 = -(p_1s + te)$ とする)，秘密鍵を $\mathsf{sk} = s$ とする。

暗号化 平文情報 $m \in R_t$ と公開鍵 $\mathsf{pk} = (p_0, p_1)$ に対し，まず χ から $u, f, g \in R$ をサンプリングし，暗号文を

$$\mathsf{Enc}(m, \mathsf{pk}) = (c_0, c_1) = (p_0u + tg + m, p_1u + tf),$$

と定義する。ただし，条件 $t < q$ より，この数式では元 $m \in R_t$ を環 R_q の元として見なして計算する。つまり，暗号文は $(R_q)^2$ の元として表現される。

復号 任意の長さの暗号文 $\mathsf{ct} = (c_0, c_1, \ldots, c_\xi)$ に対して，復号は

$$\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}) = [\tilde{m}]_q \bmod t \in R_t,$$

で計算できる。ただし，$\tilde{m} = \sum_{i=0}^{\xi} c_i s^i \in R_q$ であり，$[\tilde{m}]_q$ は元 $\tilde{m}$ の各係数の $[-q/2, q/2)$ への剰余とする。また，$\mathbf{s} = (1, s, s^2, \ldots, s^\xi)$ としたとき，この復号処理を $\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}) = [\langle \mathsf{ct}, \mathbf{s} \rangle]_q \bmod t$ と書き直すこともできる。

この復号アルゴリズムの正当性については，暗号アルゴリズムで得られる暗号文 $\mathsf{ct} = (c_0, c_1)$ に対し，関係式 $p_0 + p_1s = -te$ が成り立つことから，$\langle \mathsf{ct}, \mathbf{s} \rangle = (p_0u + tg + m) + s \cdot (p_1u + tf) = m + t \cdot (g + sf - ue)$ が環 R_q 上で成り立つ。ここで，元 $m + t \cdot (g + sf - ue)$ を環 R の元と見なしたとき，その各係数が $[-q/2, q/2)$ 内に収まっている限り，$[\langle \mathsf{ct}, \mathbf{s} \rangle]_q = m + t \cdot (g + sf - ue)$ が環 R 上で成立する (元 $e, f, g, u \leftarrow \chi$ が十分小さなノイズとして選択されていることに注意)。この場合，剰余 $\bmod t$ の操作で正しい復号結果 $m \in R_t$ が得られる。

また，この暗号方式の安全性については，Ring-LWE 問題の計算量困難性仮定の下で KDM 安全 (key dependent message security) であることが証明されている [38]。

3.2.4 NTRU 問題に基づく Hoffstein らによる暗号化方式

Hoffstein らによる NTRU 問題に基づく暗号化方式 NTRUEncrypt [86] の構成には次のパラメータが必要である。

- n：正の整数 (セキュリティパラメータ)
- q：正の整数 (素数である必要はない)
- p：q と互に素で $p \ll q$ である正の整数
- ϕ: 次数 n の多項式であり環 $R_p = \mathbb{Z}_p[x]/(\phi), R_q = \mathbb{Z}_q[x]/(\phi)$ を定義する (ϕ としては例えば $x^n \pm 1, x^n - x - 1$ 等)

以下に具体的な暗号方式の構成を示す。

鍵生成 すべての係数の絶対値が小さい二つの多項式 $f \in R_q, g \in R_q$ を選ぶ。ただし，f は R_p, R_q の両方において可逆な要素とする。すなわち，ある f_p, f_q が存在し，以下を満たす。

$$f_p \cdot f = 1 \in R_p, f_q \cdot f = 1 \in R_q$$

ここで f, f_p を秘密鍵とし，$h = pf_q \cdot g \in R_q$ を公開鍵とする。なお f_p, f_q ならびに g は f と h を用いて復元可能であることに注意する。

暗号化 平文情報として，すべての係数の絶対値が p より小さい (例えば $-1, 0, 1$ のいずれかである) 要素 $m \in R_q$ とし，公開鍵 $\mathsf{pk} = h$ に対し，$r \in R_q$ を係数が小さい多項式からランダムに選び，暗号文を

$$\mathsf{Enc}(m, \mathsf{pk}) = r \cdot h + m \in R_q$$

と定義する。

復号 暗号文 $c \in R_q$ に対し，復号は

$$\mathsf{Dec}(m, \mathsf{sk}) = [f_p \cdot [f \cdot c]_q]_p$$

で求められる。ただし，$[a]_q, [a]_p$ は元 $a \in R_q$ の各係数をそれぞれ $[-q/2, q/2), [-p/2, p/2)$ に収めたものとする。

復号の正当性については, 次のように示される。$[f \cdot c]_q$ は, $f \cdot c = f \cdot (r \cdot h + m) = f(r \cdot pf_q \cdot g + m) = pr \cdot g + f \cdot m \in R_q$ と変形されるが，r, g, f, m 共に，係数が小さいものから抽出しており，また $p \ll q$ であること，更に係数が $[-q/2, q/2)$ に収められていることから適切なパラメータ選択により，$f \cdot c$ は q による剰余を伴わない等式，すなわち $f \cdot c = pr \cdot g + f \cdot m \in Z[x]/(\phi)$ が満たされる。また右辺第一項は p 倍項であることから，続く p による剰余で消去され，$f_p \cdot (pr \cdot g + f \cdot m) = f_p \cdot f \cdot m = m \in R_p$ となり正しい復号結果 m が得られる。

この NTRU Encrypt 暗号の安全性についてはアルゴリズム提案当初格子問題への安全性帰着がついていなかったが，Stehlé ら [146] により，standard model の CPA 仮定に基づくイデアル格子上の Ring-SIS 問題，ならびに Ring-LWE 問題に帰着されることが示されている。

3.2.5 Hash-and-Sign に基づく署名方式の格子問題への拡張

Hash-and-Sign に基づく署名方式は，Diffie,Hellman らによってその基本形が示されており，落とし戸つき一方向性関数 $f(x)$ ならびに $f^{-1}(x)$ を用いて署名・検証が行われる。

- M：メッセージ
- $h = hash(M)$: 暗号学的ハッシュ関数
- $\sigma = f^{-1}(h)$：署名
- $h = f(\sigma)$ が成り立つかを確認：署名検証

Diffie,Hellman らによる方式では，一方向性関数 $f(x)$ として，素数 p を法とした離散対数問題に基づく関数 $f(x) = a^x \mod p$ が提示されている。

この署名方式は，さまざまな改良が提案されているが，格子問題の困難性に基づく落とし戸つき関数を用いた Hash-and-Sign 署名方式が，Gentry らによって提案されている [82]。以下にその方式を示す。次のパラメータを準備する。

- m, n：正の整数 (セキュリティパラメータ)
- $hash(M)$: 暗号学的ハッシュ関数
- q：素数
- $L = m^{1+\epsilon}, (\epsilon > 0)$：秘密鍵の大きさの上限

以下に具体的な署名方式を示す。

鍵生成 $A \in \mathbb{Z}_q^{n \times m}$ をランダムな行列，$S \in \Lambda_q^{\perp}(\mathbf{A}, \mathbf{q}), ||S|| < L$ を短いベクトルとし，$SA = 0 \mod q$ を満たす行列の組 (A, S) を生成する (具体的な手法は [3] 参照)。秘密鍵を S，公開鍵を A とする。

署名生成 メッセージ M に対しハッシュ関数を作用させた値 $H = hash(M)$ を $D_{\mathbb{Z}^m, s}$ にマッピングし，その値を u とする。$tA = u \mod q$ を満たす t を任意に求める。秘密鍵 S を用いて，$-t$ に近い格子 $\Lambda_q^{\perp}(\mathbf{A}, \mathbf{q})$ 上の点 v を求め，$\sigma = v + t$ とする。σ を署名として出力する。

署名検証 メッセージ m にハッシュ関数を作用させた値 $h = hash(m)$ を $D_{\mathbb{Z}^m, s}$ にマッピングし，値を u とする。σ が短いベクトルでありかつ $(\sigma - u)A = 0$ である場合に正当な署名として受理する。

署名の正当性については，次のように示される。構成の仕方から，$\sigma - u = v$ であり，v は格子 $\Lambda_q^{\perp}(\mathbf{A}, \mathbf{q})$ 上の点であるから，$(\sigma - u)A \mod q = vA \mod q = 0$ が成り立つ。また 秘密鍵 S の特徴から，$\sigma \in D_{\mathbb{Z}^m, s}$ であることから，σ は短いベクトルとなる。本署名方式は LWE 仮定の元で SUF-CMA(Strong Existential Unforgeability under Chosen Message Attack) 安全であることが示されている。

3.2.6 Fiat-Shamir 署名方式の格子問題への拡張

Fiat，Shamir らによって提示された Fiat-Shamir 変換 [77] に基づく署名方式を総称して Fiat-Shamir 署名と呼ばれており，現在までさまざまな方式が提案されている。以下に基本となる方式の一つである素因数分解問題をベースとする方式を記す。合成数 $n = pq$ (p, q は素数) を法とするべき乗剰余演算 $g(x) = g^x \mod n$ を一方向性関数として利用し，秘密鍵 s，公開鍵 $a = g(s)$ を準備する。

- M：メッセージ m
- $h = hash(M)$: 暗号学的ハッシュ関数
- r：ランダムな値
- $(z, y) = (g(r)h + s, g(r))$：署名
- $g(z) = a^r y$ が成り立つかを確認：署名検証

Lyubashevsky によって，Fiat-Shamir with Aborts 型の格子ベースの署名方式が提案されている [77]. 以下にその

具体的な署名方式について述べる。次のパラメータを準備する。

- $hash(M)$: 暗号学的ハッシュ関数
- m：正の整数 (セキュリティパラメータ)
- n：2 のべき乗 (セキュリティパラメータ)
- σ：正の整数 (セキュリティパラメータ)
- κ：${}^{2^\kappa}{}_n C_\kappa > 160$ を満たす整数
- p：$(2\sigma+1)^m 2^{-128/n}$ 程度の素数
- $R = \mathbb{Z}_p[x]/(x^n+1)$：多項式剰余環
- $D = \{z \in R \mid ||g||_\infty \leq mn\sigma\kappa\}$：内積に基づくハッシュ関数向け空間
- $G = \{g \in R \mid ||g||_\infty \leq mn\sigma\kappa - \sigma\kappa\}$：署名空間

ただし，$||z||_\infty$ は z の最大値ノルムとする。以下に具体的な署名方式を示す。

R に属する m 個の多項式の集合 R^m の要素 $\hat{a}$ に対し，D^m 上のハッシュ関数 $h_{\hat{a}}(\hat{z}), (\hat{z} \in D^m)$ を以下のように定める。$h_{\hat{a}}(\hat{z}) = \hat{a} \cdot \hat{z} = a_1 z_1 + \cdots + a_m z_m \in R$。

鍵生成 短い多項式を成分とするランダムなベクトル $\hat{s}$，ならびに D^m のランダムなベクトル $\hat{a}$ によるハッシュ関数 $h_{\hat{a}}()$ を作用させた値 $S = h_{\hat{a}}(\hat{s})$ を求め，$\hat{s}$ を秘密鍵，S を公開鍵とする。

署名生成 メッセージを M とする。

多項式を成分とするベクトル $\hat{y} \in D^m$ をランダムに選択し，$c = hash(h_{\hat{a}}(\hat{y})||M), \hat{z} = \hat{y} + c\hat{s}$ を求める。$\hat{z} \in G^m$ となるまで，ベクトル $\hat{y}$ の選択をくりかえす。$\sigma = (\hat{z}, c)$ を署名として出力する。

署名検証 $\hat{z} \in G^m$ ならびに $c = hash(h_{\hat{a}}(\hat{z}) - Sc, M)$ が成り立つ場合に署名を受理する。

この署名方式の正当性は，$h_{\hat{a}}(\hat{z}) - Sc = h_{\hat{a}}(\hat{y} + c\hat{s}) - h_{\hat{a}}(\hat{s})c = h_{\hat{a}}(\hat{y})$ が成り立つことから保証される。安全性については，環 R 上のイデアルに対する γ-SVP 問題の困難性と等価であることが示されている。

3.3 格子に基づく主要な暗号方式

本節では，格子に基づく主要な暗号方式として，6 つの公開鍵暗号と 3 つの署名を取り上げ，その概要と設計原理を説明する。

格子を用いた主な公開鍵暗号の構成として，最初期の Ajtai-Dwork 型 [4]，GGH 型 [83] から近年の [131] による LWE 型 (Regev 型)，[82, 100] に代表される dual-LWE 型，[86] に代表される NTRU 型が存在する。格子を用いた署名の構成としては主に GGH/NTRUSign 型 [83, 86]，Fiat-Shamir with abort 型 [104, 105]，Hash-and-Sign 型 [82, Sect.6]，Plantard-Susilo-Win 型 [124] 等が知られている[*4]。

また，安全性の根拠となる計算問題に関しても，最短ベクトル問題に直接還元するもの，LWE 問題，SIS 問題，LWR 問題およびそれらの Module 版，Ring 版へと還元するもの，NTRU 問題に還元するものへと分類できる。

これらの構成は安全性，実装時の性能，使いやすさなどの面から様々な長所・短所を持つが，できる限り幅広くそれらを解説するため，以下の表 3.1 に挙げる 9 つの方式を紹介する。

- FrodoKEM は dual-LWE 型の公開鍵暗号であり，安全性の根拠に LWE 問題の計算困難性を仮定している。保守的な構成を設計指針としており，将来 Ring 型や Module 型の構造を持つ格子問題に効率的な解法が発見された場合でも安全性が確保されると期待される。保守的な構成という観点から取り上げる。
- NewHope は dual-LWE 型の公開鍵暗号であり，安全性の根拠に $x^n + 1, n = 2^k$ の形の多項式により定義され

[*4] この分類に関しては例えば [89, Sect. 3]，[74, Sect. 5.5] 等を参照。

表 3.1: 格子に基づく暗号の分類

文献	暗号化	鍵交換	署名
FrodoKEM [14]	○	○	
NewHope [15]	○	○	
NTRU [42]	○	○	
SABER [27]	○	○	
CRYSTALS-Kyber [20]	○	○	
ML-KEM (FIPS 203) [119]	○	○	
CRYSTALS-Dilithium [21]			○
ML-DSA (FIPS 204) [118]			○
FALCON [80]			○

る環上の Ring-LWE 問題の困難性を置いている。環の構造を活用した数論変換による高速実装，サイズの低減という観点から取り上げる。

- NTRU は NTRU 型の公開鍵暗号であり，NTRU 格子上の格子問題の計算困難性を安全性の根拠としている。NTRU 暗号は 1996 年の提案依頼改良が続けられてきたが，本方式は近年の研究成果が数多く盛り込まれた形となるため取り上げる。
- SABER は LWE 型の公開鍵暗号であり[*5]，$x^{256}+1$ を定義多項式とする環上の Module-LWR 問題の計算困難性を安全性の根拠としている。LWR 問題を用いることで実装時のサンプリングを極力負担の少ないものとしている。この観点から取り上げる。
- CRYSTALS-Kyber は dual-LWE 型の公開鍵暗号であり，安全性の根拠に $x^n+1, n=2^k$ の形の多項式により定義される環上の Module-LWE 問題の困難性を置いている。NIST PQC 標準化プロジェクトの Selected Algorithm となったことから取り上げる。
- ML-KEM は CRYSTALS-Kyber に基づく dual-LWE 型の公開鍵暗号である。NIST により FIPS 標準アルゴリズムとして制定されたことから，取り上げる。
- CRYSTALS-Dilithium は Fiat-Shamir 型の署名方式であり，$x^{256}+1$ を定義多項式とする環上の Module-LWE 問題の計算困難性を安全性の根拠としている。 環の性質を用いた数論変換による高速処理とサイズの圧縮が可能であり，公開鍵サイズと署名サイズの和を最小化することを目的としてパラメータ設計を行っている。環の性質を用いた処理の効率化の観点から取り上げる。ML-DSA は CRYSTALS-Dilithium に基づく署名方式である。NIST により FIPS 標準アルゴリズムとして制定されたことから，取り上げる。
- FALCON は Hash-and-Sign 型の署名方式であり，x^n+1 を定義多項式とする NTRU 格子上の SIS 問題の困難性を安全性の根拠としている。格子上の高速フーリエサンプリングを用いた高速な署名生成を特徴とし，方式提案後も数多くの改良が提案されていることから取り上げる。

[*5] 仕様書 [27] の設計原理の項には “Encryption: we use a simple LWR version of Regev’s LWE encryption scheme [35], where the encryption part is compressed (using the parameter T) to save on bandwith.” とあるが, Second PQC Standardization Conference の発表スライド [52, p.5] においてはひな型を dual-LWE 型暗号としている。数式の比較から，どちらも原型と考えることが可能であるため，本報告書では仕様書に従い LWE 型であるとした。

3.3.1 FIPS 203：Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)

KEM とは公開されたチャンネル上で 2 者が秘密を共有するアルゴリズム群である。KEM で安全に生成された共有の秘密は共通鍵暗号で用いられ，暗号や認証などの安全なやり取りの中で重要な役割を果たす。ML-KEM [119] は CRYSTALS-Kyber に基づく KEM で，その安全性は加群上の LWE 問題の計算量困難性に基づく。具体的には，2 のべき数 $n = 256$ に対し $R := \mathbb{Z}[X]/(X^n + 1)$ を基本環とし，素数 $q = 3329$ に対し $R_q := R/qR = \mathbb{Z}_q[X]/(X^n + 1)$ をその剰余環とする。環 R_q の元は $\mathbb{Z}_q$ を係数とする $n-1$ 以下の次数の多項式 $f = f_0 + f_1X + \cdots + f_{n-1}X^{n-1}$ と表せ，その係数ベクトル $(f_0, f_1, \ldots, f_{n-1})$ を対応させることで，$\mathbb{Z}_q$ 加群として R_q は $\mathbb{Z}_q^n$ と同型である。ML-KEM は，階数パラメータ $k \in \{2, 3, 4\}$ に対し，$\mathbb{Z}_q$ 加群 $R_q^k \simeq (\mathbb{Z}_q^n)^k$ 上の LWE 問題を安全性の根拠とした KEM である。特に，R_q における乗算を高速化するために，Number-Theoretic Transform（NTT）とよばれる変換を利用する。ここでは，ML-KEM の最も基本となる構成要素である NTT を説明したのちに，ML-KEM の基本構成について説明する。

3.3.1.1 数論変換：Number-Theoretic Transform (NTT)

NTT は，環 R_q の元 f を R_q と同型な環 T_q の元 $\widehat{f}$ に写し，T_q における乗算を利用して効率的に R_q の 2 つの元の乗算を行う手法である。これは $\mathbb{C}$ 上の高速フーリエ変換による多項式乗算と同じアイデアで，NTT はその $\mathbb{Z}_q$ 上版とみなせる。ML-KEM では，2 のべき数 $n = 2^8 = 256$ と素数 $q = 3329$ で定まる剰余環 $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ を用いる（ML-KEM の暗号パラメータについては，後述の 3.3.1.3 節を参照）。これらの暗号パラメータ (n, q) において，$\mathbb{Z}_q^* := \mathbb{Z}_q \setminus \{0\}$ は位数 $q - 1 = 3328 = 2^8 \cdot 13$ の巡回群で，$\mathbb{Z}_q^*$ は位数 $2^8 = 256 = n$ の巡回部分群 $\langle\zeta\rangle$ を唯一つ含む。具体的には，$\mathbb{Z}_q$ において $\zeta := 17 \bmod q$ が 1 の原始 n 乗根で，$\{\zeta, \zeta^3, \ldots, \zeta^{n-1}\}$ が $\mathbb{Z}_q$ に含まれる 1 の原始 n 乗根のすべてである。ここで，$N = \frac{n}{2} = 128$ とおくと，各 $i = 0, 1, \ldots, N-1$ に対して，$\zeta^{(2i+1)N} \equiv -1 \pmod q$ である。ゆえに，多項式環 $\mathbb{Z}_q[X]$ において，$X^n + 1$ は次のように N 個の 2 次式の積に分解できる。

$$X^n + 1 = \prod_{i=0}^{N-1} \left(X^2 - \zeta^{2i+1}\right) = \prod_{i=0}^{N-1} \left(X^2 - \zeta^{2\mathsf{BitRev}_7(i)+1}\right) \in \mathbb{Z}_q[X].$$

ただし，$\mathsf{BitRev}_7(i)$ は符号なし 7 ビット整数 i のビット逆順整数を表し，実装上の都合のため ML-KEM ではこの順序を利用する。以下では，数論変換の原理を説明するために，$i = 0, 1, \cdots, N-1$ の単純な順序を用いる。上記の $X^n + 1$ の分解により，次の（$\mathbb{Z}_q$ 加群としての）同型を得る。

$$R_q = \mathbb{Z}_q[X]/(X^n + 1) \simeq \bigoplus_{i=0}^{N-1} \mathbb{Z}_q[X]/\left(X^2 - \zeta^{2i+1}\right) =: T_q.$$

具体的には，この同型は

$$\mathsf{NTT} : R_q \longrightarrow T_q, \quad f \longmapsto \widehat{f} := \left(f \bmod (X^2 - \zeta^{2i+1})\right)_{i=0}^{N-1} \tag{3.1}$$

で定まる。特に，T_q を **NTT 空間**，$\widehat{f} = \mathsf{NTT}(f) \in T_q$ を $f \in R_q$ の **NTT 表現**とよぶ。

■NTT 表現について $f = f_0 + f_1X + \cdots + X^{n-1} \in R_q$ の偶数と奇数の次数に関する多項式をそれぞれ

$$f_e := f_0 + f_2Y + f_4Y^2 + \cdots + f_{2N-2}Y^{N-1}, \quad f_o := f_1 + f_3Y + f_5Y^2 + \cdots + f_{2N-1}Y^{N-1} \in \mathbb{Z}_q[Y]$$

とおく。構成から $f = f_e(X^2) + f_o(X^2)X$ なので，各 $i = 0, 1, \ldots, N-1$ に対して，

$$\widehat{f}_{2i} := f_e(\zeta^{2i+1}) = \sum_{j=0}^{N-1} f_{2j}\zeta^{(2i+1)j}, \quad \widehat{f}_{2i+1} := f_o(\zeta^{2i+1}) = \sum_{j=0}^{N-1} f_{2j+1}\zeta^{(2i+1)j}$$

とおくと，

$$f \equiv \widehat{f}_{2i} + \widehat{f}_{2i+1}X \pmod{\left(X^2 - \zeta^{2i+1}\right)} \tag{3.2}$$

が成り立つ。これより，f の NTT 表現について，$\widehat{f} = \left(\widehat{f}_{2i} + \widehat{f}_{2i+1}X\right)_{i=0}^{N-1} \in T_q$ とかける。

■NTT 表現の行列表示　$\mathbb{Z}_q$ の元を成分とする $N \times N$ 行列を

$$\mathbf{B} = A(\zeta) := \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \zeta^3 & \zeta^6 & \cdots & \zeta^{3(N-1)} \\ 1 & \zeta^5 & \zeta^{10} & \cdots & \zeta^{5(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{2N-1} & \zeta^{2(2N-1)} & \cdots & \zeta^{(N-1)(2N-1)} \end{pmatrix} \in (\mathbb{Z}_q)^{N \times N}$$

とおく。R_q の元 $f = f_0 + f_1X + \cdots + X^{n-1}$ の偶数と奇数の次数に関するそれぞれの係数ベクトル $(f_0, f_2, \ldots, f_{2N-2}), (f_1, f_3, \ldots, f_{2N-1}) \in \mathbb{Z}_q^N$ に対して

$$\begin{pmatrix} \widehat{f}_0 \\ \widehat{f}_2 \\ \widehat{f}_4 \\ \vdots \\ \widehat{f}_{2N-2} \end{pmatrix} = \begin{pmatrix} f_e(1) \\ f_e(\zeta^3) \\ f_e(\zeta^5) \\ \vdots \\ f_e(\zeta^{2N-1}) \end{pmatrix} = \mathbf{B} \begin{pmatrix} f_0 \\ f_2 \\ f_4 \\ \vdots \\ f_{2N-2} \end{pmatrix}, \quad \begin{pmatrix} \widehat{f}_1 \\ \widehat{f}_3 \\ \widehat{f}_5 \\ \vdots \\ \widehat{f}_{2N-1} \end{pmatrix} = \begin{pmatrix} f_o(1) \\ f_o(\zeta^3) \\ f_o(\zeta^5) \\ \vdots \\ f_o(\zeta^{2N-1}) \end{pmatrix} = \mathbf{B} \begin{pmatrix} f_1 \\ f_3 \\ f_5 \\ \vdots \\ f_{2N-1} \end{pmatrix}$$

が成り立つ。つまり，$f \in R_q$ の偶数と奇数の次数の係数ベクトルはそれぞれ $\mathbf{B}$ による線形変換（つまり，離散フーリエ変換）で $\widehat{f} \in T_q$ の偶数と奇数の添え字番号のベクトルに写る。$\mathbf{B}$ の逆行列は $\mathbf{C} := \frac{1}{N}A(\zeta^{-1})$ なので，式 (3.1) の NTT 写像の逆写像 NTT^{-1} は行列 $\mathbf{C}$ を用いて計算可能である（つまり，逆離散フーリエ変換から計算可能）。

■NTT 空間における乗算　R_q の 2 つの元 f, g に対して，その積を $h := f \cdot g \in R_q$ とおく。h の NTT 表現 $\widehat{h} \in T_q$ について，式 (3.2) から，各 $i = 0, 1, \ldots, N-1$ に対して

$$\widehat{h}_{2i} + \widehat{h}_{2i+1}X \equiv h = f \cdot g \equiv \left(\widehat{f}_{2i} + \widehat{f}_{2i+1}X\right)\left(\widehat{g}_{2i} + \widehat{g}_{2i+1}X\right) \bmod \left(X^2 - \zeta^{2i+1}\right)$$

が成り立つ。ここで，2 つの NTT 表現 $\widehat{f} = \left(\widehat{f}_{2i} + \widehat{f}_{2i+1}X\right)_{i=0}^{N-1}, \widehat{g} = \left(\widehat{g}_{2i} + \widehat{g}_{2i+1}X\right)_{i=0}^{N-1} \in T_q$ の積を

$$\begin{aligned} \widehat{f} \circ \widehat{g} &:= \left(\left(\widehat{f}_{2i} + \widehat{f}_{2i+1}X\right) \cdot \left(\widehat{g}_{2i} + \widehat{g}_{2i+1}X\right) \bmod \left(X^2 - \zeta^{2i+1}\right)\right)_{i=0}^{N-1} \\ &= \left(\widehat{f}_{2i}\widehat{g}_{2i} + \widehat{f}_{2i+1}\widehat{g}_{2i+1}\zeta^{2i+1} + \left(\widehat{f}_{2i}\widehat{g}_{2i+1} + \widehat{f}_{2i+1}\widehat{g}_{2i}\right)X\right)_{i=0}^{N-1} \in T_q \end{aligned}$$

と定めると，

$$\mathsf{NTT}(f \cdot g) = \mathsf{NTT}(f) \circ \mathsf{NTT}(g) \iff f \cdot g = \mathsf{NTT}^{-1}\left(\widehat{f} \circ \widehat{g}\right) \in R_q$$

が成り立つ（つまり，式 (3.1) の NTT 写像は環の同型写像である）。特に，NTT 空間 T_q における乗算は，成分ごとの演算であるため，（R_q における乗算に比べて）効率的に計算可能である。

3.3.1.2　ML-KEM の基本構成と処理概要

加群 $R_q^k \simeq (\mathbb{Z}_q^n)^k$ 上の LWE 問題に基づく ML-KEM は 2 つのステップで構成される。1 つ目は R_q^k 上の LWE 問題から公開鍵暗号（K-PKE）を構成し，2 つ目は藤崎-岡本変換により KEM に変換する。藤崎-岡本変換の性質により，公開鍵暗号方式から構成される KEM はより一般的な攻撃モデルにおいて安全であり，IND-CCA2 安全性を満たす。

■K-PKE の処理概要　ここでは，K-PKE の処理概要とその原理が分かるように，簡略化した形で各アルゴリズムの処理を説明する。特に，処理の高速化のために，NTT 変換を適宜利用する。

K-PKE 鍵生成　鍵生成アルゴリズム（FIPS 203 の Algorithm 13, K-PKE.KeyGen(d)）では，乱数 d を入力として，暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ と復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ を次のように出力する。

- 入力：乱数 d
- 出力：暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ と復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$
  1. $(\rho, \sigma) \leftarrow \mathsf{G}(d\|k)$：ハッシュ関数 G を用いて擬似ランダムな乱数の組 (ρ, σ) を生成
  2. $\widehat{\mathbf{A}} = \left(\widehat{\mathbf{A}}[i,j]\right)_{0\le i,j<k} \in (T_q)^{k\times k}$: 乱数 ρ を用いて，NTT 表現の公開鍵行列を生成
  3. $\mathbf{s} = (\mathbf{s}[i])_{0\le i<k} \in R_q^k$：各 $\mathbf{s}[i] \in R_q$ のすべて $\mathbb{Z}_q$ 係数は中心二項分布 CBD からサンプルする（十分小さい）
  4. $\mathbf{e} = (\mathbf{e}[i])_{0\le i<k} \in R_q^k$：各 $\mathbf{e}[i] \in R_q$ のすべての $\mathbb{Z}_q$ 係数は CBD からサンプルする（十分小さい）
  5. $\widehat{\mathbf{s}} = (\mathsf{NTT}(\mathbf{s}[i]))_{0\le i<k} \in T_q^k$：各 $\mathbf{s}[i]$ を NTT 変換
  6. $\widehat{\mathbf{e}} = (\mathsf{NTT}(\mathbf{e}[i]))_{0\le i<k} \in T_q^k$：前のステップ同様，各 $\mathbf{e}[i]$ を NTT 変換
  7. $\widehat{\mathbf{t}} = \widehat{\mathbf{A}} \circ \widehat{\mathbf{s}} + \widehat{\mathbf{e}} = \left(\sum_{j=0}^{k-1} \widehat{\mathbf{A}}[i,j] \circ \widehat{\mathbf{s}}[j] + \widehat{\mathbf{e}}[i]\right)_{0\le i<k} \in T_q^k$：NTT 空間上で LWE 関係式を生成
  8. $\mathtt{ek}_{\mathtt{PKE}} = \left(\widehat{\mathbf{t}}, \rho\right), \mathtt{dk}_{\mathtt{PKE}} = \widehat{\mathbf{s}}$（公開鍵行列 $\widehat{\mathbf{A}}$ は ρ から復元可能であることに注意）
  9. $(\mathtt{ek}_{\mathtt{PKE}}, \mathtt{dk}_{\mathtt{PKE}})$ を出力

ステップ 2 において，NTT 表現の公開鍵行列の各成分 $\widehat{\mathbf{A}}[i,j]$ は，入力する乱数から擬似ランダムな T_q の元を出力する SampleNTT 関数（FIPS 203 の Algorithm 7）を用いて生成する（具体的には，$\widehat{\mathbf{A}}[i,j] \leftarrow \mathsf{SampleNTT}(\rho\|i\|j)$ と生成）。ステップ 3, 4 において，各 $\mathbf{s}[i]$ または $\mathbf{e}[i]$ の多項式のすべての $\mathbb{Z}_q$ 係数は，SamplePolyCBD 関数（FIPS 203 の Algorithm 8）を用いて生成する。具体的には，$\eta \in \{2,3\}$ に対する $\mathbb{Z}_q$ 上の二項分布 CBD_η を

(i) $(x_1, \ldots, x_\eta, y_1, \ldots, y_\eta) \in \{0,1\}^{2\eta}$ を一様ランダムにサンプルする
(ii) $\sum_{i=1}^{\eta}(x_i - y_i) \bmod q \in \mathbb{Z}_q$ を出力

と定め，$\mathbf{s}[i]$ と $\mathbf{e}[i]$ の各 $\mathbb{Z}_q$ 係数は CBD_η からサンプルする（CBD_η の引数の一つとして，ステップ 1 で生成した σ を用いる）。ステップ 8 において，FIPS 203 では $\left(\widehat{\mathbf{t}}, \rho\right)$ と $\widehat{\mathbf{s}}$ をそれぞれ符号化関数 ByteEncode（FIPS 203 の Algorithm 5）で符号化したものを暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ と復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ とする。

鍵生成アルゴリズムにおいて，ρ から NTT 表現の公開鍵行列 $\widehat{\mathbf{A}}$ が復元可能なので，暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ は NTT 表現の LWE インスタンスの組 $\left(\widehat{\mathbf{A}}, \widehat{\mathbf{t}}\right)$ に対応する。特に，$\mathbf{t} := \mathsf{NTT}^{-1}\left(\widehat{\mathbf{t}}\right) \in R_q^k, \mathbf{A} := \mathsf{NTT}^{-1}\left(\widehat{\mathbf{A}}\right) \in (R_q)^{k\times k}$ とおくと，R_q^k 上の LWE 関係式 $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$ が成り立つ。一方，復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ は NTT 表現の LWE の秘密 $\widehat{\mathbf{s}}$ であるので，暗号鍵から復号鍵を見つけるのは $T_q^k \simeq R_q^k$ 上の探索 LWE 問題である。特に，適切な暗号パラメータ（後述の 3.3.1.3 節を参照）を利用した場合，その LWE 問題を解くのは計算量的に非常に困難である。また，鍵生成アルゴリズムにおいて，NTT 空間上で公開鍵行列 $\widehat{\mathbf{A}}$ を直接生成すると共に，ステップ 7 で NTT 空間上で LWE 関係式を生成することで，計算の高速化を図る。

K-PKE 暗号化　暗号化アルゴリズム（FIPS 203 の Algorithm 14, K-PKE.Encrypt）では，暗号化鍵 $\mathtt{ek}_{\mathtt{PKE}}$ と平文 m を入力として，次のように暗号文 c を出力する。

- 入力：暗号化鍵 $\mathtt{ek}_{\mathtt{PKE}} = \left(\widehat{\mathbf{t}}, \rho\right)$，平文 m と乱数 r
- 出力：暗号文 c
  1. ρ から NTT 表現の公開鍵行列 $\widehat{\mathbf{A}} \in (T_q)^{k\times k}$ を復元
  2. $\mathbf{y} = (\mathbf{y}[i])_{0\le i<k} \in R_q^k$：各 $\mathbf{y}[i] \in R_q$ のすべての $\mathbb{Z}_q$ 係数は二項分布 CBD からサンプルする（十分小さい）
  3. $\mathbf{e}_1 = (\mathbf{e}_1[i])_{0\le i<k} \in R_q^k$：各 $\mathbf{e}[i] \in R_q$ のすべての $\mathbb{Z}_q$ 係数は CBD からサンプルする（十分小さい）
  4. $e_2 \in R_q$：すべての $\mathbb{Z}_q$ 係数は CBD からサンプルする（十分小さい）
  5. $\widehat{\mathbf{y}} = (\mathsf{NTT}\,(\mathbf{y}[i]))_{0\le i<k} \in T_q^k$

6. $\mathbf{u} = \mathsf{NTT}^{-1}\left(\widehat{\mathbf{A}}^\top \circ \widehat{\mathbf{y}}\right) + \mathbf{e}_1 = \mathbf{A}^\top\mathbf{y} + \mathbf{e}_1 = \left(\sum_{j=0}^{k-1} \mathbf{A}[j,i]\mathbf{y}[j] + \mathbf{e}_1[i]\right)_{0\le i<k} \in R_q^k$
   （ただし，$\mathbf{A} = \mathsf{NTT}^{-1}(\widehat{\mathbf{A}}) = (\mathbf{A}[i,j])_{0\le i,j<k} \in (R_q)^{k\times k}$ とする）
7. $\mu = \mathsf{Decompress}\,(\mathsf{ByteDecode}(m)) \in R_q$：平文 m をビット列化した後に R_q の元に変換
8. $v = \mathsf{NTT}^{-1}\left(\widehat{\mathbf{t}}^\top \circ \widehat{\mathbf{y}}\right) + e_2 + \mu = \mathbf{t}^\top\mathbf{y} + e_2 + \mu \in R_q$
9. $c = (\mathbf{u}, v) \in R_q^k \times R_q$ を出力

ステップ 2, 3, 4 において，r をシードとした擬似乱数を引数とした SamplePolyCBD 関数で，すべての $\mathbb{Z}_q$ 係数が十分小さい多項式を生成する。ステップ 7 では，バイト列で表現された平文 m を ByteDecode 関数（FIPS 203, Algorithm 6）でビット列 $(m_0, m_1, \ldots, m_{n-1})$ に変換した後に，各ビット $m_i \in \{0,1\}$ を Decompress 関数で $\mu_i := \left\lceil \frac{q}{2} \cdot m_i \right\rfloor \in \mathbb{Z}_q$ に変換する。また，各 μ_i を係数とする多項式を $\mu = \mu_0 + \mu_1 x + \cdots + \mu_{n-1}x^{n-1} \in R_q$ とする。ステップ 9 において，FIPS 203 では $\mathbf{u}$ と v はそれぞれ Compress 関数で圧縮した後，ByteEncode 関数で符号化する。

上記の暗号化アルゴリズムにおいて，暗号文は $c = (\mathbf{u}, v) = \left(\mathbf{A}^\top\mathbf{y} + \mathbf{e}_1, \mathbf{t}^\top\mathbf{y} + e_2 + \mu\right) \in R_q^k \times R_q$ の形で，LWE に基づく Lindner-Peikert による暗号方式と同様，R_q^k 上の LWE 問題が計算困難であれば，暗号文から μ（つまり，平文 m）の情報が洩れない。また，ステップ 6 と 8 において，NTT 空間上で $\mathbf{A}^\top\mathbf{y}$ と $\mathbf{t}^\top\mathbf{y}$ を計算することで，計算の高速化を図る。

K-PKE 復号　復号アルゴリズム（FIPS 203 の Algorithm 15, K-PKE.Decrypt）では，復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ と暗号文 c を入力とし，次のように復号文 m' を出力する。

- 入力：復号鍵 $\mathtt{dk}_{\mathtt{PKE}} = \widehat{\mathbf{s}}$ と暗号文 $c = (\mathbf{u}, v)$
- 出力：復号文 m'
  1. $w = v - \mathsf{NTT}^{-1}\left(\widehat{\mathbf{s}}^\top \circ \mathsf{NTT}(\mathbf{u})\right) = v - \mathbf{s}^\top\mathbf{u} \in R_q$
  2. $m' = \mathsf{ByteEncode}\,(\mathsf{Compress}(w))$ を出力

ステップ 2 において，多項式表現の R_q の元 $w = w_0 + w_1 x + \cdots + w_{n-1}x^{n-1}$ に対して，各係数 $w_i \in \mathbb{Z}_q$ を Compress 関数で $z_i := \left\lceil \frac{2}{q} \cdot w_i \right\rfloor \bmod 2 \in \{0,1\}$ に変換する。また，ビット列 $(z_0, z_1, \ldots, z_{n-1})$ を ByteEncode 関数（FIPS 203, Algorithm 5）でバイト列に変換する。特に，ByteEncode 関数と ByteDecode 関数は互いの逆関数である。

上記の復号アルゴリズムにおいて，暗号文 $c = (\mathbf{u}, v) = \left(\mathbf{A}^\top\mathbf{y} + \mathbf{e}_1, \mathbf{t}^\top\mathbf{y} + e_2 + \mu\right) \in R_q^k \times R_q$ に対して，R_q^k 上の LWE 関係式 $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$ から，

$$\begin{aligned} w = v - \mathbf{s}^\top\mathbf{u} &= \mathbf{t}^\top\mathbf{y} + e_2 + \mu - (\mathbf{A}\mathbf{s})^\top\mathbf{y} - \mathbf{s}^\top\mathbf{e}_1 \\ &= \mathbf{t}^\top\mathbf{y} + e_2 + \mu - \left(\mathbf{t}^\top + \mathbf{e}^\top\right)\mathbf{y} - \mathbf{s}^\top\mathbf{e}_1 = \mu + \underbrace{e_2 - \mathbf{e}^\top\mathbf{y} - \mathbf{s}^\top\mathbf{e}_1}_{\text{すべての } \mathbb{Z}_q \text{ 係数が十分小さい}} \in R_q \end{aligned}$$

が成り立つ。ここで，$\mathbf{s}, \mathbf{e}, \mathbf{e}_1, \mathbf{y} \in R_q^k$ の各成分 $\mathbf{s}[i], \mathbf{e}[i], \mathbf{e}_1[i], \mathbf{y}[i] \in R_q$ と $\mu \in R_q$ のすべての $\mathbb{Z}_q$ 係数は十分小さいことに注意する。よって，Compress 関数による各 $\mathbb{Z}_q$ 係数におけるノイズ補正により

$$\mathsf{Compress}(w) = \mathsf{Compress}(\mu) = (m_0, m_1, \ldots, m_{n-1}) \in \{0,1\}^{n-1}$$

が成り立つ。最後に，ByteEncode 関数により，平文のビット列 $(m_0, m_1, \ldots, m_{n-1})$ をバイト列に変換することで，元の平文 m に復号できる（つまり，復号文 m' は平文 m に一致する）。また，ステップ 1 において，NTT 空間上で $\mathbf{s}^\top\mathbf{u}$ を計算することで，計算の高速化を図る。

■ML-KEM の処理概要　上記で構成した K-PKE 方式を用いて，ML-KEM を下記のように構成する。

ML-KEM 鍵生成　鍵生成アルゴリズム（FIPS 203, Algorithm 16）では，K-PKE 鍵生成アルゴリズムを用いて，2 つの乱数 d, z から鍵カプセル化鍵 ek とデカプセル化鍵 dk を次のように出力する。

- 入力：2 つの乱数 d, z

- 出力：鍵カプセル化鍵 $\mathtt{ek}$ とデカプセル化鍵 $\mathtt{dk}$
  1. K-PKE 鍵生成アルゴリズムで，乱数 d から $(\mathtt{ek}_{\mathrm{PKE}}, \mathtt{dk}_{\mathrm{PKE}})$ を生成
  2. $\mathtt{ek} = \mathtt{ek}_{\mathrm{PKE}}$
  3. $\mathtt{dk} = (\mathtt{dk}_{\mathrm{PKE}}, \mathtt{ek}, \mathsf{H}(\mathtt{ek}), z)$：$\mathsf{H}$ はハッシュ関数
  4. $(\mathtt{ek}, \mathtt{dk})$ を出力

ML-KEM 鍵カプセル化　鍵カプセル化アルゴリズム（FIPS 203, Algorithm 17）では，K-PKE 暗号化アルゴリズムを用いて，鍵カプセル化鍵 $\mathtt{ek}$ と乱数 m から共有の秘密鍵 K と暗号文 c を次のように出力する。

- 入力：鍵カプセル化鍵 $\mathtt{ek}$ と乱数 m
- 出力：共有の秘密鍵 K と暗号文 c
  1. $(K, r) = \mathsf{G}\,(m \| \mathsf{H}(\mathtt{ek}))$：$\mathsf{G}$ はハッシュ関数
  2. K-PKE 暗号化アルゴリズムで，$(\mathtt{ek}, m, r)$ から暗号文 c を生成
  3. (K, c) を出力

ML-KEM デカプセル化　デカプセル化アルゴリズム (FIPS 203, Algorithm 18) では，K-PKE 復号アルゴリズムを用いて，デカプセル化 $\mathtt{dk}$ と暗号文 c から，共有の秘密鍵 K を次のように出力する。また，c が改竄されていないことを保証するために，K-PKE 暗号化アルゴリズムで復号文から暗号文 c' を生成し，c と c' が一致するか検証する。

- 入力：デカプセル化 $\mathtt{dk} = (\mathtt{dk}_{\mathrm{PKE}}, \mathtt{ek}, \mathsf{H}(\mathtt{ek}), z)$ と暗号文 c
- 出力：共有の秘密鍵 K
  1. K-PKE 復号アルゴリズムで，復号鍵 $\mathtt{dk}_{\mathrm{PKE}}$ と暗号文 c から，復号文 m' を生成
  2. $(K', r') = \mathsf{G}\,(m' \| \mathsf{H}(\mathtt{ek}))$
  3. $\bar{K} = \mathsf{J}(z \| c)$：$\mathsf{J}$ はハッシュ関数
  4. K-PKE 暗号化アルゴリズムで，$(\mathtt{ek}, m', r')$ から暗号文 c' を生成
  5. $c \neq c'$ の場合は，$K' = \bar{K}$ とおく
  6. K' を出力

3.3.1.3　暗号パラメータ

ML-KEM における主な暗号パラメータと対応する鍵や暗号文のサイズと安全性レベルは以下である。具体的には，LWE の次元 $n = 256$ と剰余素数 $q = 3329$ は ML-KEM-512, -768, -1024 の 3 種類の暗号パラメータで共通であるが，主に 3 種類の階数パラメータ $k \in \{2, 3, 4\}$ により安全性レベルが異なる。(ML-KEM のパラメータ名は，$n \times k \in \{512, 768, 1024\}$ の値により名づけられている。)

	暗号パラメータ			サイズ（単位：バイト）				安全性
	n	q	k	カプセル化鍵	デカプセル化鍵	暗号文	共有の秘密鍵	レベル
ML-KEM-512	256	3329	2	800	1632	768	32	レベル 1
ML-KEM-768	256	3329	3	1184	2400	1088	32	レベル 3
ML-KEM-1024	256	3329	4	1568	3168	1568	32	レベル 5

3.3.1.4 CRYSTALS-Kyber との違い

- Kyber の round 3 version では，共有する秘密鍵は長さが可変な値として扱われていた。一方，ML-KEM では，その長さは 256 ビットに固定している。また，その鍵は直接共通鍵として利用できる。
- ML-KEM.Encaps と ML-KEM.Decaps のアルゴリズムでは，第 3 ランド仕様とは異なる藤崎-岡本変換を利用する。具体的には，ML-KEM.Encaps は共有する秘密の導出において暗号文のハッシュ値を含まず，ML-KEM.Decaps ではその変更に合わせている。
- 第 3 ラウンドの仕様では，ML-KEM.Ecaps アルゴリズム内の初期乱数 m は使う前にハッシュ化される。具体的には，アルゴリズム 16 の 1 と 2 行目の間に，$m \leftarrow H(m)$ のステップがあったが，ML-KEM ではその処理は不必要で行わない。
- ML-KEM では, 第 3 ラウンドの仕様にはなかった入力データの検証ステップを含む。例えば, ML-KEM.Encaps では，カプセル化キーを含むバイト配列が，モジュラー還元なしで q を法とする整数配列に正しくデコードされることを必要とする。

3.3.2 FIPS 204: Module-Lattice-Based Digital Signature Standard（ML-DSA）

ML-DSA [118] は CRYSTALS-Dilithium に基づく署名方式である。ML-KEM と同じように，2 のべき数 $n = 256$ に対し $R := \mathbb{Z}[X]/(X^n + 1)$ を基本環とし，素数 $q = 8380417$ に対し $R_q := R/qR = \mathbb{Z}_q[X]/(X^n + 1)$ をその剰余環とする。階数パラメータ $k \in \{2, 3, 4\}$ に対し，ML-DSA の安全性は $\mathbb{Z}_q$ 加群 $R_q^k \simeq (\mathbb{Z}_q^n)^k$ 上の LWE 問題の計算量困難性に依存する。また，ML-KEM と同様に，R_q における乗算を高速化するために，NTT 変換を利用する。ここでは，主に ML-DSA の構成と処理概要について説明する。

3.3.2.1 ML-DSA における NTT 変換

ML-DSA では, 2 のべき数 $n = 2^8 = 256$ と素数 $q = 2^{23} - 2^{13} + 1 = 8380417$ で定まる剰余環 $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ を用いる（ML-DSA の暗号パラメータについては，後述の 3.3.2.3 節を参照)。これらの暗号パラメータの組 (n, q) において，$\mathbb{Z}_q^*$ は位数 $q - 1 = 2^{13} \cdot 1023$ の巡回群である。ML-DSA では，$\mathbb{Z}_q$ における 1 の原始 512 乗根 $\zeta := 1753 \bmod q$ をとる。このとき，多項式環 $\mathbb{Z}_q[X]$ において，$X^n + 1$ は次のように n 個の 1 次式の積に分解できる。

$$X^n + 1 = \prod_{i=0}^{n-1} \left(X - \zeta^{2i+1}\right) = \prod_{i=0}^{n-1} \left(X - \zeta^{2\mathsf{BitRev}_8(i)+1}\right) \in \mathbb{Z}_q[X].$$

ただし，$\mathsf{BitRev}_8(i)$ は符号なし 8 ビット整数 i のビット逆順整数とし，ML-DSA ではこの順序を利用する。具体的には，各 $i = 0, 1, \ldots, n-1$ に対し $\zeta_i := \zeta^{2\mathsf{BitRev}_8(i)+1}$ とおき，環としての同型

$$\mathsf{NTT} : R_q \simeq \bigoplus_{i=0}^{n-1} \mathbb{Z}_q[X]/(X - \zeta_i) \simeq \bigoplus_{i=0}^{n-1} \mathbb{Z}_q =: T_q, \quad f \mapsto \widehat{f} := (f(\zeta_0), f(\zeta_1), \ldots, f(\zeta_{n-1}))$$

を用いて，R_q における乗算を効率的に行う。

3.3.2.2 ML-DSA の構成と処理概要

加群 $R_q^k \simeq (\mathbb{Z}_q^n)^k$ 上の LWE 問題に基づく ML-DSA は，下記のアルゴリズム群で構成される。ただし，ML-DSA の処理概要とその原理が分かるように，簡略化した形で各アルゴリズムの処理を説明する。

■ML-DSA 鍵生成　鍵生成アルゴリズム（FIPS 204, Algorithm 6）では，乱数 ξ を入力として，公開鍵 `pk` と秘密鍵 `sk` を次のように出力する（ただし，ℓ は次元パラメータとする)。

- 入力：乱数 ξ
- 出力：公開鍵 $\mathtt{pk}$ と秘密鍵 $\mathtt{sk}$
  1. $(\rho, \rho', K) = \mathsf{H}(\xi)$：ハッシュ関数 H で乱数 ξ から 3 つのデータの組 (ρ, ρ', K) を一意的に生成
  2. $\widehat{\mathbf{A}} = \mathsf{ExpandA}(\rho) \in (T_q)^{k \times \ell}$：擬似ランダムな行列 $\mathbf{A} \in (R_q)^{k \times \ell}$ を生成し，その NTT 表現を $\widehat{\mathbf{A}}$ を計算
  3. $(\mathbf{s}_1, \mathbf{s}_2) = \mathsf{ExpandS}(\rho') \in S_\eta^\ell \times S_\eta^k$：$S_\eta$ はすべての係数が $[-\eta, \eta]$ 内の R の元全体の集合（例：$\eta \in \{2, 4\}$）
  4. $\mathbf{t} = \mathsf{NTT}^{-1}\left(\widehat{\mathbf{A}} \circ \mathsf{NTT}(\mathbf{s}_1)\right) + \mathbf{s}_2 = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \in R_q^k$：NTT 変換を利用して計算
  5. $(\mathbf{t}_1, \mathbf{t}_0) = \mathsf{Power2Round}(\mathbf{t}) \in R_q^k \times R_q^k$：$\mathbf{t} \in R_q^k$ を上位と下位ビットに分割
  6. $\mathtt{pk} = (\rho, \mathbf{t}_1)$ とおき，そのハッシュ値 $tr = \mathsf{H}(\mathtt{pk})$ を計算
  7. $\mathtt{pk}$ と $\mathtt{sk} = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$ を出力

ステップ 2 において，ExpandA 関数（FIPS 204, Algorithm 32）は，乱数シード ρ から擬似ランダムな $\mathbf{A}$ を生成し，その NTT 表現 $\widehat{A}$ を計算する。ステップ 3 において，ExpandS 関数（FIPS 204, Algorithm 33）は，棄却サンプリングを用いてある範囲 $[-\eta, \eta]$ 内の係数をもつ R の元の組を生成する（$\eta \in \{2, 4\}$）。ステップ 5 において，Power2Round 関数（FIPS 204, Algorithm 35）を用いて，$\mathbf{t} \in R_q^k$ の各成分のすべての係数を上位と下位のビットに分割する。

鍵生成アルゴリズムにおいて，本質的に公開鍵 $\mathtt{pk}$ は $(\mathbf{A}, \mathbf{t})$ に対応し，その公開鍵に関する付属情報をいくつか含むが秘密鍵 $\mathtt{sk}$ は $(\mathbf{s}_1, \mathbf{s}_2)$ に対応する。公開鍵と秘密鍵の間に，R_q^k 上の LWE 関係式 $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$ が成り立つ。これより，公開鍵から秘密鍵を見つけるのは R_q^k 上の探索 LWE 問題で，適切な暗号パラメータ（後述の 3.3.2.3 節を参照）を利用した場合，その LWE 問題を解くのは計算量的に非常に困難である。

■ML-DSA 署名生成　署名生成アルゴリズム（FIPS 204, Algorithm 7）では，秘密鍵 $\mathtt{sk}$ と平文 M' を入力として，平文に対応する署名 σ を次のように出力する。

- 入力：秘密鍵 $\mathtt{sk} = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$ と平文 M'
- 出力：署名 σ
  1. $\widehat{\mathbf{s}}_1 = \mathsf{NTT}(\mathbf{s}_1) \in T_q^\ell, \widehat{\mathbf{s}}_2 = \mathsf{NTT}(\mathbf{s}_2) \in T_q^k, \widehat{\mathbf{t}}_0 = \mathsf{NTT}(\mathbf{t}_0) \in T_q^k$：NTT 表現を計算
  2. $\widehat{\mathbf{A}} = \mathsf{ExpandA}(\rho) \in (T_q)^{k \times \ell}$：$\rho$ から $\widehat{\mathbf{A}}$ を復元
  3. $\mu = \mathsf{H}(tr \| M')$：秘密鍵の一部 tr と平文 M' から定まるハッシュ値
  4. 次を繰り返す：
     (a) $\mathbf{y} = (\mathbf{y}[i])_{i=0}^{\ell} \in R_q^\ell$：各 $\mathbf{y}[i] \in R_q$ の各 $\mathbb{Z}_q$ 係数をある範囲で擬似ランダムにサンプル（十分小さい）
     (b) $\mathbf{w} = \mathsf{NTT}^{-1}\left(\widehat{\mathbf{A}} \circ \mathsf{NTT}(\mathbf{y})\right) = \mathbf{A}\mathbf{y} \in R_q^k$：NTT 変換を利用
     (c) $\mathbf{w}_1 = \mathsf{HighBits}(\mathbf{w}) \in R_q^k$：$\mathbf{w}$ の各成分の上位ビット
     (d) $\tilde{c} = \mathsf{H}(\mu \| \mathbf{w}_1)$
     (e) $c = \mathsf{SampleInBall}(\tilde{c}) \in R_q$：各係数を $\{-1, 0, 1\}$ からサンプルする（十分小さい）
     (f) $\widehat{c} = \mathsf{NTT}(c) \in T_q$
     (g) $c\mathbf{s}_1 = \mathsf{NTT}^{-1}(\widehat{c} \circ \widehat{\mathbf{s}}_1) \in R_q^\ell, c\mathbf{s}_2 = \mathsf{NTT}^{-1}(\widehat{c} \circ \widehat{\mathbf{s}}_2) \in R_q^k$：NTT 空間の乗算を利用
     (h) $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1 \in R_q^\ell$
     (i) $\mathbf{r}_0 = \mathsf{LowBits}(\mathbf{w} - c\mathbf{s}_2) \in R_q^k$：$\mathbf{w} - c\mathbf{s}_2$ の各成分の下位ビット
     (j) $\mathbf{z}$ と $\mathbf{r}_0$ のすべての $\mathbb{Z}_q$ 係数が十分小さい場合，次の処理を行う：
        i. $c\mathbf{t}_0 = \mathrm{NTT}^{-1}(\widehat{c} \circ \widehat{\mathbf{t}}_0) \in R_q^k$：NTT 空間の乗算を利用
        ii. $\mathbf{h} = \mathsf{MakeHint}\left(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0\right)$：長さ k の不一致真理値ベクトル
        $c\mathbf{t}_0$ のすべての $\mathbb{Z}_q$ 係数が十分小さく，かつ $\mathbf{h}$ 内の 1 の個数が十分少ないとき，ステップ 5 に進む
  5. $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$ を出力

ステップ 4 (e) において，乱数 $\tilde{c}$ を引数とする SampleInBall 関数（FIPS 204, Algorithm 29）で，すべての $\mathbb{Z}_q$ 係数を $\{-1, 0, 1\}$ からサンプルした多項式 $c \in R_q$ を生成する（ただし，係数ベクトルのハミング重みは 64 以下)。ステップ 4 (f) ii において，MakeHint 関数（FIPS 204, Algorithm 39）は，$\mathsf{HighBits}(\mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0)$ と $\mathsf{HighBits}(\mathbf{w} - c\mathbf{s}_2)$ の $\mathbb{Z}_q$ 係数の不一致真理値による長さ k のベクトル $\mathbf{h}$ を計算する。次の署名検証時で $\mathbf{w}_1$ を復元するために $\mathbf{h}$ を用いる。

署名生成アルゴリズムにおいて，ステップ 4 が主処理で，すべての $\mathbb{Z}_q$ 係数が十分小さい $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1 \in R_q^\ell$ を見つけるまで $\mathbf{y} \in R_q^\ell$ を取り直す。具体的には，擬似ランダムにサンプルしたすべての $\mathbb{Z}_q$ 係数が十分小さい $\mathbf{y} \in R_q^\ell$ から，コミットメント $\mathbf{w}_1$ を生成し，$\mathbf{w}_1$ と μ から定まるハッシュ値であるチャレンジ $\tilde{c}$ を求める。また，レスポンスとして，すべての $\mathbb{Z}_q$ 係数が十分小さい $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$ を生成する。チャレンジ $\tilde{c}$，レスポンス $\mathbf{z}$，コミットメント $\mathbf{w}_1$ のヒント $\mathbf{h}$ の 3 つの組 $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$ を平文に対応する署名とする。

■ML-DSA 署名検証　署名検証アルゴリズム（FIPS 204, Algorithm 8）では，公開鍵 $\mathtt{pk} = (\rho, \mathbf{t}_1)$ と署名 $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$ 付きの平文 M' を入力として，署名検証の結果を次のように真偽値で出力する。

- 入力：公開鍵 $\mathtt{pk} = (\rho, \mathbf{t}_1)$，署名 $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$ 付きの平文 M'
- 出力：真偽値
  1. $\widehat{\mathbf{A}} = \mathsf{ExpandA}(\rho)$：$\rho$ から $\widehat{\mathbf{A}}$ を復元
  2. $tr = \mathsf{H}(\mathtt{pk})$：pk のハッシュ値
  3. $\mu = \mathsf{H}(tr \| M')$：秘密鍵の一部 tr と平文 M' から定まるハッシュ値
  4. $c' = \mathsf{SampleInBall}(\tilde{c}) \in R_q$：各係数を $\{-1, 0, 1\}$ からサンプルする
  5. $\mathbf{w}'_{\mathrm{Approx}} = \mathsf{NTT}^{-1}\left(\widehat{\mathbf{A}} \circ \mathsf{NTT}(\mathbf{z}) - \mathsf{NTT}(c') \circ \mathsf{NTT}(\mathbf{t}_1 \cdot 2^d)\right) = \mathbf{A}\mathbf{z} - c'\mathbf{t}_1 \cdot 2^d \in R_q^\ell$
     （ただし，d は上位と下位ビットを分割する閾値）
  6. $\mathbf{w}'_1 = \mathsf{UseHint}(\mathbf{h}, \mathbf{w}'_{\mathrm{Approx}})$：署名生成時のコミットメントを復元
  7. $\tilde{c}' = \mathsf{H}(\mu \| \mathbf{w}'_1)$：$\mu$ と $\mathbf{w}'_1$ から定まるハッシュ値
  8. $\mathbf{z}$ のすべての $\mathbb{Z}_q$ 係数が十分小さく，かつ $\tilde{c} = \tilde{c}'$ のとき，「真」を出力。それ以外は，「偽」を出力

ステップ 6 において，UseHint 関数（FIPS 204, Algorithm 40）で，$\mathbf{w}'_{\mathrm{Approx}}$ が $\mathbf{w}$ に十分近いとき，ヒント $\mathbf{h}$ を元に署名生成時のコミットメント $\mathbf{w}_1$ を復元する（つまり，$\mathbf{w}'_1 = \mathbf{w}_1$)。具体的には，$\sigma$ が正当な署名であれば，$c' = c$ で $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$ なので，LWE 関係式 $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$ と $\mathbf{t}_1 \cdot 2^d \approx \mathbf{t}$（$\mathbf{t}_1$ は $\mathbf{t}$ の上位ビット）より

$$\begin{aligned}\mathbf{w}'_{\mathrm{Approx}} = \mathbf{A}\mathbf{z} - c\mathbf{t}_1 \cdot 2^d &= \mathbf{A}\mathbf{y} + c\mathbf{A}\mathbf{s}_1 - c\mathbf{t}_1 \cdot 2^d \\ &= \mathbf{w} + c(\mathbf{t} - \mathbf{s}_2) - c\mathbf{t}_1 \cdot 2^d \approx \mathbf{w} - c\mathbf{s}_2 \approx \mathbf{w}\end{aligned}$$

が成り立つ（$c\mathbf{s}_2 \in R_q^k$ のすべての $\mathbb{Z}_q$ 係数は十分小さいことに注意)。このとき，ステップ 7 で $\tilde{c}' = \tilde{c}$ となり検証に成功する。一方，平文 M' が改竄または署名 σ が偽造された場合は，非常に高い確率で $\tilde{c} \neq \tilde{c}'$ となり，検証に失敗する。

3.3.2.3　暗号パラメータ

ML-DSA における主な暗号パラメータと対応する鍵や署名のサイズと安全性レベルは以下である。具体的には，LWE の次元 $n = 256$ と剰余素数 $q = 8380417$ は ML-DSA-44, -65, -87 の 3 種類の暗号パラメータで共通であるが，主に公開鍵行列 $\mathbf{A} \in (R_q)^{k \times \ell}$ のサイズ (k, ℓ) により安全性レベルが異なる（特に，ML-DSA のパラメータ名は，(k, ℓ) により名づけられている)。

	暗号パラメータ			サイズ（単位：バイト）			安全性
	n	q	(k, ℓ)	秘密鍵	公開鍵	署名	レベル
ML-DSA-44	256	8380417	(4, 4)	2560	1312	2420	レベル 2
ML-DSA-65	256	8380417	(6, 5)	4032	1952	3309	レベル 3
ML-DSA-87	256	8380417	(8, 7)	4896	2592	4627	レベル 5

3.3.2.4 CRYSTALS-Dilithium との違い

- CRYSTALS-Dilithum の version 3.1 と round 3 version との違いは，安全性を確保するために，署名アルゴリズム内の秘密ランダムシード ρ' とメッセージ表現 μ の長さが 384 から 512 ビットへの増大である。加えて，公開鍵のハッシュに関する変数 tr のサイズを 384 から 256 ビットに減少させる一方，鍵生成において変数 ζ を ρ' に再ラベル付けし，そのサイズを 256 から 512 ビットに増大させている。
- MD-DSA と CRYSTALS-Dilithum の version 3.1 との違いについて，ML-DSA では tr の長さを 512 ビットに増やし，ML-DSA-65 と ML-DSA-87 のパラメータ設定それぞれで $\tilde{c}$ の長さを 384 と 512 ビットに増大している。CRYSTALS-Dilithum version 3.1 では，デフォルトの署名アルゴリズムは署名者の秘密鍵とメッセージから疑似ランダム生成された ρ' について確定的で，optional version では ρ' は 512 ビットのランダム列としてサンプリングされる。一方，ML-DSA では，ρ' は署名者の秘密鍵，メッセージ，と Approved RBG から生成された 256 ビットの文字列 rnd から生成される。また，ML-DSA 標準では，rnd が 256 ビットの定数文字列である optional deteministic version を許可している。

3.3.3 CRYSTALS-Kyber

歴史: CRYSTALS-Kyber は NIST PQC 標準化プロジェクトへの応募方式の一つとして 2017 年 11 月に Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancréde Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé の 10 名により共同で発表され [18]，その後 2018 年 4 月の国際会議 Euro S&P に Roberto Avanzi を除いた 9 名の共著により査読付き論文として発表された [33]. NIST PQC 標準化プロジェクトの第 3 ラウンドからは Jintai Ding が加わり 11 名での提案となった。NIST の耐量子計算機暗号標準化において唯一暗号化・鍵交換目的での Selected Algorithm として残った方式である [142]。

NIST PQC 標準化プロジェクトのラウンドが進むごとに主に暗号化処理のパラメータに関して修正が行われ，現在の最新版は 2021 年 8 月に公開されたバージョン 3.02[20] である。以下の記述はこの仕様書に従う。

参照 URL: 開発者による公式ページ `https://pq-crystals.org/kyber/` および GitHub のリファレンスコード `https://github.com/pq-crystals/kyber` を参照した。

設計原理: CRYSTALS-Kyber は Module-LWE 問題を安全性の根拠とする暗号化方式であり，dual-LWE 暗号方式をひな型[*6]として $x^{256}+1$ を定義多項式とした環上で処理を行うことで効率化している。

ベースとして IND-CPA 安全な公開鍵暗号を構成し，それを藤崎-岡本変換のデカプセル化失敗時の戻り値を調整した Hofheinz らの変種 [88] により IND-CCA2 安全な KEM へと変換している。

アルゴリズムの詳細: 表 3.3，3.4，3.5 に Lindner-Peikert[100] による格子ベース公開鍵暗号と CRYSTALS-Kyber の鍵生成，暗号化，復号アルゴリズムを並置する。

パブリックパラメータは以下で与えられる。

[*6] 仕様書では，アルゴリズムの形が Lyubashevsky-Peikert-Rosen の Ring-LWE ベース暗号 [106] に似ているとしている。

- n, q: 環を定義するための多項式 x^n+1 の次数と法を示す。用いられる多項式環は $R := \mathbb{Z}[x]/(x^n+1), R_q := \mathbb{Z}_q[x]/(x^n+1)$ であり，常に $n = 256, q = 3329 = 2^8 \cdot 13 + 1$ と固定されている[*7]。
- k: モジュール格子のランクとする。
- η_1, η_2: 鍵生成および暗号化時に生成するノイズベクトルの大きさを指定する。
- d_u, d_v: 暗号文多項式 $(\boldsymbol{u}, v) \in R_q^k \times R_q$ を表現するためのビット数を指定する。

用いられるサブルーチンのうち主なものを以下に列挙する。

- $\mathsf{NTT}(f)$ は $f = \sum_{i=0}^{255} f_i x^i \in R_q$ の NTT 表現 $\hat{f} = \sum_{i=0}^{255} \hat{f}_i x^i \in R_q$ を求める関数で，

$$\hat{f}_{2i} = \sum_{j=0}^{127} f_{2j} \zeta^{(2\mathsf{br}_7(i)+1)j} \quad \text{および} \quad \hat{f}_{2i+1} = \sum_{j=0}^{127} f_{2j+1} \zeta^{(2\mathsf{br}_7(i)+1)j}$$

  により定義される。ただし，$\mathsf{br}_7(i)$ は 7bits の整数を引数にとり，そのビット順序を反転した整数を出力する関数である。$\zeta = 17$ は $\mathbb{Z}_q$ における原始元である。
  この表記は複数の R_q の元を並べたベクトル $\boldsymbol{s} = (s_0, s_1, \ldots, s_{k-1}) \in R_q^k$ にも有効で，$\mathsf{NTT}(\boldsymbol{s}) = (\mathsf{NTT}(s_0), \mathsf{NTT}(s_1), \ldots, \mathsf{NTT}(s_{k-1}))$ 等と解釈する。
- $\mathsf{Parse}(\mathsf{XOF}(\rho, i, j))$: XOF (extendable output function) を用いてシードの ρ, i, j から十分な長さの擬似乱数列を生成し，それを Parse 関数により R_q の元に変換する。
- $\mathsf{CBD}_\eta(\mathsf{PRF}(\sigma, i))$: 大きさ $\eta \in \mathbb{N}$ の Central Binomial Distribution (CBD) を生成する。擬似乱数生成器 PRF は長さ 32Bytes の σ と 1Byte の i をシードとして 512ηbits の擬似乱数列 $\beta_0\beta_1 \cdots \beta_{512\eta-1}$ へと変換する。この列を 2ηbits ごとに切り分け $i = 0, \ldots, 255$ に対して $f_i = \sum_{j=0}^{\eta-1} \beta_{i\cdot 2\eta+j} - \sum_{j=0}^{\eta-1} \beta_{i\cdot 2\eta+\eta+j}$ を計算。i 次の係数を f_i とした 255 次多項式を CBD_η 関数の出力とする。
- $\mathsf{Encode}_\ell(\hat{\boldsymbol{s}}), \mathsf{Decode}_\ell(\boldsymbol{b})$: Encode_ℓ 関数は 255 次の多項式 $\hat{\boldsymbol{s}} \in R_q$ を入力とし，各係数を ℓbits のビット列に直したものを結合した 256ℓbits のビット列を出力とする。Decode 関数はその逆を行う関数で，ビット列を多項式環の元に変換する。
- $\mathsf{Compress}_q(x, d), \mathsf{Decompress}_q(x, d)$: $x \in \mathbb{Z}_q$ を近似的に $d$bits に変換，逆変換を行う関数であり，暗号文のサイズ削減に用いられる。具体的には

$$\begin{aligned} \mathsf{Compress}_q(x, d) &:= \lceil (2^d/q) \cdot x \rfloor \bmod 2^d, \text{および} \\ \mathsf{Decompress}_q(x, d) &:= \lceil (q/2^d) \cdot x \rfloor \end{aligned}$$

  で定義される。

擬似乱数生成器の実装について: アルゴリズムの仕様の中で用いられる擬似乱数生成器 XOF, PRF, G, H, KDF について，元々の SHAKE ハッシュ関数などを用いたものに加え，NIST PQC 標準化プロジェクト 第 2 ラウンドに合わせてアップデートされたバージョン 2.0[19] からは “90s version” として AES と SHA のみを用いたものが提案されている。これらの関数がデファクトスタンダードとして既に多くのハードウェア上で実装されている事から，高速化を狙ったものである。以下の表 3.2 に用いられる関数をまとめる。なお，本節で紹介する IND-CPA 安全な方式の中では XOF, PRF および G のみが用いられ，他の 2 つは IND-CCA2 安全な方式の構成において呼び出される。

90s version の XOF 関数では CTR モードの AES-256 を，ρ を鍵，12Bytes の nonce を $\mathrm{nonce}[0] = i$, $\mathrm{nonce}[1] = j$, $\mathrm{nonce}[\ell] = 0$ for $\ell = 2, \ldots, 11$ とパディングして用いる。同様に PRF 関数では AES-256 の CTR モードを ρ を鍵，

[*7] NIST PQC 標準化プロジェクト 第 1 ラウンド提出時には $q = 7681$ であったが，第 2 ラウンドからはこの値に変更された。

12Bytes の nonce を nonce[0] $= i$, nonce[ℓ] $= 0$ for $\ell = 1, \ldots, 11$ として用いる。また，オリジナルバージョンの SHAKE-128 の呼び出し方に関してはリファレンス実装[*8] を参照した。

表 3.2: CRYSTALS-Kyber における擬似乱数生成器の実装 [20, Sect. 1.4]

	$\mathsf{XOF}(\rho, i, j)$	$\mathsf{PRF}(\sigma, i)$	$\mathrm{H}(\boldsymbol{b})$	$\mathrm{G}(\boldsymbol{b})$	$\mathsf{KDF}(\boldsymbol{b})$
オリジナル	SHAKE-128$(\rho\|\|i\|\|j)$	SHAKE-256$(\sigma\|\|i)$	SHA3-256$(\boldsymbol{b})$	SHA3-512$(\boldsymbol{b})$	SHAKE-256$(\boldsymbol{b})$
90s	AES-256	AES3-256	SHA-256$(\boldsymbol{b})$	SHA-512$(\boldsymbol{b})$	SHA-256$(\boldsymbol{b})$

表 3.3: Lindner-Peikert 格子ベース暗号および CRYSTALS-Kyber における鍵生成関数の比較

	Lindner-Peikert [100, Sect. 3.1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	CRYSTALS-Kyber [20, Algorithm 4] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
0:		$d \xleftarrow{\$} \mathcal{B}^{32}$
		$(\rho, \sigma) \leftarrow \mathrm{G}(d)$ // $\mathcal{B}^{256} \times \mathcal{B}^{256}$ の疑似ランダムビット列
1:	A: $n_1 \times n_2$ ランダム行列	$\hat{A}[i][j] \leftarrow \mathsf{Parse}(\mathsf{XOF}(\rho, j, i))$
		for $i = 0, \ldots, k-1$ and $j = 0, \ldots, k-1$
2:	S: 成分の小さい $n_2 \times \ell$ 行列	$\boldsymbol{s}[i] \leftarrow \mathsf{CBD}_{\eta_1}(\mathsf{PRF}(\sigma, i))$ for $i = 0, \ldots, k-1$
		$\hat{\boldsymbol{s}} \leftarrow \mathsf{NTT}(\boldsymbol{s})$
3:	E: 成分の小さい $n_1 \times \ell$ 行列	$\boldsymbol{e}[i] \leftarrow \mathsf{CBD}_{\eta_1}(\mathsf{PRF}(\sigma, i+k))$ for $i = 0, \ldots, k-1$
		$\hat{\boldsymbol{e}} \leftarrow \mathsf{NTT}(\boldsymbol{e})$
4:	$B = AS + E$	$\hat{\boldsymbol{t}} \leftarrow \hat{A} \circ \hat{\boldsymbol{s}} + \hat{\boldsymbol{e}}$
return	$pk = (A, B), sk = S$	$pk = (\mathsf{Encode}_{12}(\hat{\boldsymbol{t}} \bmod q)\|\|\rho), sk = \mathsf{Encode}_{12}(\hat{\boldsymbol{s}} \bmod q)$

CRYSTALS-Kyber の鍵生成関数（表 3.3 右）を説明する。表の中で $\mathcal{B}$ は 1Byte 分の情報を表す集合 $\{0, 1, \ldots, 255\}$ を表す。ランダムに生成した 32Bytes の d をシードとして，ハッシュ関数 G を用いて 32Bytes の擬似ランダムビットの組 (ρ, σ) を生成する。これらはそれぞれ，行列 $A \in R_q^{k \times k}$ とノイズ多項式 $\boldsymbol{s}, \boldsymbol{e} \in R_q^k$ をサンプリングするためのシードとして用いられる。通常空間で R_q を一様ランダムにサンプルしたものに NTT をかけた後の分布はまた R_q 内の一様分布となるため，A は最初から NTT 空間でサンプリングされているものとみなされる。

$\boldsymbol{s}, \boldsymbol{e} \in R_q^k$ については CBD_{η_1} を用いて通常空間でのサンプリングを行い，その成分を個別に数論変換する。数論変換の性質により，最後の $\hat{\boldsymbol{t}}$ は $\mathsf{NTT}(A\boldsymbol{s} + \boldsymbol{e})$ となる。公開鍵サイズを圧縮するため，$A, \hat{\boldsymbol{t}}$ をそれぞれシード ρ，Encode 関数による圧縮形式で保存する。秘密鍵の $\hat{\boldsymbol{s}}$ に関しても同様である。

CRYSTALS-Kyber の暗号化関数（表 3.4 右）を説明する。圧縮形で入力された公開鍵から $\hat{\boldsymbol{t}}, \hat{A}$ を復元する。このとき，処理の効率化のために行列は転置された形で復元される。

暗号化のため成分の小さい $\boldsymbol{r}, \boldsymbol{e}_1 \in R_q^k$ と $e_2 \in R_q$ をサンプリングする。通常空間と NTT 空間を使い分けて処理を効率化しているが，最終的な暗号文 $c_1||c_2$ は通常空間でのベクトル $\boldsymbol{u} \in R_q^k$ と多項式 $v \in R_q$ を $\mathsf{Compress}_q$ 関数で圧縮したものとなる。ここで，2 種類のノイズ η_1, η_2 を使い分けるのは，η_1 のみによるノイズの大きさと，最後の Encode 関数によるラウンディングからの決定的ノイズと η_2 のノイズを合成したものの大きさが釣り合うように調整するためである [20, Sect. 1.5]。

CRYSTALS-Kyber の復号関数（表 3.5 右）は圧縮されたビット列の展開，NTT 空間の利用などで表現が煩雑になっているが，Lindner-Peikert 暗号の復号処理と本質的に同様である。最後の $\mathsf{Compress}_q(\cdot, 1)$ 関数が Lindner-Peikert 暗

[*8] https://github.com/pq-crystals/kyber/blob/master/ref/symmetric-shake.c, 2024/12/24 参照

表 3.4: Lindner-Peikert 格子ベース暗号および CRYSTALS-Kyber における暗号化関数の比較

	Lindner-Peikert[100, Sect. 3.1] $\mathsf{Enc}(pk=(A,B),\boldsymbol{m}\in\{0,1\}^{\ell})\to\mathsf{ct}$	CRYSTALS-Kyber [20, Algorithm 5] $\mathsf{Enc}(pk=(T\|\|\rho),m\in\mathcal{B}^{32})\to\mathsf{ct}$
0:		$\hat{\boldsymbol{t}}\leftarrow\mathsf{Decode}_{12}(T)$ $\hat{A}^T[i][j]\leftarrow\mathsf{Parse}(\mathsf{XOF}(\rho,i,j))$ //行列 $\hat{\boldsymbol{A}}$ の転置の形での復元
1:	$\boldsymbol{s}',\boldsymbol{e}',\boldsymbol{e}''$: 成分の小さいベクトル それぞれ Kyber の $\boldsymbol{r},\boldsymbol{e}_1,\boldsymbol{e}_2$ に対応	$\boldsymbol{r}[i]\leftarrow\mathsf{CBD}_{\eta_1}(\mathsf{PRF}(r,i))$ for $i=0,\ldots,k-1$ $\boldsymbol{e}_1[i]\leftarrow\mathsf{CBD}_{\eta_2}(\mathsf{PRF}(r,i+k))$ for $i=0,\ldots,k-1$ $e_2\leftarrow\mathsf{CBD}_{\eta_2}(\mathsf{PRF}(r,2k))$
2:	$\boldsymbol{u}=\boldsymbol{s}'A+\boldsymbol{e}'$ $\boldsymbol{v}=\boldsymbol{s}'B+\boldsymbol{e}''+\boldsymbol{m}\cdot\left\lfloor\frac{q}{2}\right\rfloor$	$\hat{\boldsymbol{r}}\leftarrow\mathsf{NTT}(\boldsymbol{r})$ $\boldsymbol{u}\leftarrow\mathsf{NTT}^{-1}(\hat{\boldsymbol{A}}^T\circ\hat{\boldsymbol{r}})+\boldsymbol{e}_1$ $v\leftarrow\mathsf{NTT}^{-1}(\hat{\boldsymbol{t}}^T\circ\hat{\boldsymbol{r}})+e_2+\mathsf{Decompress}_q(\mathsf{Decode}_1(m),1)$ $c_1\leftarrow\mathsf{Encode}_{d_u}(\mathsf{Compress}_q(\boldsymbol{u},d_u))$ $c_2\leftarrow\mathsf{Encode}_{d_v}(\mathsf{Compress}_q(v,d_v))$
return	$\mathsf{ct}=(\boldsymbol{u},\boldsymbol{v})$	$\mathsf{ct}=(c_1\|\|c_2)$

表 3.5: Lindner-Peikert 格子ベース暗号および CRYSTALS-Kyber における復号関数の比較

	Lindner-Peikert[100, Sect. 3.1] $\mathsf{Dec}(sk,\mathsf{ct})\to\boldsymbol{m}'$	CRYSTALS-Kyber [20, Algorithm 6] $\mathsf{Dec}(sk,\mathsf{ct}=(c_1\|\|c_2))\to m'\in\mathcal{B}^{32}$
1:	$\overline{\boldsymbol{m}}=\boldsymbol{v}-\boldsymbol{u}S$ $m'_i=\begin{cases}0 & \|\overline{m}_i\|\le\lfloor q/4\rfloor\\ 1 & \text{それ以外}\end{cases}$	$\boldsymbol{u}\leftarrow\mathsf{Decompress}_q(\mathsf{Decode}_{d_u}(c_1),d_u)$ $v\leftarrow\mathsf{Decompress}_q(\mathsf{Decode}_{d_v}(c_2),d_v)$ $\hat{\boldsymbol{s}}\leftarrow\mathsf{Decode}_{12}(sk)$ $m'\leftarrow\mathsf{Encode}_1(\mathsf{Compress}_q(v-\mathsf{NTT}^{-1}(\hat{\boldsymbol{s}}^T\circ\mathsf{NTT}(\boldsymbol{u})),1))$
return	$\boldsymbol{m}'=(m'_1,\ldots,m'_\ell)$	m'

号における $\overline{\boldsymbol{m}}$ から $\boldsymbol{m}'$ への変換に対応している。

安全性とパラメータ: ベースとなる IND-CPA 安全な公開鍵暗号の安全性は多項式環 $R_q:=\mathbb{Z}_q[x]/(x^n+1)$ 上の判定版 Module-LWE 問題へと ROM，QROM モデルの下で帰着される。

パラメータの設定は Module-LWE 問題を構造の無い LWE 問題とみなし Primal, Dual の双方の攻撃を BKZ アルゴリズムを用いて解いた場合の必要ブロックサイズに対応する Core SVP 計算量を通じて行われている。Module-LWE 問題へと帰着する際に，二項分布によるノイズと $\mathsf{Compress}_q$ 関数の四捨五入によるノイズを総合して詳細な解析を行っている。また，パラメータ設定用のスクリプトは [67] で公開されている。

暗号の性能を決めるパラメータは $n,k,q,\eta_1,\eta_2,d_u,d_v$ の 7 個であり，大まかに以下の特徴を持つ。格子の次元は多項式の次数 n と Module-LWE 問題のランク k の積であり，これらのパラメータを大きくとることで暗号の安全性が上がるが処理速度が低下し，鍵と暗号文のサイズが膨らむ。法 q を大きくとることでノイズ耐性が上がり復号エラー率が下がるが，格子が疎になり暗号の安全性が低下する。

(η_1,η_2) は鍵生成と暗号化に用いられるノイズ多項式の大きさで，大きくとることで暗号の安全性が上がるが復号エラー率が下がる。また，ノイズの中心二項分布を生成する際に必要とされるランダムビットの長さが増える。

(d_u,d_v) は暗号文 $(\boldsymbol{u},v)$ をビット列で表現するための精度を指定する。小さくとることで暗号文サイズが削減できるが，桁落ちが発生し復号エラー率が上がる。また，これらの値を小さくとることは暗号文にノイズを与えることにな

り，安全性が僅かではあるが向上するが，復号エラー率への影響の方が大きい。

安全性レベル 1,3,5 に対応するパラメータの値を表 3.6 に示す。また，δ は IND-CCA2 KEM おける復号エラー率を示す。正しい暗号文が KEM のデカプセル化 [20, Algorithm 9] で棄却される確率である。

表 3.6: CRYSTALS-Kyber のパラメータ [20, Table 1] および [5, Sect. D]. 公開鍵，秘密鍵，平文，暗号文サイズの単位はそれぞれ Byte である。

(n, k, q)	(η_1, η_2)	(d_u, d_v)	安全性レベル	公開鍵サイズ	秘密鍵サイズ	平文サイズ	暗号文サイズ	復号エラー率 δ
$(256, 2, 3329)$	$(3, 2)$	$(10, 4)$	レベル 1	800	$1,632$	32	768	2^{-139}
$(256, 3, 3329)$	$(2, 2)$	$(10, 4)$	レベル 3	$1,184$	$2,400$	32	$1,088$	2^{-164}
$(256, 4, 3329)$	$(2, 2)$	$(11, 5)$	レベル 5	$1,568$	$3,168$	32	$1,568$	2^{-174}

3.3.4 CRYSTALS-Dilithium

歴史: CRYSTALS-Dilithium は 2017 年 6 月に Cryptology ePrint Archive において Léo Ducas，Tancrède Lepoint，Vadim Lyubashevsky，Peter Schwabe，Gregor Seiler，Damien Stehlé の 6 名の連名で公表 [64] され，その後論文内での予告通りに 2017 年 11 月に NIST PQC 標準化プロジェクトへの応募方式 [62] として Eike Kiltz を加えた 7 名を開発者として提出された。査読付き論文としては国際会議 CHES 2018 において公開された版 [63] が存在する。

NIST PQC 標準化プロジェクトのラウンドが進むごとに微修正が行われ，現在の最新版は 2021 年 2 月に公開された仕様書 V3.1[21] である。本節の記述はこの仕様書に従う。

参照 URL: 開発者による公式ページ `https://pq-crystals.org/dilithium/` を参照した。

設計原理: CRYSTALS-Dilithium は格子ベースの署名方式であり，Lyubashevsky[104] による Fiat-Shamir with Aborts 型の構成を行っている。秘密鍵復元問題の安全性の根拠を，$x^{256} + 1$ を定義多項式とした環上における Module-LWE 問題に，署名の強偽造不可能性の根拠を SelfTargetMSIS 問題に置いている。通信コストを下げるため，公開鍵サイズと署名サイズの和の最小化を目的としてパラメータの設計を行っている。

最新の実装では，署名の検証にかかる計算時間の 80% はハッシュ関数 Keccak の処理時間であり，速度的にはこれ以上改良できない限界であるとしている [103]。

アルゴリズムの詳細: 表 3.7，3.8，3.10 に Lyubashevsky による Fiat-Shamir with Aborts 型の格子ベース署名，CRYSTALS-Dilithium のテンプレートアルゴリズム [21, Fig. 1] および実装のための擬似コード [21, Fig.4] を並置して記述する。

パブリックパラメータは以下で与えられる。

- n, q: 環を定義するための多項式 $x^n + 1$ の次数と法を示す。用いられる多項式環は $R := \mathbb{Z}[x]/(x^n + 1), R_q := \mathbb{Z}_q[x]/(x^n + 1)$ であり，提案方式の中では常に $n = 256, q = 2^{23} - 2^{13} + 1 = 8380417$ を用いる。
- k: モジュール格子のランクとする。
- l: ハッシュの (R_q における) 次元パラメータとする。
- d: 鍵生成時に $\boldsymbol{t}$ から分離する下位ビットの長さ
- η: 秘密鍵ベクトルのサンプリング空間の大きさ。

- τ: 署名生成時のベクトル $\boldsymbol{c}$ のサンプリング空間の大きさ。$\beta := \eta \cdot \tau$
- γ_1: 署名生成用ベクトル $\boldsymbol{y}$ のサンプリング空間の大きさ。
- γ_2: 署名生成用ベクトル $\boldsymbol{w}$ から取り出す上位ビットの長さ。

用いられるサブルーチンのうち主なものを以下に列挙する。

- NTT(a) は $\boldsymbol{a} = \sum_{i=0}^{255} a_i x^i$ の NTT 表現 $\hat{a} \in \mathbb{Z}_q^{256}$ を求める関数で,

$$\hat{a} = (a(r_0), a(-r_0), a(r_1), a(-r_1), \ldots, a(r_{127}), a(-r_{127}))$$

  で計算される。ただし，$r = 1753, r_i = r^{\mathrm{brv}(128+i)} \bmod q, \mathrm{brv}(k)$ 関数は k を 8bits の 2 進数としてみたときのビット順序を反転された数を出力するものとする。[21, Sect. 2.2]
- H: ビット列の伸長のためのハッシュ関数。CRYSTALS-Dilithium の実装では SHAKE256 ハッシュ関数を用いる。
- ExpandA(ρ): 乱数生成のシード ρ を用いて，ランダム行列 $A \in R_q^{k \times l}$ を生成し，その NTT 表現

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,l} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,l} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k,1} & a_{k,2} & \cdots & a_{k,l} \end{bmatrix} \to \hat{A} = \begin{bmatrix} \mathsf{NTT}(a_{1,1}) & \mathsf{NTT}(a_{1,2}) & \cdots & \mathsf{NTT}(a_{1,l}) \\ \mathsf{NTT}(a_{2,1}) & \mathsf{NTT}(a_{2,2}) & \cdots & \mathsf{NTT}(a_{2,l}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{NTT}(a_{k,1}) & \mathsf{NTT}(a_{k,2}) & \cdots & \mathsf{NTT}(a_{k,l}) \end{bmatrix}$$

  を計算し出力する。
- ExpandS(ρ'): 署名に用いる多項式 $\boldsymbol{s}_1, \boldsymbol{s}_2$ を生成するための関数で，512bits のシードを入力とする。
- $\mathsf{Power2Round}_q(t,d), \mathsf{HighBits}_q(t,\alpha), \mathsf{LowBits}_q(t,\alpha)$: $\mathbb{Z}_q$ の元 t で，$0 \le t < q$ を満たすものを $t = r_1 \cdot 2^d + r_0, -q/2 < r \le q/2$ と分解したときに $\mathsf{Power2Round}_q(t,d) = (r_1, r_0)$ と定義する。$\mathbb{Z}_q$ 成分の多項式 $t \in R_q$，および R_q 成分のベクトル $\boldsymbol{t}$ に対しても成分ごとに同様の操作を行うものとして定義する。具体的には，$\boldsymbol{t} = \left(\sum t_{j,i} x^i\right)_{j=1,\ldots,k}$ と書いたときに $\mathsf{Power2Round}_q\left(t_{j,i}, d\right) \to (t_{j,i,1}, t_{j,i,0})$ とすれば，$\mathsf{Power2Round}_q(\boldsymbol{t}, d) \to (\boldsymbol{t}_1, \boldsymbol{t}_0)$ は $\boldsymbol{t}_1 = \left(\sum q_{j,i} x^i\right)_{j=1,\ldots,k}, \boldsymbol{t}_0 = \left(\sum r_{j,i} x^i\right)_{j=1,\ldots,k}$, ただし $t_{j,i} = q_{j,i} \cdot 2^d + r_{j,i}$ と定義したものとする。

  また，α を $q-1$ の約数としたとき，同様に整数 t を $t = r_1 \cdot \alpha + r_0, -q/2 < r_0 \le q/2$ の形で分解し，$\mathsf{HighBits}_q(t,\alpha), \mathsf{LowBits}_q(t,\alpha)$ をそれぞれ r_1, r_0 で定義する。
- $\mathsf{MakeHint}_q(z,r,\alpha), \mathsf{UseHint}_q(h,r,\alpha)$: $\mathsf{MakeHint}_q$ 関数は $\mathsf{HighBits}_q(r,\alpha) \neq \mathsf{HighBits}_q(r+z,\alpha)$ であれば 1 を，そうでなければ 0 を返す関数である。$\mathsf{UseHint}_q$ 関数は引数から $\mathsf{HighBits}_q(r+z,\alpha)$ を復元する関数である。また，ベクトル $\mathbf{z} = (z_1, \ldots, z_k), \mathbf{r} = (r_1, \ldots, r_k)$ に対して，$\mathsf{MakeHint}_q(\mathbf{z}, \mathbf{r}, \alpha)$ は $\mathsf{MakeHint}_q(z_i, r_i, \alpha)$ を第 i 成分としたベクトルとする。復元成功の十分条件は [21, Lemma 4] で与えられている。
- SampleInBall($\tilde{c}$) 関数は係数のうち τ 個が ± 1 で，それ以外が 0 である多項式の集合 B_τ から一様サンプリングを行う。τ はパブリックパラメータとして与えられており，引数の $\tilde{c}$ はサンプリングのシードとして用いられる。生成された多項式 $c \in R$ の NTT 表現 $\hat{c} = \mathsf{NTT}(c)$ が出力される。
- $\#_1\boldsymbol{h}$ はベクトル $\boldsymbol{h} = (h_0, \ldots, h_{255})$ の中で，$h_i = 1$ となる成分の個数を返す。Dilithium の中では MakeHint 関数の出力となる 0-1 ベクトルであるため，ベクトルのハミング重みである。

表 3.7 の鍵生成関数について記述する。256bits のシード ζ をハッシュ関数 H により合計 1024bits に伸長し，そのうち ρ, ρ' をそれぞれ公開鍵 A のシード，秘密鍵 $\boldsymbol{s}_1, \boldsymbol{s}_2$ のシードとして用いる。鍵サイズ圧縮のため，行列 A はシード ρ の形で表現され，必要に応じて展開される。秘密鍵 $\boldsymbol{s}_1, \boldsymbol{s}_2$ は R の元をそれぞれ ℓ, k 個並べたベクトルであり，各成分は集合 $S_\eta = \{\boldsymbol{w} \in R : \|\boldsymbol{w}\|_\infty \le \eta\}$ から一様ランダムにサンプリングされる。

表 3.7: CRYSTALS-Dilithium における鍵生成関数の比較

	格子ベース署名 [104, Fig. 4] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	CRYSTALS-Dilithium テンプレート [21, Fig. 1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	CRYSTALS-Dilithium 実装のための擬似コード [21, Fig. 4] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
1:	$\hat{\boldsymbol{s}}$: 短い多項式を 成分とするベクトル	$\boldsymbol{s}_1 \leftarrow S_\eta^l, \boldsymbol{s}_2 \leftarrow S_\eta^k$	$\zeta \xleftarrow{\$} \{0,1\}^{256}$ $\mathsf{H}(\zeta) \to (\rho, \rho', K) \in \{0,1\}^{256} \times \{0,1\}^{512} \times \{0,1\}^{256}$ $\mathsf{ExpandS}(\rho') \to (\boldsymbol{s}_1, \boldsymbol{s}_2) \in S_\eta^\ell \times S_\eta^k$
2:	a: ハッシュ関数	$\hat{A} \xleftarrow{\$} R_q^{k \times l}$	$\mathsf{ExpandA}(\rho) \to \hat{A} \in R_q^{k \times \ell}$
3:	$t \leftarrow a(\hat{\boldsymbol{s}})$	$\boldsymbol{t} = A\boldsymbol{s}_1 + \boldsymbol{s}_2$	$\boldsymbol{t} \leftarrow \mathsf{NTT}^{-1}(\hat{A} \cdot \mathsf{NTT}(\boldsymbol{s}_1)) = A\boldsymbol{s}_1 + \boldsymbol{s}_2$ $\mathsf{Power2Round_q}(\boldsymbol{t}, d) \quad \to (\boldsymbol{t}_1, \boldsymbol{t}_0)$ $\mathsf{H}(\rho \| \boldsymbol{t}_1) \to tr \in \{0,1\}^{256}$
return	$sk = (a, \hat{\boldsymbol{s}}), pk = (a, \boldsymbol{t})$	$sk = (A, \boldsymbol{t}, s_1, s_2), pk = (A, \boldsymbol{t})$	$sk = (\rho, K, tr, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0), pk = (\rho, \boldsymbol{t}_1)$

Step 2 および 3 では Fiat-Shamir 型格子署名における秘密鍵 $\hat{\boldsymbol{s}}$ のハッシュ関数 $a(\hat{\boldsymbol{s}})$ の計算が，ベクトル $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ と行列 A を用いた $A\boldsymbol{s}_1 + \boldsymbol{s}_2$ の計算に対応している。計算されたベクトル $\boldsymbol{t} \in R_q^k$ に対して，$\mathsf{Power2Round}_q$ 関数により上位ビットと下位ビットに分割する。

最後に，メッセージに連結するためのランダムビット tr をハッシュ関数 H を用いて生成する。

表 3.8: CRYSTALS-Dilithium における署名生成関数の比較

	格子ベース署名 [104, Fig. 4] $\mathsf{Sign}(sk = (a, \hat{\boldsymbol{s}}),$ $\mu \in \{0,1\}^*) \to \sigma$	CRYSTALS-Dilithium テンプレート [21, Fig. 1] $\mathsf{Sign}(sk = (A, t, \boldsymbol{s}_1, \boldsymbol{s}_2),$ $\mu \in \{0,1\}^*) \to \sigma$	CRYSTALS-Dilithium 実装のための擬似コード [21, Fig. 4] $\mathsf{Sign}(sk = (\rho, K, tr, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0),$ $M \in \{0,1\}^*\}) \to \sigma$
0:			$\mathsf{ExpandA}(\rho) \to \hat{A}$ $\mathsf{H}(tr \| M) \to \mu \in \{0,1\}^{512}$
1:	$z \leftarrow \perp$	$z \leftarrow \perp$	$\kappa \leftarrow 0, (\boldsymbol{z}, \boldsymbol{h}) \leftarrow \perp$ $\mathsf{H}(K \| \mu) \to \rho' \in \{0,1\}^{512}$ $\hat{\boldsymbol{s}}_1 \leftarrow \mathsf{NTT}(\boldsymbol{s}_1)$; $\hat{\boldsymbol{s}}_2 \leftarrow \mathsf{NTT}(\boldsymbol{s}_2)$ $\hat{\boldsymbol{t}}_0 \leftarrow \mathsf{NTT}(\boldsymbol{t}_0)$
2:	**while** $z = \perp$ **do**	**while** $z = \perp$ **do**	**while** $(\boldsymbol{z}, \boldsymbol{h}) = \perp$ **do**
3:	$\hat{\boldsymbol{y}}$: 短い多項式を 成分とするベクトル	$\boldsymbol{y} \leftarrow D_{\gamma_1 - 1}^{l \times 1}$	$\mathsf{ExpandMask}(\rho', \kappa) \to \boldsymbol{y} \in \tilde{S}_{\gamma_1}^l$
4:	$c \leftarrow H(a(\hat{\boldsymbol{y}}) \| \mu)$	$\boldsymbol{w}_1 \leftarrow \mathsf{HighBits}(A\boldsymbol{y}, 2\gamma_2)$ $c = H(\mu \| \boldsymbol{w}_1)$	$\boldsymbol{w} \leftarrow \mathsf{NTT}^{-1}(\hat{A} \cdot \mathsf{NTT}(\boldsymbol{y})) = \boldsymbol{Ay}$ $\boldsymbol{w}_1 \leftarrow \mathsf{HighBits}_q(\boldsymbol{w}, 2\gamma_2)$ $H(\mu \| \boldsymbol{w}_1) \to \tilde{c} \in \{0,1\}^{256}$ $\mathsf{SampleInBall}(\tilde{c}) \to \hat{c} \in B_\tau \subset R_q$
5:	$\hat{\boldsymbol{z}} \leftarrow \hat{\boldsymbol{y}} + c\hat{\boldsymbol{s}}$ if $\hat{\boldsymbol{z}} \notin G^m$ **then** $z \leftarrow \perp$	$\boldsymbol{z} \leftarrow \boldsymbol{y} + c\boldsymbol{s}_1$ $\boldsymbol{r}_0 \leftarrow \mathsf{LowBits}(A\boldsymbol{y} - c\boldsymbol{s}2, 2\gamma_2)$ **if** $(\|\boldsymbol{z}\|_\infty \geq \gamma_1 - \beta)$ OR $(\|\boldsymbol{r}_0\|_\infty \geq \gamma_2 - \beta)$ **then** $z \leftarrow \perp$	$\boldsymbol{z} \leftarrow \boldsymbol{y} + \mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\boldsymbol{s}}_1)$ $\boldsymbol{r}_0 \leftarrow \mathsf{LowBits}_q(\boldsymbol{w} - \mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\boldsymbol{s}}_2), 2\gamma_2)$ **if** $(\|\boldsymbol{z}\|_\infty \geq \gamma_1 - \beta)$ OR $(\|\boldsymbol{r}_0\|_\infty \geq \gamma_2 - \beta)$ **then** $(\boldsymbol{z}, \boldsymbol{h}) \leftarrow \perp$ **else** $\boldsymbol{h} \leftarrow \mathsf{MakeHint}_q(\cdot)$ ... (*) $\kappa \leftarrow \kappa + l$
return	$\sigma = (\hat{\boldsymbol{z}}, c)$	$\sigma = (\boldsymbol{z}, c)$	$\sigma = (\boldsymbol{z}, \boldsymbol{h}, \tilde{c})$

表 3.8 の署名生成関数について記述する。ρ から行列 A の NTT 表現 $\hat{A}$ を復元する。ランダムビット tr を用いてメッセージのハッシュ値 μ を計算し，この値に署名をつける。κ は ExpandMask 関数の中で呼び出す SHAKE256 の

シードとなる値で，$\mathsf{H}(K||\mu) \to \rho'$ とともに用いられる。計算効率化の目的で R_q の元の乗算には NTT 表現を用いるため，予め $\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0$ を NTT 表現に変換しておく。

Fiat-Shamir 型署名の標準的な構成方法と同様に，署名の初期値 $(\boldsymbol{z}, \boldsymbol{h})$ を $\perp$ とし，**while** ループの中で生成された署名が集合 G に含まれているかどうかを検査し含まれていない場合にはループをやり直す。

ExpandMask 関数の中では，(ρ', κ) をシードとしてランダムベクトル $\boldsymbol{y} \in R_q^l$ をサンプリングする。ここで，各成分は $\tilde{S}_{\gamma_1} = \left\{ \sum_{i=0}^{255} w_i x^i : -\eta < w_i \leq \eta \right\}$ から一様ランダムにサンプリングされる。このサンプリングは表 3.8 中央の $\boldsymbol{y} \leftarrow D_{\gamma_1}^{l \times 1}$ に対応する。

署名生成のためのベクトル $c \in R_q^l$ は 256bits のシード $\tilde{c}$ により表現され，この値自体は μ と $\boldsymbol{w}_1$ を連結したハッシュ値から計算される。ここで，μ はメッセージからの要素であり，$\boldsymbol{w}_1$ は公開鍵 A と直前でサンプリングした $\boldsymbol{y}$ から来る要素である。計算効率のため，内積 $c \cdot \boldsymbol{s}_1$ は NTT 表現で計算された後に逆変換をかけ $\boldsymbol{z} = \boldsymbol{y} + c \cdot \boldsymbol{s}_1$ となる。

ステップ 5 では $z \notin G$ のチェックのため，$\boldsymbol{z}$ と $\boldsymbol{w} - c\boldsymbol{s}_2$ の下位ビットの ℓ_∞ ノルムがそれぞれ比較される。両方が閾値よりも小さい場合には次のヒント生成関数 $(*)$ が実行される。ヒント生成関数は表 3.9 により示され，$\mathsf{MakeHint}_q$ 実行後に再びノルムの大きさがチェックされ，閾値よりも大きな場合には $(\boldsymbol{z}, \boldsymbol{h}) \leftarrow \perp$ となる。つまり，2 回の **if** 文の中での 4 回の不等号検査のうち一つでも満たされない条件があれば，シード κ を増やし $\boldsymbol{y}$ の生成からやり直すことになる。ここで，$\mathsf{MakeHint}_q$ 関数の中での $-c\boldsymbol{s}_2 + c\boldsymbol{t}_0$ の計算は前半を $\boldsymbol{r}_0$ で用いたものを使いまわし，後半を $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\boldsymbol{t}}_0)$ の形で計算する。

表 3.9: 署名生成関数におけるヒント生成時のチェック関数

$\boldsymbol{h} \leftarrow \mathsf{MakeHint}_q(-c\boldsymbol{t}_0, \boldsymbol{w} - c\boldsymbol{s}_2 + c\boldsymbol{t}_0, 2\gamma_2)$
if $\|c\boldsymbol{t}_0\|_\infty \geq \gamma_2$ OR $\#_1\boldsymbol{h} > \omega$ **then** $(\boldsymbol{z}, \boldsymbol{h}) \leftarrow \perp$

表 3.10: CRYSTALS-Dilithium における署名検証関数の比較

	格子ベース署名 [104, Fig. 4]	CRYSTALS-Dilithium テンプレート [21, Fig. 1]	CRYSTALS-Dilithium 実装のための擬似コード [21, Fig. 4]
	$\mathsf{Vrfy}(pk = (a, \boldsymbol{t}), \mu \in \{0,1\}^*,$ $\sigma = (\hat{\boldsymbol{z}}, c))$	$\mathsf{Vrfy}(pk = (A, \boldsymbol{t}), \mu \in \{0,1\}^*,$ $\sigma = (\boldsymbol{z}, c))$	$\mathsf{Vrfy}(pk = (\rho, \boldsymbol{t}_1), M \in \{0,1\}^*,$ $\sigma = (\boldsymbol{z}, \boldsymbol{h}, \tilde{c}))$
0:			$\mathsf{ExpandA}(\rho) \to \hat{A}$ $\mathsf{H}(\mathsf{H}(\rho\|\|\boldsymbol{t}_1)\|\|M) \to \mu \in \{0,1\}^{512}$ $\mathsf{SampleInBall}(\tilde{c}) \to \hat{c}$
1:	**if** $\hat{\boldsymbol{z}} \in G^m$ AND $c = H(a(\hat{\boldsymbol{z}}) - \boldsymbol{t}c, \mu)$ **then** accept **else** reject	$\boldsymbol{w}_1' = \mathsf{HighBits}(A\boldsymbol{z} - c\boldsymbol{t}, 2\gamma_2)$ **if** $\|\boldsymbol{z}\|_\infty < \gamma_1 - \beta$ AND $c = H(M\|\|\boldsymbol{w}_1')$ **then** accept **else** reject	$\boldsymbol{w}_1' \leftarrow \mathsf{UseHint}_q(\boldsymbol{h}, A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d, 2\gamma_2)$ **if** $\|\boldsymbol{z}\|_\infty < \gamma_1 - \beta$ AND $\tilde{c} = H(\mu\|\|\boldsymbol{w}_1')$ AND $\#_1\boldsymbol{h} \leq \omega$ **then** accept **else** reject

表 3.10 の署名検証関数について記述する。公開鍵，署名に含まれる乱数のシード $\rho, \tilde{c}$ から $\hat{A}, \hat{c}$ を復元し，メッセージに対応するハッシュ値 μ を計算する。$A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d$ は $\hat{A} \cdot \mathsf{NTT}(\boldsymbol{z}) - \mathsf{NTT}(c) \cdot \mathsf{NTT}(\boldsymbol{t}_1 \cdot 2^d)$ の形で計算する。これらの値から $\mathsf{UseHint}_q$ を用いて $\boldsymbol{w}_1'$ を復元し，$\boldsymbol{z}$ のノルム，$\boldsymbol{h}$ の 1 の数の確認を行い，正しければ accept を出力する。

安全性とパラメータ: CRYSTALS-Dilithium の安全性は，$x^n + 1$ を定義多項式とする環上のモジュール格子問題である。ROM の下で，秘密鍵復元の困難性が Module-LWE 問題に，署名の強偽造不可能性が SelfTargetMSIS 問題にそれぞれ帰着される。SelfTargetMSIS 問題は Module-SIS 問題の変種であり，署名の偽造不可能性からのタイトな古典帰着が知られている。一方で，Module-SIS 問題への古典帰着もタイトではないものの証明が与えられており，その意味では Module-SIS 問題を安全性の根拠と捉えることもできる。

方式の発表後，仕様書内の安全性証明における理論の飛躍が発見された [26, p.3]。具体的には署名の受動的攻撃下における強偽造不可能性 (EUF-NMA) から選択平文攻撃下における強偽造不可能性 (EUF-CMA 安全性) への帰着を行う際に，アドバーサリーが想定された挙動を行うためには秘密鍵の情報を必要とするため，確率的多項式時間であることが保証できない。Barbosa らは 2023 年に問題点と修正，およびコンピュータ支援による安全性証明の論文 [26] を発表している。なお，証明の修正であるため方式自体の変更は行われていない。

一方で，QROM においても鍵復元，署名偽造が同様に Module-LWE 問題，SelfTargetMSIS 問題それぞれ帰着されるものの Module-SIS 問題までの量子帰着が知られていない。

具体的なパラメータは，LWE 問題と SIS 問題の双方に対して BKZ アルゴリズムで解いた際の必要ブロックサイズと Core-SVP の見積から求められている。

仕様書に掲載されたパラメータセットを表 3.11 に示す。セキュリティ強度を規定するパラメータのうち，問題が定義される環とモジュールのランクに関わるものが (n, k, l, q) の 4 個，ノイズに関わるものが $(\eta, \gamma_1, \gamma_2, \beta, \tau, d)$ の 6 個である。

表 3.11: CRYSTALS-Dilithium 署名方式のパラメータ [21, Table 1] , [5, Table 8]。公開鍵，秘密鍵，署名サイズの単位はそれぞれ Byte である。

(n, k, l, q)	$(\eta, \gamma_1, \gamma_2, \beta, \tau, d)$	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ
$(256, 4, 4, 8380417)$	$(2, 2^{17},\ 95232,\ 78, 49, 13)$	レベル 2	$1,312$	$2,528$	$2,420$
$(256, 6, 5, 8380417)$	$(4, 2^{19}, 261888, 196, 49, 13)$	レベル 3	$1,952$	$4,000$	$3,293$
$(256, 8, 7, 8380417)$	$(2, 2^{19}, 261888, 120, 60, 13)$	レベル 5	$2,592$	$4,864$	$4,595$

注： 秘密鍵サイズは仕様書 [21] には掲載されていないが，NIST の第 3 ラウンド報告レポート [5] を参照した。

変種: [21, Table 3] には NIST の提唱する安全性レベル 1 よりも弱いパラメータ，安全性レベル 5 よりも強いパラメータが掲載されている。NIST PQC 標準化が決定して以降仕様書のバージョン [21] から様々な変更が加えられ，標準方式 ML-DSA (3.3.2 節も参照) が公開された。変更の概要は本報告書 3.3.2.4 節を参照。

補足情報: NIST PQC 標準化プロジェクトの第 3 ラウンド報告レポートにおいて署名方式 FALCON との比較が行われ，CRYSTALS-Dilithium はそのシンプルさから一般的な実装に向いているが，FALCON は署名の短さからリソースの制限されたデバイスで使われることが期待されている [5, p.19]。

3.3.5 FALCON

歴史: FALCON は 2017 年 11 月の NIST PQC 標準化プロジェクトの公募に Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang の 10 名を開発者として公表された [79]。その後修正が加えられ，現在の最新版は 2020 年 10 月に公開された v1.2[80] である。以下の記述はこの仕様書に従う。

参照 URL: 開発者による公式ページ `https://falcon-sign.info/` を参照した。

設計原理: FALCON は多項式 $x^n + 1, n = 2^k$ により定義される NTRU 格子上の SIS 問題の困難性を安全性の根拠とした格子ベースの署名方式であり，形式的には Gentry ら [82] の Hash-and-Sign 型の格子ベース署名をひな型としている。高速フーリエサンプリングを用いるため，定義多項式の次数を 2^k の形としていることからパラメータ選択の自由度に制限があり，NIST PQC 標準化プロジェクトの提案方式では安全性レベル 1 および 5 のパラメータセットのみが提案されている。

アルゴリズムの詳細: 表 3.12，3.13，3.14 に，Gentry ら [82] の Hash-and-Sign 型格子ベース署名と FALCON の鍵生成，署名生成，署名検証関数を並置する。

パブリックパラメータは以下で与えられる。

- n, q: 環を定義する多項式 $\phi(x) = x^n + 1$ と法 q で，演算は $\mathbb{Z}_q[x]/(\phi)$ で行われる。
- σ: 離散 Gauss 分布の大きさを指定する。
- β: 有効な署名のノルムの上限を指定する。

アルゴリズム中で用いられるサブルーチンのうち，主なものを列挙する。

- $\mathsf{FFT}(f), \mathsf{invFFT}(\boldsymbol{s})$: 多項式 $f \in \mathbb{R}[x]/(\phi)$ に対して，そのフーリエ変換 $\mathsf{FFT}(f)$ を n 次元ベクトル $(f(\zeta_k))_{k=0,\ldots,n-1}$ で定義する[*9]。ただし，$\zeta_k := \exp((2k+1)\pi i/n)$。逆演算を $\mathsf{invFFT} : \mathbb{R}^n \to \mathbb{R}[x]/(\phi)$ で示す。変換，逆変換ともに標準的な高速フーリエ変換の手法が利用可能である。コンピュータ上での計算には浮動小数点演算を用いるため，実行環境ごとに差が出ないように IEEE754 で規定される浮動小数点の表現と演算を用いることが指定されている。
  多項式を成分とするベクトル，行列に対しても FFT は成分ごとのフーリエ変換と定義し，invFFT も適切な切り分けにより実数成分の行列，ベクトルから多項式成分の行列，ベクトルへ変換するものとする。
  また，演算 $\mathsf{FFT}(f) \odot \mathsf{FFT}(g)$ を成分ごとの積と定義する。FFT 表現での多項式の積 $\mathsf{FFT}(fg)$ の計算に対応する。
- $\mathsf{HashToPoint}(\mathsf{str}, q, n)$: ビット列 str を多項式 $c \in \mathbb{Z}_q[x]/(\phi)$ に SHAKE256 ハッシュ関数を用いて写像する。
- Compress, Decompress: 多項式 $s \in \mathbb{Z}[x]$ を文字列に変換する関数とその逆関数とする。
- $\mathsf{NTRUGen}(\phi, q)$: 計算が行われる環 $\mathbb{Z}_q[x]/(\phi)$ を指定するパラメータを入力とし，秘密鍵 $\hat{B}$ の元となる多項式 f, g, F, G を出力する。このとき，f, g は係数が離散 Gauss 分布の n 次多項式，F, G は $fG - gF \equiv q \mod \phi$ を満たすように計算される。

表 3.12: Hash-and-Sign 型格子ベース署名および FALCON における鍵生成関数の比較

	Gentry らの格子ベース署名 [82, Sect. 7.1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	FALCON[80, Algorithm 4] $\mathsf{KeyGen}(\phi, q) \to (pk, sk)$
1:	$BA \equiv 0 \pmod q$ を満たす行列の組 (A, B) を生成 B: 成分の小さい行列 A: ランダム行列	$f, g, F, G \leftarrow \mathsf{NTRUGen}(\phi, q)$ $B \leftarrow \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$ $\hat{B} \leftarrow \mathsf{FFT}(B)$ $G \leftarrow \hat{B} \times \hat{B}^*$ $T \leftarrow \mathsf{ffLDL}^*(G)$ **for** each leaf leaf of T **do** $\quad \mathsf{leaf.value} \leftarrow \sigma/\sqrt{\mathsf{leaf.value}}$ $h \leftarrow gf^{-1} \bmod q$
return	$pk = A, sk = B$	$pk = h, sk = (\hat{B}, T)$

[*9] 数式上は差が無いが高速フーリエ変換による実装を行ったサブルーチンも同じ記号で示すため，Fast Fourier の意味で FFT と名づけられている。

NTRU 型暗号の秘密鍵 (f, g) のうち，f は環 $\mathbb{Z}_q/(\phi)$ の中で逆元を持つため，適当な $F, G \in \mathbb{Z}[x]$ を用いて

$$fG - gF = q \bmod \phi \tag{3.3}$$

と書くことができる。この関係式と公開鍵 $h = f^{-1}g$ を Hash-and-Sign フレームワーク [82] における行列 A, B と捉えると，

$$A = \begin{bmatrix} 1 \\ h \end{bmatrix},\ B = \begin{bmatrix} g & -f \\ G & -F \end{bmatrix} \tag{3.4}$$

と表現することができる。このとき，行列 A は多項式 h の情報のみで表現可能であるため，$pk = h$ となる。

また，署名の生成には $\boldsymbol{s}A \equiv H(m)$ を満たす短いベクトル $\boldsymbol{s}$ を生成する必要があり，効率化のため Ducas-Prest[66] の高速フーリエサンプリングを用いる。サンプリングアルゴリズムに必要な情報が B の FFT 表現

$$\mathsf{FFT}(B) = \begin{bmatrix} \mathsf{FFT}(g) & \mathsf{FFT}(-f) \\ \mathsf{FFT}(G) & \mathsf{FFT}(-F) \end{bmatrix} \tag{3.5}$$

およびそれを元にした LDL 木と呼ばれる木構造 T である。木の中には $\hat{B}$ のグラム行列 $G = \hat{B} \times \hat{B}^{\star}$ の[*10] LDL 分解における L の情報が格納され，それを用いて Babai の最近平面アルゴリズムの高速化および離散 Gauss 分布の高速なサンプリングが可能となる。サンプリングを行うための付加情報として，木の全ての葉にある値を $\mathsf{leaf.value}$ から $\sigma/\sqrt{\mathsf{leaf.value}}$ に書き換えることで鍵生成が完了する。

表 3.13: Hash-and-Sign 型格子ベース署名および FALCON における署名生成関数の比較

	Gentry らの格子ベース署名 [82, Sect. 7.1] $\mathsf{Sign}(sk = (\hat{B}, T), m \in \{0,1\}^*) \to \sigma$	FALCON[80, Algorithm 10] $\mathsf{Sign}(sk = (\hat{B}, T), m \in \{0,1\}^*, \lfloor \beta^2 \rfloor) \to \sigma$
1:	$\boldsymbol{c} \leftarrow H(m)$ //平文のハッシュ値をベクトル化	$r \leftarrow \{0,1\}^{320}$ $c \leftarrow \mathsf{HashToPoint}(r \| m, q, n)$ $\hat{\boldsymbol{t}} \leftarrow \left(-\frac{1}{q}\mathsf{FFT}(c) \odot \mathsf{FFT}(F), \frac{1}{q}\mathsf{FFT}(c) \odot \mathsf{FFT}(f)\right)$
2:	T を使い，$\boldsymbol{s}A \equiv \boldsymbol{c} \pmod q$ を 満たすベクトル $\boldsymbol{s}$ をサンプリング	**do** 　**do** 　　$\boldsymbol{z} \leftarrow \mathsf{ffSampling}_n(\hat{\boldsymbol{t}}, T)$ 　　$\hat{\boldsymbol{s}} \leftarrow (\hat{\boldsymbol{t}} - \hat{\boldsymbol{z}})\hat{B}$ 　**while** $\|\boldsymbol{s}\|^2 > \lfloor \beta^2 \rfloor$ 　$(s_1, s_2) \leftarrow \mathsf{invFFT}(\hat{\boldsymbol{s}})$ 　$\mathsf{s} \leftarrow \mathsf{Compress}(s_2, 8 \cdot \mathsf{sbytelen} - 328)$ **while** $(\mathsf{s} = \perp)$
return	$\sigma = \boldsymbol{s}$	$\sigma = (r, \mathsf{s})$

表 3.13 の署名生成関数の説明を記述する。平文にランダムビット r を結合した後，HashToPoint 関数で多項式 $c \in \mathbb{Z}_q/(\phi)$ を出力する。関係式 (3.3), (3.4) より，ベクトル $\hat{\boldsymbol{t}}$ は $(\mathsf{FFT}(c), \mathsf{FFT}(0))\hat{B}^{-1}$ と等しい事がわかる。これらの情報を用いて，署名ベクトルのサンプリングを行う。

関数 $\mathsf{ffSampling_n}$ は，離散 Gauss 分布のサンプリングを行い，FFT 表現で出力するサブルーチンである。具体的には，整数ベクトル $\boldsymbol{z} \in \mathbb{Z}^{2n}$ を，$\boldsymbol{t} = [c, 0]B^{-1}$ を中心として $\exp(-\|(\boldsymbol{z} - \boldsymbol{t})B\|^2/2\sigma^2)$ に比例した確率でサンプリング

[*10] $B^{\star}$ は体 $\mathbb{Q}[x]/(\phi)$ におけるエルミート共役。詳細は [80, p.23]

を行う。実装の効率化のため，実際には近似を行っている [80, Sect. 3.9.1, 3.9.2]。このとき，$(\boldsymbol{t}-\boldsymbol{z})B$ は原点を中心とした集合

$$\boldsymbol{t}+\Lambda(B)=\{(c,0)+x\in(\mathbb{Z}[x]/(\phi))^2 : x\in\Lambda(B)\}$$

上の離散 Gauss 分布となるため，$\boldsymbol{s}$ は短く，かつ

$$\boldsymbol{s}A\equiv([c,0]B^{-1}-\boldsymbol{z})BA\equiv[c,0]\begin{bmatrix}1\\ h\end{bmatrix}=c \text{ in } \mathbb{Z}_q[x]/(\phi)$$

が成り立つ。このとき，$\boldsymbol{s}A=c$ の関係から $s_1+s_2h=c$ が成り立つ。この関係式が署名の検証時に用いられる。

サンプリングされた $\hat{\boldsymbol{s}}$ が $\|\hat{\boldsymbol{s}}\|^2\le\lfloor\beta^2\rfloor$ を満たしていれば invFFT により通常空間の表現に戻し，Compress 関数を用いて圧縮された文字列 s を生成し，ハッシュ関数のシード r とともに署名とする。

表 3.14: Hash-and-Sign 型格子ベース署名および FALCON における署名検証関数の比較

	Gentry らの格子ベース署名 [82, Sect. 7.1]	FALCON[80, Algorithm 16]
	$\mathsf{Vrfy}(m\in\{0,1\}^*,\sigma=s,pk=A)$	$\mathsf{Vrfy}(m\in\{0,1\}^*,\sigma=(r,\mathsf{s}),pk=h,\lfloor\beta^2\rfloor)$
1:	$\boldsymbol{t}\leftarrow H(m)$	$c\leftarrow\mathsf{HashToPoint}(r\|\|m,q,n)$
2:	**if** $\boldsymbol{t}-\boldsymbol{s}A\equiv 0 \pmod q$	$s_2\leftarrow\mathsf{Decompress}(\mathsf{s},8\cdot\mathsf{sbytelen}-328)$
	AND $\boldsymbol{s}$ が短い **then return** accept	**if** $(s_2=\perp)$
		return reject
		$s_1\leftarrow c-s_2h \bmod q$
		if $\|(s_1,s_2)\|^2\le\lfloor\beta^2\rfloor$
		return accept
		else
		return reject

表 3.14 の署名検証関数の説明を記述する。平文，ハッシュ関数のシード値，署名文字列から各要素を復元し，$s_1=c-s_2h$ を計算する。署名が正しく生成されていれば $\boldsymbol{s}A=c$ の関係から，s_1 は短い元となるはずなので，$\|(s_1,s_2)\|^2\le\lfloor\beta^2\rfloor$ が満たされ検証が完了する。

安全性とパラメータ: FALCON の安全性は $\phi(x)=x^n+1, q=12289$ を定義多項式とする NTRU 格子上の計算問題として表現される。鍵復元の困難性は SIS 問題，署名偽造はターゲットベクトルに近い点を求める計算問題として定式化される。後者は Kannan の埋め込みにより短いベクトルを求める計算問題に変換される。セキュリティに関わるパラメータは n,q,σ,β の 4 個で，n は格子の次元を表し，大きく取ることで安全性が上がるが処理速度が低下する。q は環を定義するための法で，大きくとることでノイズ耐性が上がるが格子が疎になり安全性が低下する。σ は Gauss 分布の大きさを指定するパラメータで，大きくとることで安全性が上がるがエラー率が上がる。β は署名ベクトルの長さの上限を指定するパラメータで，大きくとることで署名生成時のやり直し回数が下がるが，安全性が低下する。

具体的な困難性の評価およびパラメータ設定は，SIS 問題を BKZ アルゴリズムを用いて解いた場合の Core-SVP 計算量により導出している。

変種: 実装の複雑さによるサイドチャネル攻撃からの防御，セキュリティパラメータの多様性確保などを目的とした改良が多数提案されている。

特に，鍵生成と署名生成における離散 Gauss 分布生成の改良が多い。一例として，Gauss 分布生成の演算を浮動小数点から整数演算に変更した Zalcon[78]，Gauss 分布の代わりに中心二項分布とした Peregrine [143]，実装が複雑な

[*11] 秘密鍵サイズは仕様書には掲載されていないが，NIST の第 3 ラウンド報告レポート [5, Sect. D] を参照した。

表 3.15: FALCON のパラメータ [80, Table 3.3], [5, Table 8] 公開鍵，秘密鍵，署名サイズの単位はそれぞれ Byte である。

$(n, q, \sigma, \lfloor \beta^2 \rfloor)$	安全性レベル	公開鍵サイズ	秘密鍵サイズ [*11]	署名サイズ
(512, 12289, 165.736617183, 34034726)	レベル 1	897	7,553	666
(1024, 12289, 168.388571447, 70265242)	レベル 5	1,793	13,953	1,280

高速フーリエサンプリングを環上の CVP アルゴリズムをベースとしたより単純なものに置き換えた Mitaka [71] などが存在する。Peregrine は韓国の耐量子計算機暗号公募 KpqC[158] へと提出されているものの，同じ秘密鍵から生成した署名に対する統計的攻撃法による実時間での鍵復元手法が知られている [99]。

また，Falcon では環の定義多項式が $\phi(x) = x^n + 1, n = 2^k$ の形に制限されていることから安全性レベル 1,5 のパラメータのみが提案されていたが，NTRU 格子をモジュール格子とすることでパラメータ設定の多様性を確保した Mod Falcon [46] も存在する。

上記 Mitaka 内で用いられる離散 Gauss 分布生成アルゴリズムは実装が比較的単純である半面，生成された鍵および署名ベクトルのノルムが大きく鍵長と署名長が長いという欠点があった。近年では Antrag[117] が両者の中間的な手法として，FFT 表現でのサンプリングを通じて鍵生成における離散 Gauss 分布のノルムを下げ鍵長と署名長を短くする戦略を取っている。また，SOLMAE[94] も同様のサンプリング手法を用いた上で，エラーベクトルの圧縮表現などを用いて署名長を短縮する技術 [73] と組み合わせ KpqC へと提案されている。

補足情報: 2022 年に NIST より標準化がアナウンスされ，将来的に NIST FIPS 206 (FN-DSA) として出版される予定であるが，他の格子暗号方式 (FIPS 203 および 204) と比較して発表が遅れている。これは基準となる仕様書版 [80] からの修正箇所 [129] が多いことが原因であると考えられる。

鍵生成および署名生成アルゴリズムの中で浮動小数点演算が用いられているため実行環境ごとの結果の不安定性，定数時間での実装が難しいことによるタイミング攻撃の可能性がある。対策として固定小数点を用いた実装への変更が検討されている [129, p. 13]。

また，ML-DSA と比較して beyond unforgeability [51] の性質を完全には持たないことから，署名生成におけるハッシュ値の計算方法の変更が検討されている（[129, p. 15] および [70] を参照）。

3.3.6 FrodoKEM

歴史: 2016 年の国際会議 CCS において LWE ベースの鍵共有プロトコル Frodo が Joppe Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, Douglas Stebila の 8 名の連名で公表された [32]。

2017 年 11 月の NIST PQC 標準化プロジェクトの公募に提出された LWE ベースの公開鍵暗号方式 FrodoPKE および鍵カプセル化メカニズム Frodo KEM[115] では [32] の著者 8 名から Craig Costello が抜け，Erdem Alkim, Patrick Longa, Christopher Peikert の 3 名を加えた 10 名が inventors，Karen Easterbrook, Brian LaMacchia を Additional submitters とした合計 12 名での提案となっている。

NIST PQC 標準化プロジェクトへの提出後のディスカッションを通じて修正が加えられた後に ISO 標準化に提案されている。現在の最新版は 2024 年 12 月に公表された [81] である。本節の記述は NIST PQC 標準化プロジェクトを通じてアップデートされた 2021 年 6 月の第 3 ラウンド版 [14]，および 2024 年 12 月に公表された ISO 標準への提出版 [81] に従う。

参照 URL: 開発者による公式ページ `https://frodokem.org/` およびリファレンスコード `https://github.com/Microsoft/PQCrypto-LWEKE` を参照した。

設計原理: FrodoKEM は LWE 問題を安全性の根拠とする公開鍵暗号方式であり，将来 Ring 型や Module 型の構造を持つ格子問題を用いた暗号に致命的な脆弱性が発見された場合でも安全性が確保されると期待されることを特徴としている[*12]。LWE 問題自体の単純さからパラメータ設定の小回りが利くことも長所の一つとしている。形式的には Gentry-Peikert-Vaikuntanathan の [82, Sect. 7.1]，Lindner-Peikert [100] をひな型とする dual-LWE 暗号に分類される。IND-CPA 安全な公開鍵暗号方式を構成した後に，Hofheinz ら [88] のモジュール化藤崎-岡本変換 $\mathsf{FO}^{\not\perp}$ に Bos らの [33] の修正を施した変換手法 [14, Def. 2.19] を適用し IND-CCA2 安全な KEM を構成している。

3.3.6.1 NIST PQC 第 3 ラウンド版

アルゴリズムの詳細: 表 3.16，3.17，3.18 に Lindner-Peikert[100] による格子ベース公開鍵暗号と FrodoKEM の構成の基礎となる FrodoPKE の鍵生成，暗号化，復号アルゴリズムを並置する。

パブリックパラメータは以下で与えられる。

- q: 計算の剰余環 $\mathbb{Z}_q$ を指定する。ここでは $q = 2^D$ の形とし，$D = 15, 16$ に固定される。
- $n, \overline{m}, \overline{n}$: 行列のサイズを指定する。$n$ は 8 の倍数とする。また，平文は $\overline{m} \times \overline{n} = 8 \times 8$ 行列に符号化される。
- B, ℓ: 平文行列に符号化する情報量を指定する。行列の各成分は $0, \ldots, 2^B - 1$ の整数で表現され，合計で $\ell = B \cdot \overline{m} \cdot \overline{n}$ ビットの情報を埋め込むことができる。
- $\mathsf{len}_{\mathsf{seed}_A}, \mathsf{len}_{\mathsf{seed}_{SE}}$: 擬似ランダム行列 A, S, E を生成するためのシードとなるビット列の長さで，提案パラメータセットでは $\mathsf{len}_{\mathsf{seed}_A}$ は 128 に固定され，$\mathsf{len}_{\mathsf{seed}_{SE}}$ はセキュリティレベルに合わせて $128, 192, 256$ の値を取る。
- T_χ: SampleMatrix 関数で用いられる確率分布の表で，セキュリティレベルごとに離散 Gauss 分布からの Rényi divergence が小さくなるように設計されている。具体的な値は [14, Table 3] を参照。

関数内で用いられるサブルーチン群を以下に記述する。

- Frodo.Gen 関数はシードとなるビット列 seed と SHAKE ハッシュ関数を用いて擬似ランダム行列 $A \in \mathbb{Z}_q^{n \times n}$ を生成する関数である。i 行目を生成する際に整数 i を 16 ビットのビット列にエンコードした $\langle i \rangle$ を用いて $\mathsf{SHAKE}(\langle i \rangle || \mathsf{seed}, 16n)$ を呼び出し，得られた $16n$ ビットを 16 ビットごとに分割することで n 個の整数 $c_0, \ldots, c_{n-1} \in \{0, \ldots, 2^{16} - 1\}$ として，$A_{i,j} = c_j \mod q$ の形で各成分に振り分けている。このように関数を構成することで，各 i 行目を生成する操作がハードウェアによる並列実装に適した形となる。また，q は 2 べきの形で取られるため，各 $A_{i,j}$ の分布に偏りは生じない。
- Frodo.SampleMatrix$((\boldsymbol{r}^{(0)}, \ldots, \boldsymbol{r}^{(nm-1)}), n, m, T)$ 関数は行列のサイズ $n \times m$ と (i, j) 成分の生成に用いるシード $\boldsymbol{r}^{(in+j)}$ の列，乱数の確率分布を示すテーブル T を入力とする。それぞれのシードの長さは 16 ビットに固定されている。T は整数上の中心対称な確率分布が $\Pr[|T| = t]$ のテーブルとして与えられており，シード $\boldsymbol{r}^{(in+j)}$ の先頭 15 ビットで (i, j) 成分の絶対値を，残りの 1 ビットで符号を決定しサンプリングを行う。
- Frodo.Encode, Frodo.Decode 関数は $\ell = B \cdot \overline{m} \cdot \overline{n}$ ビットの平文を $\overline{m} \times \overline{n}$ 行列に埋め込む関数とその逆演算を行う関数である。B ビットの整数 k を $\mod q$ に埋め込むため，行列の成分を $k \cdot \lfloor q/2^B \rfloor$ とする。

FrodoKEM の鍵生成関数（表 3.16 右）を説明する。鍵生成のためのシード seed_A と seed_{SE} を生成した後, Frodo.Gen 関数と Frodo.SampleMatrix 関数を用いて A, S, E を生成する。このとき，行列乗算時のメモリアクセスの順序を考えて S は転置の形で格納される。公開鍵行列 A は鍵サイズ圧縮のために成分ではなくシード seed_A の形で格納される。

[*12] FrodoKEM の仕様書 [14] では構造を持たないことを “algebraically unstructured” と表現している。

表 3.16: Lindner-Peikert 格子ベース暗号および FrodoKEM における鍵生成関数の比較

	Lindner-Peikert[100, Sect. 3.1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	FrodoKEM [14, Algorithm 9] $\mathsf{FrodoPKE.KeyGen}(1^\lambda) \to (pk, sk)$
1:	A: $n_1 \times n_2$ ランダム行列	$\mathsf{seed}_A \xleftarrow{\$} \{0,1\}^{\mathsf{len}_{\mathsf{seed}_A}}$
		$A \leftarrow \mathsf{Frodo.Gen}(\mathsf{seed}_A)$
2:	S: 成分の小さい $n_2 \times \ell$ 行列	$\mathsf{seed}_{SE} \xleftarrow{\$} \{0,1\}^{\mathsf{len}_{\mathsf{seed}_{SE}}}$
		//擬似乱数ビットの生成
		$(\boldsymbol{r}^{(0)}, \ldots, \boldsymbol{r}^{(2n\overline{n}-1)}) \leftarrow \mathsf{SHAKE}(\mathsf{0x5F} \| \mathsf{seed}_{SE}, 2n\overline{n} \cdot \mathsf{len}_\chi)$
		$S^T \leftarrow \mathsf{Frodo.SampleMatrix}((\boldsymbol{r}^{(0)}, \ldots, \boldsymbol{r}^{(n\overline{n}-1)}), \overline{n}, n, T_\chi)$
3:	E: 成分の小さい $n_1 \times \ell$ 行列	$E \leftarrow \mathsf{Frodo.SampleMatrix}((\boldsymbol{r}^{(n\overline{n})}, \ldots, \boldsymbol{r}^{(2n\overline{n}-1)}), \overline{n}, n, T_\chi)$
4:	$B = AS + E$	$B = AS + E$
return	$pk = (A, B), sk = S$	$pk = (\mathsf{seed}_A, B), sk = S^T$

表 3.17: Lindner-Peikert 格子ベース暗号および FrodoKEM における暗号化関数の比較

	Lindner-Peikert[100, Sect. 3.1] $\mathsf{Enc}(pk = (A, B),$ $\boldsymbol{m} \in \{0,1\}^\ell) \to \mathsf{ct}$	FrodoKEM [14, Algorithm 10] $\mathsf{FrodoPKE.Enc}(pk = (A, B), \mu \in \{0,1\}^\ell)) \to \mathsf{ct}$
0:		$A \leftarrow \mathsf{Frodo.Gen}(\mathsf{seed}_A)$ // A の復元
1:	$\boldsymbol{s}', \boldsymbol{e}', \boldsymbol{e}''$: 成分の小さいベクトル	$\mathsf{seed}_{SE} \xleftarrow{\$} \{0,1\}^{\mathsf{len}_{\mathsf{seed}_{SE}}}$
		$(\boldsymbol{r}^{(0)}, \ldots, \boldsymbol{r}^{(2n\overline{n}-1)}) \leftarrow \mathsf{SHAKE}(\mathsf{0x96} \| \mathsf{seed}_{SE},$
		$(2\overline{m} \cdot n + \overline{m} \cdot \overline{n}) \cdot \mathsf{len}_\chi)$
		//擬似乱数ビットの生成
		$S' \leftarrow \mathsf{Frodo.SampleMatrix}((\boldsymbol{r}^{(0)}, \ldots, \boldsymbol{r}^{(\overline{m} \cdot n-1)}), \overline{m}, n, T_\chi)$
		$E' \leftarrow \mathsf{Frodo.SampleMatrix}((\boldsymbol{r}^{(\overline{m} \cdot n)}, \ldots, \boldsymbol{r}^{(2\overline{m} \cdot n-1)}), \overline{m}, n, T_\chi)$
		$E'' \leftarrow \mathsf{Frodo.SampleMatrix}((\boldsymbol{r}^{(2\overline{m} \cdot n)}, \ldots, \boldsymbol{r}^{(2\overline{m} \cdot n+\overline{m} \cdot \overline{n}-1)}), \overline{m}, \overline{n}, T_\chi)$
2:	$\boldsymbol{u} = \boldsymbol{s}'A + \boldsymbol{e}'$	$B' = S'A + E'$; $V = S'B + E''$
	$\boldsymbol{v} = \boldsymbol{s}'B + \boldsymbol{e}'' + \boldsymbol{m} \cdot \lfloor \frac{q}{2} \rfloor$	$C_1 = B'$; $C_2 = S'B + E'' + \mathsf{Frodo.Encode}(\mu)$
return	$\mathsf{ct} = (\boldsymbol{u}, \boldsymbol{v})$	$\mathsf{ct} = (C_1, C_2)$

FrodoKEM の暗号化関数（表 3.17 右）を説明する。seed_A から行列 A を復元した後，暗号化用の乱数行列 S', E', E'' を擬似乱数列 $\boldsymbol{r}^{(i)}$ から生成する。擬似乱数列の生成には SHAKE ハッシュ関数を用いるが，パディング値 0x96 が鍵生成で用いられた 0x5F と異なるため鍵生成の S, E とは異なる行列が得られることに注意。残りの処理はひな型の Lindner-Peikert 暗号を行列化したものである。

FrodoKEM の復号関数（表 3.18 右）は Lindner-Peikert 暗号の復号処理を行列化したものである。

安全性とパラメータ: ベースとなる IND-CPA 安全な公開鍵暗号の安全性は判定版 LWE 問題に帰着される。実装上の効率化のため，鍵生成，暗号化処理において離散 Gauss 分布を近似した確率分布 T_χ を用いているが，その際の安全性の低下は Rényi divergence を用いた議論により評価されている [14, Sect. 5.1]。n は格子の次元で，大きくとることで安全性レベルが上がるが処理コストも上がる。q は環を定義する法で，大きく取ることで平文空間も大きくなるが，格子が疎になり安全性が下がる。σ は離散 Gauss 分布の大きさを決定するパラメータで，大きくとることで安全性が上がるが，復号エラー率が上がる。

表 3.18: Lindner-Peikert 格子ベース暗号および FrodoKEM における復号関数の比較

	Lindner-Peikert[100, Sect. 3.1] $\mathsf{Dec}(sk, \mathsf{ct}) \to \boldsymbol{m}'$	FrodoKEM [14, Algorithm 11] $\mathsf{FrodoPKE.Dec}(sk, \mathsf{ct}) \to \boldsymbol{m}'$
1:	$\overline{\boldsymbol{m}} = \boldsymbol{v} - \boldsymbol{u}S$ $m'_i = \begin{cases} 0 & \lvert m_i \rvert \leq \lfloor q/4 \rceil \\ 1 & \text{それ以外} \end{cases}$	$M = C_1 - C_2 S$ $\boldsymbol{m}' = \mathsf{Frodo.Decode}(M)$
return	$\boldsymbol{m}' = (m'_1, \ldots, m'_\ell)$	$\boldsymbol{m}'$

FrodoKEM のパラメータは LWE 問題の Primal 攻撃，Dual 攻撃双方での BKZ アルゴリズムを用いた計算量評価から求められている。保守的なパラメータ設定のため，Core-SVP, BKZ の計算量評価を，計算量の上界を示す既存の攻撃手法のみではなく，下界からの議論が行われていることも方式の特徴である。

表 3.19: FrodoKEM CCA のパラメータ [14, Table 5]. σ の値は T_χ の元となる離散 Gauss 分布の標準偏差を示す。秘密鍵サイズはデカプセル化時に用いられる鍵情報の中から，公開鍵に相当するものを除いた分である。公開鍵，秘密鍵，平文，暗号文サイズの単位はそれぞれ Byte である。

(n, q, σ)	安全性レベル	公開鍵サイズ	秘密鍵サイズ	平文サイズ	暗号文サイズ
$(\ 640, 2^{15}, 2.8)$	レベル 1	$9,616$	$10,272$	16	$9,720$
$(\ 976, 2^{16}, 2.3)$	レベル 3	$15,632$	$15,664$	24	$15,744$
$(1344, 2^{16}, 1.4)$	レベル 5	$21,520$	$21,568$	32	$21,632$

変種: 行列 A の生成に SHAKE-256 ではなく，AES-128 を使ったバージョンも提案されている。AES-NI 命令を用いた Intel CPU による実装では SHAKE を用いたものよりも 2.5 倍程度高速である [14, Sect. 3.2]。

また，Encode Decode 関数に誤り訂正符号を用いて暗号文サイズを 1 割ほど削減したバージョンが提案されている [139, 138]。

補足情報: 構造を持つ格子 (structured lattice) でないという理由から NIST PQC 標準化プロジェクトの 第 3 ラウンド候補として残っていたが，標準化から漏れた理由として，構造を持つ格子でないという性質を持つ符号ベース暗号の BIKE,HQC や同種写像ベース暗号の SIKE と比較すると，パフォーマンスの観点から不利であったと NIST の標準化レポート [5, p.17] に述べられている。

NIST PQC 標準化プロジェクトの 第 2 ラウンドのバージョンでは，藤崎-岡本変換を行った IND-CCA2 KEM の実装において再暗号化後のチェックが定数時間ではないことから鍵復元攻撃が可能であることが示され [85]，修正されている。実装にかかわる攻撃として，ロウハンマー (Rowhammer) 攻撃による鍵復元攻撃が新たに発見された [75]。

NIST PQC 標準化プロセスの中で第 3 ラウンド候補であり，標準化には至らなかったものの，フランス，ドイツ，オランダ，チェコ等の国で耐量子計算機暗号の推奨・許容リストに入っている[*13]。開発者らは ISO での標準化を目指し，2024 年に予備提案 (Preliminary Standardization Proposal) [17, 81] を提出している。次節に詳細を述べる。

3.3.6.2 ISO 標準への予備提案版

2024 年 12 月に公開された ISO への予備提案版 [17, 81] では，NIST PQC 版からセキュリティ強化のための修正が行われている。1 つ目は multi-target security と呼ばれる，複数の公開鍵が与えられたときに攻撃者がどれかひとつの

[*13] 各国の PQC 推奨・許容暗号リストの状況は第 1 章も参照。

鍵に関して IND-CCA 安全性を破ることすら困難であるという要件である。通常の IND-CCA 安全性では 1 つの公開鍵に対する識別を行うのに対し，multi-target では攻撃者が自身に都合の良い鍵を選ぶことができるため，攻撃者に有利な設定となる。

2 つ目は multi-ciphertext security と呼ばれる，1 つの公開鍵で暗号化された複数の暗号文が与えられたときに，攻撃者がどれか 1 つの暗号文に関する安全性を破ることが困難であるという要件である。これら 2 つの要件に対応するため，鍵カプセル化時に用いられる疑似乱数 seed_{SE} の長さに変更が加えられ，新たな乱数列 salt が導入されている。

表 3.20 から 3.22 に NIST 版 FrodoKEM と ISO 版 FrodoKEM の比較を示す。データの型変換を行う Pack, Unpack 関数の呼び出しは本質的でないため省略した。なお，2023 年，2024 年に公開された著者らの ISO 向け仕様書 [17, 81] では NIST 版（上記表 3.16 から 3.18 で示したもの）を eFrodoKEM，新たなバージョンを FrodoKEM としているため表記に注意。

表 3.20: FrodoKEM の NIST PQC 標準化プロジェクト 第 3 ラウンド版 [14, Algorithm 11] と ISO 版 [81, Sec. 8.1] の鍵生成関数の比較。両者ともに同じ関数であるが，表 3.23 に示すようにパラメータ $\mathsf{len}_{\mathsf{seed}_{SE}}$ の値が異なる。KeyGen 関数は表 3.16 の右側の関数内で乱数 $\mathsf{seed}_A, \mathsf{seed}_{SE}$ をランダムではなく，引数の値を用いて実行した結果を示す。

NIST 版 FrodoKEM[14, Algorithm 12] $\mathsf{KeyGen}(1^\lambda) \rightarrow (pk', sk')$	ISO 版 FrodoKEM [81, Sec. 8-1] $\mathsf{KeyGen}(1^\lambda) \rightarrow (pk', sk')$
乱数を生成 $s, \mathsf{seed}_{SE}, z \xleftarrow{\$} \{0,1\}^{\mathsf{len}_s + \mathsf{len}_{\mathsf{seed}_{SE}} + \mathsf{len}_z}$ $\mathsf{seed}_A \leftarrow \mathsf{SHAKE}(z, \mathsf{len}_{\mathsf{seed}_A})$ $(pk, sk) \leftarrow \mathsf{FrodoKEM.KeyGen}(\mathsf{seed}_A, \mathsf{seed}_{SE})$ $//pk = (\mathsf{seed}_\mathsf{A}, B)$ および $sk = S^T$ $pkh \leftarrow \mathsf{SHAKE}(\mathsf{seed}_A \|B, \mathsf{len}_{pkh})$	乱数を生成 $s, \mathsf{seed}_{SE}, z \xleftarrow{\$} \{0,1\}^{\mathsf{len}_s + \mathsf{len}_{seed_{SE}} + \mathsf{len}_z}$ $\mathsf{seed}_A \leftarrow \mathsf{SHAKE}(z, \mathsf{len}_{\mathsf{seed}_A})$ $(pk, sk) \leftarrow \mathsf{FrodoKEM.KeyGen}(\mathsf{seed}_A, \mathsf{seed}_{SE})$ $//pk = (\mathsf{seed}_\mathsf{A}, B)$ および $sk = S^T$ $pkh \leftarrow \mathsf{SHAKE}(\mathsf{seed}_A \|B, \mathsf{len}_{pkh})$
$pk' = (\mathsf{seed}_\mathsf{A}, B), sk' = (s, \mathsf{seed}_A, b, S^T, pkh)$	$pk' = (\mathsf{seed}_\mathsf{A}, B), sk' = (s, \mathsf{seed}_A, b, S^T, pkh)$

表 3.21: FrodoKEM の NIST PQC 標準化プロジェクト 第 3 ラウンド版 [14, Algorithm 12] と ISO 版 [81, Sec. 8.2] の鍵カプセル化関数の比較。$\mathsf{FrodoPKE.Enc}(\mathsf{seed}_{SE}, pk', \mu)$ 関数は表 3.17 右側の関数内で乱数 seed_{SE} をランダムではなく，引数の値を用いて実行した結果を示す。ISO 版では新たに salt が追加されている。

NIST 版 FrodoKEM[14, Algorithm 12] $\mathsf{Encaps}(pk') \rightarrow (\mathsf{ct}, ss)$	ISO 版 FrodoKEM [81, Sec. 8-1] $\mathsf{Encaps}(pk') \rightarrow (\mathsf{ct}, ss)$
$\mu \xleftarrow{\$} \{0,1\}^{\mathsf{len}_\mu}$ // ランダムな鍵を生成	$\mu \xleftarrow{\$} \{0,1\}^{\mathsf{len}_\mu}$ // ランダムな鍵を生成 $\mathsf{salt} \xleftarrow{\$} \{0,1\}^{\mathsf{len}_{\mathsf{salt}}}$ // ランダムな salt 値を生成
$pkh \leftarrow \mathsf{SHAKE}(pk, \mathsf{len}_{pkh})$ $(\mathsf{seed}_{SE}, k) \leftarrow \mathsf{SHAKE}(pkh, \mu, \mathsf{len}_{\mathsf{seed}_{SE}} + \mathsf{len}_k)$ $\mathsf{FrodoPKE.Enc}(\mathsf{seed}_{SE}, pk', \mu) \rightarrow (C_1, C_2)$ $\mathsf{SHAKE}(C_1 \|C_2\|k, \mathsf{len}_{ss}) \rightarrow ss$	$pkh \leftarrow \mathsf{SHAKE}(pk, \mathsf{len}_{pkh})$ $(\mathsf{seed}_{SE}, k) \leftarrow \mathsf{SHAKE}(pkh, \mu, \mathsf{salt}, \mathsf{len}_{\mathsf{seed}_{SE}} + \mathsf{len}_k)$ $\mathsf{FrodoPKE.Enc}(\mathsf{seed}_{SE}, pk', \mu) \rightarrow (C_1, C_2)$ $\mathsf{SHAKE}(C_1 \|C_2\|\mathsf{salt}\|k, \mathsf{len}_{ss}) \rightarrow ss$
$\mathsf{ct} = (C_1, C_2), ss$	$\mathsf{ct} = (C_1, C_2, \mathsf{salt}), ss$

表 3.23 に各安全性レベルごとの乱数 seed と seed_{SE} の長さをまとめる。$\mathsf{len}_{\mathsf{salt}} = 0$ の場合，salt は空となり NIST 第 3 ラウンド版と ISO 版は同じプロトコルとなる。seed を追加したことにより，暗号文長はレベル 1,3,5 の場合表 3.19 と比較して 32,48,64Byte 長くなる。

表 3.22: FrodoKEM の NIST PQC 標準化プロジェクト 第 3 ラウンド版 [14, Algorithm 13] と ISO 版 [81, Sec. 8.3] のデカプセル化関数の比較。$\mathsf{FrodoPKE.Enc}(\mathsf{seed}'_{SE}, pk', \mu')$ 関数は表 3.17 右側の関数内で乱数 seed_{SE} をランダムではなく，引数の値を用いて実行した結果を示す。ISO 版では新たに salt が追加されている。

NIST 版 FrodoKEM[14, Algorithm 14]	ISO 版 FrodoKEM [81, Sec. 8-2]
$\mathsf{Decaps}(sk', \mathsf{ct}) \to ss$	$\mathsf{Decaps}(sk', \mathsf{ct}) \to ss$
$\mu' \leftarrow \mathsf{FrodoPKE.Dec}(sk = S^T, \mathsf{ct})$	$\mu' \leftarrow \mathsf{FrodoPKE.Dec}(sk = S^T, \mathsf{ct})$
$(\mathsf{seed}'_{SE}, k') \leftarrow \mathsf{SHAKE}(pkh, \mu, \mathsf{len}_{\mathsf{seed}_{SE}} + \mathsf{len}_k)$	$(\mathsf{seed}'_{SE}, k') \leftarrow \mathsf{SHAKE}(pkh, \mu', \mathsf{salt}, \mathsf{len}_{\mathsf{seed}_{SE}} + \mathsf{len}_k)$
$\mathsf{FrodoPKE.Enc}(\mathsf{seed}'_{SE}, pk', \mu')) \to (C'_1, C'_2) =: \mathsf{ct}'$	$\mathsf{FrodoPKE.Enc}(\mathsf{seed}'_{SE}, pk', \mu)) \to (C'_1, C'_2) =: \mathsf{ct}'$
if $(\mathsf{ct} = \mathsf{ct}')$ $\bar{k} \leftarrow k'$ **else** $\bar{k} \leftarrow s$	**if** $(\mathsf{ct} = \mathsf{ct}')$ $\bar{k} \leftarrow k'$ **else** $\bar{k} \leftarrow s$
$\mathsf{SHAKE}(C_1 \| C_2 \| \bar{k}, \mathsf{len}_{ss}) \to ss$	$\mathsf{SHAKE}(C_1 \| C_2 \| \mathsf{salt} \| \bar{k}, \mathsf{len}_{ss}) \to ss$

表 3.23: NIST 版と ISO 版におけるパラメータ $\mathsf{len}_{\mathsf{salt}}$ と $\mathsf{len}_{\mathsf{seed}_{SE}}$ の違い。セキュリティ強化のため salt が追加され，seed_{SE} の長さが変更されている。

	eFrodoKEM (NIST 第 3 ラウンド版)		FrodoKEM (ISO 版)	
安全性レベル	$\mathsf{len}_{\mathsf{salt}}$	$\mathsf{len}_{\mathsf{seed}_{SE}}$	$\mathsf{len}_{\mathsf{salt}}$	$\mathsf{len}_{\mathsf{seed}_{SE}}$
レベル 1	0	128	256	256
レベル 3	0	192	384	384
レベル 5	0	256	512	512

3.3.7 NewHope

歴史: NewHope の最初のバージョンは 2016 年に国際会議 USENIX Security において Erdem Alkim，Léo Ducas，Thomas Pöppelmann，Peter Schwabe により鍵共有プロトコルとして発表された [13]。また，直後に reconciliation によるエラー訂正プロセスを省略し簡略化した NewHope-Simple[12] が ePrint Archive において発表されている。

2017 年 11 月の NIST PQC 標準化プロジェクトの公募に応募された Version 1.0[125] は新たに Roberto Avanzi，Joppe Bos，Antonio de la Piedra，Douglas Stebila の 4 人が加わった合計 8 人での提案とし，[12] をベースとして公開鍵暗号方式を構成している。NIST PQC 標準化プロジェクトへ提出後のディスカッションを通じて修正が加えられ，現在の最新版は 2020 年 4 月に公表された Version 1.1[15] である。

NIST PQC 標準化プロジェクトの 第 2 ラウンドに提出された Version 1.02[126] では，Martin R. Albrecht，Emmanuela Orsini，Valery Osheter，Kenneth G. Paterson，Guy Peer，Nigel P，Smart の 6 人が Contributor として列挙されている。

本節の記述は最新版の仕様書 [15] に従う。

参照 URL: 開発者による公式ページ `https://newhopecrypto.org/` および GitHub 上のリファレンス実装 `https://github.com/newhopecrypto/newhope` を参照した。

設計原理: NewHope は環 $\mathbb{Z}[x]/(x^n + 1), n = 2^k$ 上の Ring-LWE 問題の計算困難性を安全性の根拠とする公開鍵暗号方式であり，形式的には Gentry-Peikert-Vaikuntanathan の [82, Sect. 7.1]，Lindner-Peikert [100] をひな型とする dual-LWE 暗号に分類される。ベースとなる暗号方式におけるベクトルと行列の演算を多項式環の要素に置き換え IND-CPA 安全な公開鍵暗号方式を提案している。その際，数論変換を用いた乗算などの実装テクニックを用いて処理

を高速化している。環の定義多項式を $x^n+1, n=2^k$ の形としている点も高速化に寄与しているが，その一方でパラメータ選択の自由度に制限があり，NIST PQC 標準化プロジェクトの提案方式では安全性レベル 1 およびレベル 5 のパラメータセットのみが提案されている。

IND-CPA 安全な公開鍵暗号から IND-CCA 安全な KEM の構成には [88] のモジュール化された藤崎-岡本変換 $\mathsf{QFO}_m^{\not\perp}$ を用いているが，その際に公開鍵暗号 CRYSTALS-Kyber（本報告書の 3.3.3 節も参照）の IND-CCA 安全な KEM の構成 [33, Sect. 4] に用いられた手法を取り入れ微修正を施している。結果として，構成された KEM が ROM, QROM の双方のモデルにおいて CCA 安全であることが保証されている。

アルゴリズムの詳細: 表 3.24，3.25，3.26 に Lindner-Peikert[100] による格子ベース公開鍵暗号と NewHope の鍵生成，暗号化，復号アルゴリズムを並置する。

パブリックパラメータは以下で与えられる。

- n, q: 演算を行う環 $R_q := \mathbb{Z}_q[x]/(x^n+1)$ を定義する。特に指定のない場合には多項式の係数は自動的に区間 $[-q/2, q/2)$ 内に収められるものとする。高速数論変換のため，n は 2 のべき乗の形であり，さらにアルゴリズム中で用いられる ω, γ が存在するために q は $q \equiv 1 \pmod{2n}$ を満たす素数として選ばれる。NewHope のパラメータ設定では $n = 512, 1024, q = 12289$ が選ばれている。
- k: ノイズの大きさを設定する。
- ω, γ: 数論変換で用いる。$\mathbb{Z}_q^\times$ における 1 の原始 n 乗根を ω, その平方根を $\gamma := \sqrt{\omega} \mod q$ とする。NewHope のパラメータ設定では $n = 512$ に対して $(\omega, \gamma) = (3, 10968)$，$n = 1024$ に対して $(\omega, \gamma) = (49, 7)$ が取られている。

関数内で用いられるサブルーチン群を以下に記述する。

- $\mathsf{Sample}(\mathsf{seed}, \mathsf{nonce})$ 関数は，各係数を平均ゼロに調整した二項分布 ψ_8 から独立にサンプリングした n 次多項式を 32Bytes の seed と非負整数値 nonce から生成する。自然数 k に対して，ψ_k の出力は独立にサンプリングした $2k$ 個のビット $b_i, b_i' \xleftarrow{\$} \{0,1\}\ (k = 1, \ldots, k)$ に対する $\sum_{i=1}^{k}(b_i - b_i')$ として定義される。
- $\mathsf{PolyBitRev}(a \in R_q)$: 高速数論変換を用いた乗算の場合，結果のインデックス順序が入れ替わるため配列の要素 $c[i]$ を $x^{\mathsf{BitRev}}(i)$ として解釈しなければならない。ただし，ビット順序反転は $h = \log_2(n), i = \sum_{j=0}^{h-1} b_j 2^j$ と 2 進数展開したときに，$\mathsf{BitRev}(i) := \sum_{j=0}^{h-1} b_j 2^{h-j-1}$ 計算される。関数は多項式 $a(x) = \sum_{i=0}^{n-1} a_i x^i$ に対して，指数部分をビット順序反転した多項式 $\sum_{i=0}^{n-1} a_i x^{\mathsf{BitRev}(i)}$ を出力する。
- $\mathsf{NTT}(a), \mathsf{NTT}^{-1}(\hat{a})$: R_q の多項式同士の乗算を高速化するため，数論変換 (NTT: Number Theoretic Transform)[15, p. 7-9] を用い，鍵と暗号文の処理を極力 NTT 空間で行う工夫がなされている。パラメータ n, q, ω, γ を固定したとき，多項式 $a(x) = \sum_{i=0}^{n-1} a_i x^i \in R_q$ の数論変換

$$\hat{a} = \mathsf{NTT}(a) := \sum_{i=0}^{n-1} \hat{a}_i x^i,\ \hat{a}_i := \sum_{j=0}^{n-1} \gamma^j a_j \omega^{ij} \pmod{q}$$

および逆変換を

$$a = \mathsf{NTT}^{-1}(\hat{a}) := \sum_{i=0}^{n-1} a_i x^i,\ a_i := \left(n^{-1}\gamma^{-i} \sum_{j=0}^{n-1} \hat{a}_j \omega^{-ij} \right) \pmod{q}$$

とする。これらは線形変換であるので，$a, b \in R_q$ に対して $\mathsf{NTT}(a) + \mathsf{NTT}(b) = \mathsf{NTT}(a+b)$ 等の性質がなりたつ。また，$a * b := \sum_{i=0}^{n-1} c_i x^i, c_i = \sum_{j=0}^{n-1} a_j b_{i-j \mod n} \mod q$ は $a * b = \mathsf{NTT}^{-1}(\mathsf{NTT}(a) \circ \mathsf{NTT}(b))$ を満たす。ただし，記号 $\circ$ は多項式の係数同士の積を取ることを表す。

$a * b$ を定義式通りに計算すると mod q での演算が $O(n^2)$ 回必要であるのに対して，高速数論変換を用いた方法では $O(n\log n)$ 回の演算で可能である。

数論変換を用いた高速乗算ではどこかのタイミングで添え字のビット順序を反転する必要がある。単純な IND-CPA PKE の実装における最適なのみを考えるのであればこのような置換は必要ないが，IND-CCA KEM のデカプセル化処理内での再暗号化まで含めて実装を最適化した結果，NewHope では最初の KeyGen, Enc 関数の中で置換が行われている [15, p.8]。

なお，仕様書 [15] には多項式とバイト列の相互変換を行う関数 EncodePK, EncodePolynomial, EncodeC, Compress, DecodePK, DecodePolynomial, DecodeC, Decompress が定義されているが，何れもデータの再配置を行う関数であり方式を説明する上では本質的ではないため省略した。

表 3.24: Lindner-Peikert 格子ベース暗号および NewHope における鍵生成関数の比較

	Lindner-Peikert[100, Sect. 3.1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	NewHope[15, Algorithm 1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
1:	A: $n_1 \times n_2$ ランダム行列	$\mathsf{seed} \xleftarrow{\$} \{0,1,\ldots,255\}^{32}, z = \mathsf{SHAKE256}(64, \mathsf{seed})$ //32Bytes の seed を 64Bytes に伸長 $\hat{a} = \mathsf{GenA}(\mathsf{z}[0:31]) \in R_q$
2:	S: 成分の小さい $n_2 \times \ell$ 行列	$s = \mathsf{PolyBitRev}(\mathsf{Sample}(\mathsf{z}[32:63], 0)) \in R_d$; $\hat{s} = \mathsf{NTT}(s)$
3:	E: 成分の小さい $n_1 \times \ell$ 行列	$e = \mathsf{PolyBitRev}(\mathsf{Sample}(\mathsf{z}[32:63], 1)) \in R_d$; $\hat{e} = \mathsf{NTT}(e)$
4:	$B = E - AS$	$\hat{b} = \hat{a} \circ \hat{s} + \hat{e}$ //$b = \mathsf{NTT}(a * s + e)$
return	$pk = (A, B), sk = S$	$pk = (\hat{a}, \hat{b}), sk = \hat{s}$

NewHope の鍵生成関数（表 3.24 右）を説明する。ステップ 1 では 32Bytes(= 256bits) の乱数のシードを SHAKE-256 ハッシュ関数を用いて 64Bytes の擬似乱数列 z を生成し，それを前半の $\mathsf{z}[0:31]$ と後半の $\mathsf{z}[32:63]$ に分割する。Lindner-Peikert 型暗号（左側）におけるランダム行列 A の生成に対応して，$\mathsf{GenA}(\cdot)$ 関数は 32Bytes の列をシードとして，ランダムな R_q の元を生成する。各係数が一様独立に Z_q の元からサンプリングされる。実装は [15, Algorithm 5] を参照。出力された $\hat{a}$ がランダムな多項式 a の数論変換であることは，ランダム多項式の数論変換がまたランダム多項式となることから従う。

ステップ 2,3 ではそれぞれ係数の小さい多項式 s, e の数論変換を計算する。ステップ 1 で生成した擬似乱数の後半 $z[32:63]$ をシードに用いて小さい値を係数に持つ多項式のサンプリングを行い，数論変換のためのビット順序の反転処理をしたものを s，その数論変換を $\hat{s}$ とする。$e, \hat{e}$ についても同様。

ステップ 4 では数論変換後の多項式から公開鍵 $\hat{b}$ を計算する。数論変換の性質より，これは $a * b + e$ の NTT 表現となる。

NewHope の暗号化関数（表 3.25 右）を説明する。暗号化のためのランダム多項式 $\hat{t}, e', e''$ を生成するため，鍵生成のステップ 2-3 と同様の処理を行う。$\hat{t}$ は数論変換後の形式であるが，e', e'' は通常の形式のまま用いる。

表 3.25: Lindner-Peikert 格子ベース暗号および NewHope における暗号化関数の比較

	Lindner-Peikert[100, Sect. 3.1] $\mathsf{Enc}(pk=(A,B), \boldsymbol{m}\in\{0,1\}^{\ell}) \to \mathsf{ct}$	NewHope[15, Algorithm 1] $\mathsf{Enc}(pk=(A,B), M\in\{0,1,\ldots,255\}^{32}) \to \mathsf{ct}$
1:	$\boldsymbol{t}, \boldsymbol{e}', \boldsymbol{e}''$: 成分の小さいベクトル	$\mathsf{coin} \xleftarrow{\$} \{0,1,\ldots,255\}^{32}$ // ランダムシード $s' = \mathsf{PolyBitRev}(\mathsf{Sample}(\mathsf{coin},0)) \in R_d$; $\hat{t} = \mathsf{NTT}(s')$ $e' = \mathsf{PolyBitRev}(\mathsf{Sample}(\mathsf{coin},1)) \in R_d$ $e'' = \mathsf{Sample}(\mathsf{coin},2) \in R_d$
2:	$\boldsymbol{u} = \boldsymbol{t}A + \boldsymbol{e}'$ $\boldsymbol{v} = \boldsymbol{t}B + \boldsymbol{e}'' + \boldsymbol{m}\cdot\left\lfloor\frac{q}{2}\right\rfloor$	$\hat{u} = \hat{a}\circ\hat{t} + \mathsf{NTT}(e')$ $v' = \mathsf{NTT}^{-1}(\hat{b}\circ\hat{t}) + e'' + \mathsf{Encode}(M)$
return	$\mathsf{ct} = (\boldsymbol{u}, \boldsymbol{v})$	$\mathsf{ct} = (\hat{u}, v')$

Encode 関数は 32Bytes(=256bits) の平文を n 次多項式の係数として埋め込む。NewHope のパラメータでは $n = 512, 1024$ が用いられるため，1bit の情報が複数箇所に埋め込まれることになる。具体的には平文の $i$bit 目を b_i，多項式の j 次の係数を v_j としたときに $j = 0, \ldots, n-1$ に対して

$$v_j = \begin{cases} 0 & (b_{j \bmod 256} = 0) \\ \left\lfloor\frac{q}{2}\right\rfloor & (b_{j \bmod 256} = 1) \end{cases}$$

とする。

表 3.26: Lindner-Peikert 格子ベース暗号および NewHope における復号関数の比較

	Lindner-Peikert[100, Sect. 3.1] $\mathsf{Dec}(sk, \mathsf{ct}) \to m'$	NewHope[15, Algorithm 1] $\mathsf{Dec}(sk, \mathsf{ct}) \to M'$
1:	$\overline{\boldsymbol{m}} = \boldsymbol{v} + \boldsymbol{u}S$ $m'_i = \begin{cases} 0 & \lvert m_i\rvert \le \lfloor q/4\rfloor \\ 1 & \text{それ以外} \end{cases}$	$\overline{m} = v - \mathsf{NTT}^{-1}(\hat{u}\circ\hat{s})$ $M' = \mathsf{Decode}(\overline{m})$
return	$\boldsymbol{m}' = (m'_1, \ldots, m'_\ell)$	M'

NewHope の復号関数(表 3.33 右)を説明する。健全性の証明より，$v - \mathsf{NTT}^{-1}(\hat{u}\circ\hat{s}) = v - u*s$ が $\mathsf{Encode}(M)$ と小さいノイズ和であることが示されるため，Decode 関数はノイズの除去と平文 M' の復元を同時に行う [15, Algorithm 11]。多項式 $\overline{m}$ の係数の中で，$\overline{m}_{i+256k}$ $(k = 0, \ldots, n/256-1)$ の中にビット M'_i の情報が埋め込まれているため，それらを多数決で決定する。具体的には，$\sum_{k=0}^{n/256-1} |\overline{m}_{i+256k} - (q-1)/2|$ が $M'_i = 0$ の場合には $(n/256)\cdot(q-1)/2 \approx (n/512)q$ に近く，$M'_i = 1$ の場合には 0 に近い値を取るため，和から $(n/1024)q$ を引いた後に符号を見ることでビット列の復元が完了する。

安全性とパラメータ: ベースとなる IND-CPA 安全な公開鍵暗号の安全性は環 $Z_q[x]/(x^n+1)$ 上の判定版 LWE 問題に量子帰着されることが示されている。実装上の効率化のため，NewHope では鍵生成，暗号化の際に離散 Gauss 分布の替わりに中心を 0 とした二項分布を用いているが，その分の安全性の低下は [15, Theorem 4.1] で Rényi ダイバージェンスを用いた議論により評価されている。

Ring-LWE 問題の具体的な困難性の評価には，LWE 問題に対する Primal 攻撃，Dual 攻撃双方での，BKZ アルゴリズムを用いた必要ブロックサイズから導き出した CoreSVP 計算量による評価を用いている。

表 3.27: NewHope CPA-KEM, CCA-KEM のパラメータ [15, Table 2, 3]. 2 番目のパラメータは NIST 耐量子計算機暗号 Call for proposal 基準でレベル 5 であると主張されているが，表 [15, Table 3] では 233-bit 安全性となっている。公開鍵，秘密鍵，平文，暗号文サイズの単位はそれぞれ Byte である。

(n, q, k, γ)	安全性 レベル	公開鍵 サイズ	秘密鍵サイズ (鍵カプセル化後)	平文 サイズ	暗号文サイズ (鍵カプセル化後)
(512, 12289, 8, 10968)	レベル 1	928	869 (1,888)	32	1,088 (1,120)
(1024, 12289, 8, 7)	レベル 5	1,824	1,792 (3,680)	32	2,176 (2,208)

変種: 鍵共有を目的とした USENIX 版 [13]，および reconciliation によるエラー訂正プロセスを省略し簡略化した NewHope-Simple[12] が存在する。

補足情報: NIST PQC 標準化プロジェクトの 第 2 ラウンドの選定レポート [112, p.16] によると，NewHope と CRYSTALS-Kyber の間で比較が行われた。双方ともに dual LWE 形式の構造を持つ格子上で考え，数論変換を用いた高速化を行うという方針で設計されている。Core-SVP ベースの困難性評価では双方とも同程度の強度であったが，実装時のベンチマークの結果は CRYSTALS-Kyber の方が若干良かった。また，安全性の根拠に用いている問題が Ring-LWE と Module-LWE という違いがあり，パラメータ設定の自由度において不利であったようである。

3.3.8 NTRU

歴史: NTRU 暗号方式自体の歴史は長く，1996 年に国際会議 CRYPTO の Rump Session において発表され，その後 1998 年に国際会議 ANTS において発表された論文 [86] が方式の源流となる。本節では，歴史的な NTRU ではなく，NIST PQC 標準化プロジェクトの Round 3 Finalist に選定された公開鍵暗号方式 NTRU について説明する[*14]。

2017 年 11 月の NIST PQC 標準化プロジェクトの公募に提出された 2 件の方式, Andreas Hülsing, Joost Rijneveld, John M. Schanck, Peter Schwabe らにより提案された NTRU-HRSS-KEM[92] と，Cong Chen, Jeffrey Hoffstein, William Whyte, Zhenfei Zhang らにより提案された NTRUEncrypt[41] が Round 2 に進む際にマージされ名称が NTRU と変更，Round 3 にかけてさらに修正が加えられたものが現在の方式となる。Round 2 submission は [92] と [41] の著者 8 名に Oussama Danba を加えた合計 9 名での提案，Round 3 submission はさらに齋藤 恆和，草川 恵太，山川 高志の 3 名を加えた合計 12 名での提案となった。

現在の最新版は 2020 年 9 月に公表された仕様書 [42] である。本節の記述はこの仕様書に従う。

参照 URL: 開発者による公式ページ `https://ntru.org/` を参照した。

設計原理: NTRU は NTRU 格子上の計算困難問題に安全性の根拠を置く公開鍵暗号方式である。具体的には鍵復元攻撃の困難性が NTRU 格子上の短いベクトルを求める問題，平文復元攻撃の困難性が，ターゲットベクトルに近い NTRU 格子上の点を求める問題[*15]として捉えられる。

ベースとなる公開鍵暗号方式は ANTS バージョン [86] の NTRU と比較して，鍵 f, g のサンプリング空間の変更，メッセージ多項式のマスキング手法の変更，暗号化関数の脱乱択化[*16] などの改良が行われている。暗号化関数が決定

[*14] Round 3 Alternative Candidates の中の NTRU Prime とは異なる方式であることに注意。

[*15] 最短・最近ベクトル問題や限界距離復号問題と似ているが，設定が少し異なるため，このように表現する。

[*16] 一般に，脱乱択化 (derandomizing) とはアルゴリズム中で乱数を用いるサブルーチンを，ほぼ同等の性能を保ったままに乱数を用いない決定的なサブルーチンに置き換えることを示す。元々の NTRU では暗号化関数内で平文をマスキングする多項式をランダムに生成していたが，提案バージョンでは多項式を関数の引数とし関数自体は決定的なものとなっている。

的であるためベースの方式自体は IND-CPA 安全性を持たないが，この性質を用いることで IND-CCA2 KEM の構成において，単純な藤崎-岡本変換を用いた構成と比較してハッシュ関数の呼び出し回数を削減することが可能である。

提案ではベースとなる公開鍵暗号方式の OW-CPA 安全性を格子問題の困難性に還元した後に，齋藤ら [137] において提案された implicit rejection の導入による NTRU-HRSS-KEM の改良の構成をベースとして，デカプセル化時の再暗号化処理のスキップによる IND-CCA2 KEM の構成を行っている。最終的な方式の IND-CCA2 安全性は適当な仮定を置くことで ROM, QROM モデルにおいて元の方式の OW-CPA 安全性に帰着される。ただし標準的な仮定においては QROM モデルでの還元はタイトではなく，いくつかの non-standard な仮定を置くことでタイトになる [5, p.39]。

アルゴリズムの詳細:

以下，表 3.29，3.30，3.31 に ANTS 版の NTRU と NIST PQC 標準化プロジェクトの Round 3 提案版の NTRU の公開鍵暗号方式を並置して解説する。

パブリックパラメータは以下で与えられる。仕様書 [42] には NTRU-HPS と NTRU-HRSS の 2 系統のパラメータセットが存在し，それぞれ $\boldsymbol{f}, \boldsymbol{g}$ のサンプリング空間，Lift 関数の構成などが異なる。

- n を定義多項式の次数，$\Phi_1 = x - 1, \Phi_n = (x^n - 1)/(x - 1)$ を円分多項式，p, q を素数の法とする。多項式の次数 n は素数で，$2, 3$ が $\mathbb{Z}_n$ の原始元となるように選ぶ。
- 素数 q と多項式 $F(x)$ に対して，記号 $\mathbb{Z}[x]/(q, F)$ で剰余環 $\mathbb{Z}_q[x]/F(x)$ を表す。特に $R/q, S/q$ はそれぞれ $\mathbb{Z}_q[x]/(\Phi_1\Phi_n), \mathbb{Z}_q[x]/(\Phi_n)$ を定義する。n の取り方より，$S/2, S/3$ は有限体となるため，0 以外の元に常に逆元が存在することになる。
- 集合 $\mathcal{T}$, を係数が $\{0, \pm 1\}$ の多項式で次数が $n - 2$ 以下のものの全体とし，

$$\mathcal{T}' := \left\{ \boldsymbol{v} = \sum_{i=0}^{n-2} v_i x^i \in \mathcal{T} : \sum_{i=0}^{n-3} v_i v_{i+1} \geq 0 \right\}$$

  とする。
- $\mathcal{L}(d_1, d_2)$ は ANTS 版 NTRU のサンプリング空間を定義するために用いられる。次数 $n - 1$ 以下の多項式で d_1 個の係数が $+1$，d_2 個の係数が -1，残りは 0 であるものの集合とする。

暗号アルゴリズムの中で用いられるサブルーチン群は以下で与えられる。

- 多項式 $\boldsymbol{a}$ に対して，関数 $\underline{\mathsf{S3}}(\boldsymbol{a})$ を $\boldsymbol{b} \equiv \boldsymbol{a} \bmod (3, \Phi_n)$ を満たすもので，次数 $n - 2$ 以下かつ係数が $\{0, \pm 1\}$ となるものとする。これを $S/3$ の代表元とする。
- $\mathsf{Lift}(\boldsymbol{m})$ 関数は，メッセージ $\boldsymbol{m}$ のマスキングに用いられる。暗号文を $\boldsymbol{c} = \boldsymbol{r} \cdot \boldsymbol{h} + \boldsymbol{m}$ にって計算する NTRU 系の暗号の場合，$\mathcal{L}_f, \mathcal{L}_g$ の取り方によっては IND-CPA 安全性を持たない可能性がある [42, p.22]。そのため，メッセージ多項式 m を一度別の形にマスキングする必要がある。NTRUEncrypt[41] では，ランダム多項式 t を足しこむことで実現していたのだが，NTRU-HRSS[92] ではこの機能を Lift を用いて実現している。Lift 関数は $\underline{\mathsf{S3}}$ 関数を用いて実現され高速実装が可能であり，しかもマスキングのための多項式 t をサンプリングするという手間がなくなるため，より実装が単純・高速となる。そのため，Round 2 のマージにおいて NTRU-HRSS のアイディアが残った形である。

以下の表 3.28 に鍵多項式 f, g，暗号化に用いるランダム多項式 r，平文多項式 m の空間と Lift 関数の違いをまとめる。$\underline{\mathsf{S3}}(\boldsymbol{m}/(\Phi_1))$ の高速計算法は文献 [91, Append. B] に掲載されている。

ANTS 版 NTRU では Lift 関数は明示されていないが，[42, Sect. 1.3.1] によると $\underline{\mathsf{S3}}(\mathsf{Lift}(\boldsymbol{m})) = \boldsymbol{m}$ を満たす単射 $\mathcal{L}_m \to \mathbb{Z}[x]$ として解釈できる。

NTRU の鍵生成関数（3.29）の詳細を記述する。最初に Sample_fg 関数により $(\boldsymbol{f}, \boldsymbol{g})$ を $\mathcal{L}_f \times \mathcal{L}_g$ から一様ランダム

表 3.28: 方式ごとのサンプリング空間，Lift 関数の違い

	ANTS 版 NTRU [42, Sect. 1.3.1]	NTRU-HPS [42, Sect. 1.3.2]	NTRU-HRSS [42, Sect. 1.3.3]
$\mathcal{L}_f$	$\mathcal{L}(d_f, d_f - 1)$	$\mathcal{T}$	$\mathcal{T}_+$
$\mathcal{L}_g$	$\mathcal{L}(d_f, d_f - 1)$	$\mathcal{T}(q/8 - 2)$	$\{\Phi_1 \cdot \boldsymbol{v} : \boldsymbol{v} \in \mathcal{T}_+\}$
$\mathcal{L}_r$	$\{p \cdot \phi : \phi \in \mathcal{L}(d, d)\}$	$\mathcal{T}$	$\mathcal{T}$
$\mathcal{L}_m$	係数が $[-p/2, p/2]$ に含まれる多項式の集合	$\mathcal{T}(q/8 - 2)$	$\mathcal{T}$
$\mathsf{Lift}(\boldsymbol{m})$ 関数	-	$\underline{\mathsf{S3}}(\boldsymbol{m})$	$\Phi_1 \cdot \underline{\mathsf{S3}}(\boldsymbol{m}/\Phi_1)$

表 3.29: ANTS 版 NTRU および NIST PQC 版 NTRU における鍵生成関数の比較

	ANTS NTRU [86, Sect. 1.2] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	NIST PQC NTRU [42, Figure 9] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
1:	$\boldsymbol{f} \leftarrow \mathsf{Sample\_f}()$	
	// $\Phi_1\Phi_n$ の中で可逆な元をサンプリングする	
	$\boldsymbol{g} \leftarrow \mathsf{Sample\_g}()$	$(\boldsymbol{f}, \boldsymbol{g}) \leftarrow \mathsf{Sample\_fg}()$
		$\boldsymbol{f}_q \leftarrow (1/\boldsymbol{f}) \bmod (q, \Phi_n)$
2:	$\boldsymbol{h} \leftarrow (3\boldsymbol{g}/\boldsymbol{f}) \bmod (q, \Phi_1\Phi_n)$	$\boldsymbol{h} \leftarrow (3\boldsymbol{g} \cdot \boldsymbol{f}_q) \bmod (q, \Phi_1\Phi_n)$
		$\boldsymbol{h}_q \leftarrow (1/\boldsymbol{h}) \bmod (q, \Phi_n)$
3:	$\boldsymbol{f}_p \leftarrow (1/\boldsymbol{f}) \bmod (3, \Phi_1\Phi_n)$	$\boldsymbol{f}_p \leftarrow (1/\boldsymbol{f}) \bmod (3, \Phi_n)$
return	$pk = \boldsymbol{h}, sk = (\boldsymbol{f}, \boldsymbol{f}_p)$	$pk = \boldsymbol{h}, sk = (\boldsymbol{f}, \boldsymbol{f}_p, \boldsymbol{h}_q)$

にサンプリングする。ANTS 版では $\mathcal{L}_f$ からランダムにサンプリングを行い，$\bmod (2, \Phi_1\Phi_n), \bmod (3, \Phi_1\Phi_n)$ の双方で可逆であることを確認し，可逆でない場合にはサンプリングをやり直していた。一方で，NTRU-HPS,NTRU-HRSS では構成から可逆性が保証されているため，可逆性検査は行わない。

秘密鍵の中に $\boldsymbol{h}_q$ が含まれるのは復号関数の中で暗号化に用いた多項式 $\boldsymbol{r}$ を復元するためである。

表 3.30: ANTS 版 NTRU および NIST PQC 版 NTRU における暗号化関数の比較

	ANTS NTRU [86, Sect. 1.3] $\mathsf{Enc}(pk = h, \boldsymbol{m} \in \mathcal{L}_m) \to \boldsymbol{c}$	NIST PQC NTRU [42, Figure 9] $\mathsf{Enc}(pk = h, \boldsymbol{m} \in \mathcal{L}_m; \boldsymbol{r}) \to \boldsymbol{c}$
1:	$\boldsymbol{r} \leftarrow \mathsf{Sample\_r}()$	
2:	$\boldsymbol{c} \leftarrow (\boldsymbol{r} \cdot \boldsymbol{h} + \boldsymbol{m}) \bmod (q, \Phi_1\Phi_n)$	$\boldsymbol{m}' \leftarrow \mathsf{Lift}(\boldsymbol{m})$
		$\boldsymbol{c} \leftarrow (\boldsymbol{r} \cdot \boldsymbol{h} + \boldsymbol{m}') \bmod (q, \Phi_1\Phi_n)$
return	$\boldsymbol{c}$	$\boldsymbol{c}$

表 3.30 に暗号化関数を記述する。この部分はオリジナルの ANTS 版 NTRU とほぼ同様であるが，暗号化に用いるランダム多項式 $\boldsymbol{r}$ が関数の入力として明示されている点，平文多項式の表現を Lift 関数によって変えている点が異なる。

ANTS 版 NTRU では暗号化時に乱数として生成された $\boldsymbol{r}$ を平文の一部として扱うことで，暗号化関数が決定的なものとなる。これにより，IND-CCA2 KEM を構成する際のタイトな還元を実現している [137]。

表 3.31 の復号関数の説明を行う。暗号化関数が脱乱択化されたことで，復号関数は ANTS 版のものとは大きく異な

表 3.31: ANTS 版 NTRU および NIST PQC 版 NTRU における復号関数の比較

	ANTS NTRU [86, Sect. 1.4] $\mathsf{Dec}(sk=(\boldsymbol{f},\boldsymbol{f}_p),\boldsymbol{c})\to\boldsymbol{m}'$	NIST PQC NTRU [42, Figure 9] $\mathsf{Dec}(sk=(\boldsymbol{f},\boldsymbol{f}_p,\boldsymbol{h}_q),\boldsymbol{c})\to(\boldsymbol{r},\boldsymbol{m},\mathsf{flag})$
0:		**if** $\boldsymbol{c}\not\equiv 0 \pmod{(q,\Phi_1)}$ **return** $(0,0,1)$
1:	$\boldsymbol{a}\leftarrow(\boldsymbol{c}\cdot\boldsymbol{f}) \bmod (q,\Phi_1\Phi_n)$	$\boldsymbol{a}\leftarrow(\boldsymbol{c}\cdot\boldsymbol{f}) \bmod (q,\Phi_1\Phi_n)$
2:	$\boldsymbol{m}'\leftarrow(\boldsymbol{a}\cdot\boldsymbol{f}_p) \bmod (3,\Phi_1\Phi_n)$	$\boldsymbol{m}\leftarrow(\boldsymbol{a}\cdot\boldsymbol{f}_p) \bmod (3,\Phi_n)$
		$\boldsymbol{m}'\leftarrow\mathsf{Lift}(\boldsymbol{m})$
		$\boldsymbol{r}\leftarrow((\boldsymbol{c}-\boldsymbol{m}')\cdot\boldsymbol{h}_q) \bmod (q,\Phi_n)$
		if $(\boldsymbol{r},\boldsymbol{m})\in\mathcal{L}_r\times\mathcal{L}_m$ **return** $(\boldsymbol{r},\boldsymbol{m},0)$
		else return $(0,0,1)$
return	$\boldsymbol{m}'$	$(\boldsymbol{r},\boldsymbol{m},\mathsf{flag})$

るものになる。出力が平文の $(\boldsymbol{r},\boldsymbol{m})$ の他に，復号が失敗したかどうかを示すフラグ flag を返す。このフラグがデカプセル化時の implicit rejection に用いられる。

暗号化時の $\mathcal{L}_g$ の取り方，Lift 関数の性質から正しく作られた暗号文の場合ステップ 0 の $\boldsymbol{c}\equiv 0\ (q,\Phi_1)$ が保証される。$\mathsf{flag}=0$ の場合にこの等式が保証されることは，デカプセル化関数内での再暗号化スキップのために必要である。ステップ 2 で復元された $\boldsymbol{m}$ が $S/3$ の代表元となっていることは保証できないため，Lift 関数を用いて $\boldsymbol{m}'$ の復元を行い，それを用いて $\boldsymbol{r}$ を復元する。$(\boldsymbol{r},\boldsymbol{m})$ が正常な平文空間 $\mathcal{L}_r\times\mathcal{L}_m$ に含まれているならば $\mathsf{flag}=0$ をセットして復号結果を返し，そうでなければ失敗として $\boldsymbol{r}=\boldsymbol{m}=0$ とし，失敗フラグを立てて値を返す。

安全性とパラメータ: ベースとなる公開鍵暗号方式の OW-CPA 安全性の具体的な困難性を評価するために鍵復元攻撃と平文復元攻撃が考えられている。公開鍵から秘密鍵を復元する問題は，環の定義多項式と公開鍵多項式によって定義される格子内において，秘密鍵 $(\boldsymbol{f},\boldsymbol{g})$ に対応する短いベクトル[*17]を発見する問題として捉えることができる。また，暗号文から平文を復元する問題も $\boldsymbol{h}_q$ から定義される格子内で，$(0,\boldsymbol{c})$ に近いベクトルを探索する問題として捉えられるため，最近ベクトル問題の埋め込みによりこちらも格子内の短いベクトルを求める問題として定式化することが可能である。

以上により，暗号方式のパラメータ設定には与えられた格子内の短いベクトルを BKZ アルゴリズムを用いて発見するために必要なブロックサイズ β を求め，具体的な計算量は Core-SVP による評価を行っている。

Core-SVP の計算量評価には，篩アルゴリズムで用いる巨大なメモリ空間にアクセスするためのコストを定数と仮定した non-local model およびそうでないと仮定した local model の双方を用いた個別の評価 [42, Sect 5.3] を行っている。

秘密鍵サイズの括弧内は KEM のもので，32Bytes の乱数列 s の分大きくなる。

3.3.9 SABER

歴史: SABER は NIST PQC 標準化プロジェクトの公募への応募方式の一つ 2017 年 11 月に Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren の 4 名により公表され，その後同著者により 2018 年 5 月に国際会議 AFRICACRYPT において査読付き国際会議論文として発表された [53]。

NIST PQC 標準化プロジェクトの 第 2 ラウンド提出時において安全性証明に関わる微修正が行われ，第 3 ラウン

[*17] 多項式環や格子の構成等の違いにより秘密鍵と最短ベクトルが対応しない，最短ベクトルが複数存在するなどの状況があるため，短いベクトルという表現としている。

表 3.32: NTRU のパラメータ [42, Sect. 1.6, 3.2]. 公開鍵，秘密鍵，平文，暗号文サイズの単位はそれぞれ Byte である。

パラメータ名	(n, p, q)	安全性レベル		公開鍵	秘密鍵	平文	暗号文
		non-local	local	サイズ	サイズ	サイズ	サイズ
ntruhps2048509	$(509, 3, 2048)$	-	レベル 1	699	903(935)	204	699
ntruhps2048677	$(677, 3, 2048)$	レベル 1	レベル 3	930	1,202(1,234)	272	930
ntruhps4096821	$(821, 3, 4096)$	レベル 3	レベル 5	1,230	1,558(1,590)	328	1,230
ntruhrss701	$(701, 3, 8192)$	レベル 1	レベル 3	1,138	1,418(1,450)	280	1,138

ドからは新たに Andrea Basso, Jose Maria Bermudo Mera, Michiel Van Beirendonck の 3 名が加わり，開発者は合計 7 名となっている。現在の最新版は Round 3 Finalist に提出された [27] であり，以下の記述はこの仕様書に従う。

参照 URL: 開発者による公式ページ `https://www.esat.kuleuven.be/cosic/pqcrypto/saber/` およびリファレンス実装 `https://github.com/KULeuven-COSIC/SABER` を参照した。

設計原理: SABER は Module-LWR 問題を安全性の根拠とする公開鍵暗号方式であり，LWE 暗号におけるノイズ付加計算をラウンディング演算に置き換えた暗号方式を構成のひな型としている。基本となる方式に対して Module 化と実装上の改良のための修正を行い IND-CPA 安全な暗号方式を構成，藤崎-岡本変換により IND-CCA2 KEM としたものである。仕様書の設計原理 [27, Sect. 4] の項には Regev[131] 暗号の “LWR version” であると記述されている一方で，Second PQC Standardization Conference の発表スライド [52, p.5] では Lindner-Peikert[100] 型（dual-LWE 型）の構成を原型としている。数式の比較から，どちらも原型と捉えることが可能であり，本報告書では仕様書に従い LWE 型に分類する。

LWR 型暗号方式の利点として，LWE 暗号の実装時に必要とされる離散 Gauss 分布等からのサンプリング計算の回避が挙げられる。また，処理を行う際の法 p, q を 2 のべき乗とすることでラウンディング処理がビット列の部分的なコピーのみで完了すること，同様に鍵となる行列 $A \in R_q^{\ell \times \ell}$ の生成が乱数生成ルーチンからのビット列のコピーのみで完了することから，高速処理が可能であるという特徴がある。

一方で，2 のべき乗の形で p, q をとることにより，数論変換を用いた多項式同士の乗算が単純に適用できなくなるという欠点があるが仕様書 [27, Sect.4] によると，SABER で用いられる多項式は 256 次であり，通常の乗算方法を用いても NTT によるものと処理時間に大きな差は無いと主張されている。また，[47] のように一度大きな剰余空間で乗算を計算した後に $\mathbb{Z}_p, \mathbb{Z}_q$ の世界に引き戻す実装テクニックも開発されており，多項式の乗算による効率の低下に関しての大きな問題は無いと考えられる。

IND-CPA 安全な公開鍵暗号から IND-CCA2 KEM の構成には Hofheinz ら [88] による藤崎-岡本変換の変種を用いており，ROM,QROM モデルの双方で安全であることが示されている [27, Sect. 6]。公開鍵暗号から KEM の具体的な構成は [27, Algorithm 4-6] 参照。

アルゴリズムの詳細: 表 3.33，3.34，3.35 に Regev 暗号の LWR 版と SABER（IND-CPA 安全な基本バージョン）の鍵生成，暗号化，復号アルゴリズムを並置する。

パブリックパラメータは以下で与えられる。

- n: 環を定義するための多項式 $x^n + 1$ の次数であり，全てのパラメータセットで $n = 256$ とする。
- p, q, T: ラウンディングの大きさを決定するパラメータ。全て 2 のべき乗の形で，$\epsilon_q = \log_2(q), \epsilon_p = \log_2(p), \epsilon_T = \log_2(T)$ とし，$\epsilon_q > \epsilon_p > \epsilon_T$ とする。つまり $T|p|q$ の関係がある。計算を行う環は $R_q :=$

$\mathbb{Z}_q[x]/(x^n+1)$ とし，R_q の元を成分とする。

- l: モジュール格子のランクである。ベクトル，行列をそれぞれ $R_q^{l\times 1}, R_q^{l\times l}$ 等で表現する。
- 平文空間は $R_2 := \mathbb{Z}_2[x]/(x^n+1)$ であり，256bits の情報を格納できる。

表 3.33: Regev 暗号の LWR 版および SABER における鍵生成関数の比較

	Regev 暗号の LWR 版 [52, p.7] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	SABER[27, Algorithm 1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
1:	A: ランダム行列	$\mathsf{seed}_A \xleftarrow{\$} \{0,1\}^{256}$; $A \leftarrow \mathsf{gen}(\mathsf{seed}_A) \in R_q^{l\times l}$
2:	$\boldsymbol{s}$: 短いランダムベクトル	$\mathsf{r} \xleftarrow{\$} \{0,1\}^{256}$; $\boldsymbol{s} \leftarrow \beta_\mu(R_q^{l\times 1}; \mathsf{r})$
3:	$\boldsymbol{b} = \lfloor A\boldsymbol{s} \rceil_{p/q}$	$\boldsymbol{b} = ((A^T\boldsymbol{s} + \boldsymbol{h}) \bmod q) \gg (\epsilon_q - \epsilon_p) \in R_p^{l\times 1}$
return	$pk = (A, \boldsymbol{b}), sk = \boldsymbol{s}$	$pk = (\mathsf{seed}_A, \boldsymbol{b}), sk = \boldsymbol{s}$

表 3.33 の左側は Regev による LWE 暗号 [131] におけるノイズ付加関数 $A\boldsymbol{s} + \boldsymbol{e}$ をラウンディング関数 $\lfloor A\boldsymbol{s} \rceil_{p/q}$ へと置き換えたものである。ただし，実数 x に対して $\lfloor x \rceil_{p/q} := \lfloor x \cdot (p/q) \rceil$ とし，ベクトル，行列に対しては成分ごとにその操作を行うものとする。また，自然数 a, ϵ に対して $a \gg \epsilon$ は右シフト演算の結果 $a \cdot 2^{-\epsilon}$ の整数部分を返すものであり，$b \in R_q$ に対しては $b \gg \epsilon$ は成分ごとの右シフトを，ベクトル $\boldsymbol{b} \in R_p^{l\times 1}$ に関しても成分ごとの右シフトを行った結果とする。

表 3.33 の右側，SABER の鍵生成関数を説明する。ステップ 1 の gen 関数は 256bits の列を種として，R_q を成分とした擬似ランダムな $l \times l$ 行列を生成する。各成分は R_q 内の一様分布とする。

ステップ 2 の $\beta_\mu(R_q^{l\times 1}; r)$ はビット列 r を種として R_q 成分 l 次元の擬似ランダムベクトルを出力する関数である。ここで，μ は偶数であるとし，各成分の多項式は係数を独立に，パラメータ μ の二項分布の出力から平均値 $\mu/2$ を引いた値をサンプリングしたものとする。

ステップ 3 の公開鍵ベクトル $\boldsymbol{b}$ の生成は $A^T\boldsymbol{s}$ のラウンディングによるものだが，p, q がともに 2 のべき乗 $\varepsilon_p, \varepsilon_q$ であることから p/q を掛けた後のラウンディング処理が

$$\lfloor x \rceil_{p/q} := \left\lfloor \frac{p}{q} x \right\rceil = \left\lfloor \frac{p}{q} x + \frac{1}{2} \right\rfloor = \left\lfloor \frac{p}{q} \left(x + \frac{q}{2p} \right) \right\rfloor = \left\lfloor \left(x + 2^{\varepsilon_q - \varepsilon_p - 1} \right) 2^{\varepsilon_p - \varepsilon_q} \right\rfloor \tag{3.6}$$

と表現され，入力 x に定数 $h = 2^{\varepsilon_q - \varepsilon_p - 1}$ を加えた後，$2^{\varepsilon_p - \varepsilon_q}$ による乗算と切り捨て処理が $(\varepsilon_q - \varepsilon_p)$ ビットの右シフトで実現される。表中の $\boldsymbol{h}$ は，係数が全て h の多項式を成分として持つ $R_q^{l\times 1}$ のベクトルを示す。

出力される公開鍵の形式はデータ量削減のため $(A, \boldsymbol{b})$ の替わりに，行列 A を生成するためのシードを用いて $pk = (\mathsf{seed}_A, \boldsymbol{b})$ としている。

表 3.34 の右側，SABER の暗号化関数を説明する。平文空間は $R_2 = \mathbb{Z}_2[x]/(x^n+1)$ で，$n = 256$bits の情報を格納する。最初にステップ 0 として，暗号化処理に用いる行列 A を seed_A から復元する。次にステップ 1 では暗号化用のランダムベクトル $\boldsymbol{s}'$ を種となるビット列 r を用いて生成するが，与えられていなかった場合には 256bits の乱数列を擬似乱数生成器によって生成する。Enc 関数に種となるビット列が与えられているのは後にこの関数が藤崎-岡本変換を用いた IND-CCA KEM の構成に使われるためである。$\boldsymbol{s}'$ を生成する関数 β_μ，およびステップ 2 での $A\boldsymbol{s}'$ のラウンディングによる $\boldsymbol{b}'$ の生成は鍵生成と同様である。ステップ 3 では $\boldsymbol{b}, \boldsymbol{s}'$ を用いて暗号化を行うが，$(\cdots) \gg (\epsilon_p - \epsilon_T)$ の計算は (3.6) 式における p/q の役割を T/p に置き換えたものである。$2^{\epsilon_p - 1}m$ は係数が 0 もしくは $2^{\epsilon_p - 1}$ の多項式 $(\in R_p)$ となるため，復号関数内で $v' + h_1$ のラウンディングを鍵を用いて小さいノイズに変換することで平文が復元可能となる。

表 3.35 の右側，SABER の復号関数では (3.6) 式と同様の原理により $\left\lfloor \frac{2}{q} \cdot \left(\frac{T}{p} v - c_m \right) \right\rceil$ が計算される。$h_2 \in R_q$ は全ての係数が $2^{\epsilon_p - 2} - 2^{\epsilon_p - \epsilon_T - 1} + 2^{\epsilon_q - \epsilon_p - 1}$ である多項式とする。

表 3.34: Regev 暗号の LWR 版および SABER における暗号化関数の比較

	Regev 暗号の LWR 版 [52, p.7] $\mathsf{Enc}(pk=(A,b), m\in\{0,1\})\to \mathsf{ct}$	SABER[27, Algorithm 2] $\mathsf{Enc}(pk=(\mathsf{seed}_A,b), m\in R_2; r)\to \mathsf{ct}$
0:		$A\leftarrow \mathsf{gen}(\mathsf{seed}_A)\in R_q^{l\times l}$ //行列 A の復元
1:	$\boldsymbol{s}'$: 短いランダムベクトル	$\boldsymbol{s}'\leftarrow \beta_\mu(R_q^{l\times 1}; r)$
2:	$\boldsymbol{b}'=\lfloor A^T\boldsymbol{s}'\rceil_{p,q}$	$\boldsymbol{b}'=((A\boldsymbol{s}'+\boldsymbol{h}) \bmod q)\gg(\epsilon_q-\epsilon_p)\in R_p^{\ell\times 1}$
3:	$c_m=\left\lfloor \boldsymbol{b}^T\boldsymbol{s}'-m\cdot\left\lfloor\frac{p}{2}\right\rceil\right\rceil_{T/p}$	$v'=\boldsymbol{b}^T(\boldsymbol{s}' \bmod p)\in R_p$ $c_m=(v'+h_1-2^{\epsilon_p-1}m \bmod p)\gg(\epsilon_p-\epsilon_T)\in R_T$
return	$\mathsf{ct}=(c_m,\boldsymbol{b}')$	$\mathsf{ct}=(c_m,\boldsymbol{b}')$

表 3.35: Regev 暗号の LWR 版および SABER における復号関数の比較

	Regev 暗号の LWR 版 [52, p.7] $\mathsf{Dec}(sk=\boldsymbol{s}, \mathsf{ct}=(c_m,\boldsymbol{b}'))\to m'$	SABER[27, Algorithm 3] $\mathsf{Dec}(sk=\boldsymbol{s}, \mathsf{ct}=(c_m,\boldsymbol{b}'))\to m'$
1:	$v=(\boldsymbol{b}')^T\boldsymbol{s}$	$v=(\boldsymbol{b}')^T(\boldsymbol{s} \bmod p)\in R_p$
2:	$m'=\left\lfloor\frac{2}{q}\cdot\left(\frac{T}{p}v-c_m\right)\right\rceil$	$m'=((v-2^{\epsilon_p-\epsilon_r}c_m+h_2) \bmod p)\gg(\epsilon_p-1)\in R_2$
return	m'	m'

安全性とパラメータ: ベースとなる IND-CPA 安全な公開鍵暗号の安全性は環 $\mathbb{Z}[x]/(x^{256}+1)$ 上の Module-LWR 問題に帰着されることが示されている。Module-LWR 問題の具体的な困難性評価は，Albrecht らによる LWE 問題，NTRU 問題の困難性シミュレータ [8] を LWR 問題向けに修正したものを用いている。

SABER の安全性を決めるパラメータは l, n, q, p, T, μ の 6 個であり，アルゴリズムの詳細で説明したのと同様に，l が Module-LWR 問題のランク，n が多項式の次数であり，大きいほど安全性が上がるがラウンディング時のノイズの蓄積により復号エラー率が上がる。法 q を大きくとることで Module-LWR の格子体積が大きくなり安全性が下がるがラウンディング時のノイズに強くなるため復号エラー率が下がる。ラウンディングパラメータ p, T を大きくとることでラウンディング時のノイズに相当するものが大きくなり，安全性が上がると同時に復号エラー率も上がる。秘密鍵サンプリングの範囲を指定する μ は大きくとることで秘密鍵ベクトルのサンプリング空間が広がり安全性が上がるが，ラウンディング時のノイズが大きくなり復号エラー率が上がる。

表 3.36: SABER のパラメータ [27, Table 1]. 公開鍵，秘密鍵，平文，暗号文サイズの単位はそれぞれ Byte である。

(l, n, q, p, T, μ)	安全性レベル	公開鍵暗号			鍵カプセル化後		
		公開鍵サイズ	秘密鍵サイズ	暗号文サイズ	公開鍵サイズ	秘密鍵サイズ	暗号文サイズ
$(2, 256, 2^{13}, 2^{10}, 2^3, 10)$	レベル 1	672	832(256)	736	672	1,568(992)	736
$(3, 256, 2^{13}, 2^{10}, 2^4, 8)$	レベル 3	992	1,248(288)	1,088	992	2,304(1,344)	1,088
$(4, 256, 2^{13}, 2^{10}, 2^6, 6)$	レベル 5	1,312	1,664(384)	1,472	1,312	3,040(1,760)	1,472

秘密鍵サイズは一つ目の数字が圧縮前，括弧内の数字が圧縮後のものを示す。秘密鍵ベクトル $\boldsymbol{s}\in R_q$ の各成分が $[-\mu/2, \mu/2]$ の範囲であることから，$\lceil\log_2 q\rceil$ bits の整数として保存するのではなく，下位 $\lceil\log_2\mu\rceil$ bits のみを保存す

ることで圧縮できる。

3.4 格子に基づく暗号技術に関するまとめ

格子に基づく暗号技術は，LWE 問題，Ring-LWE 問題，NTRU 問題を安全性の根拠とする方式をはじめ，これまで数多く提案されており，米国 NIST PQC 標準化プロジェクトで提案された暗号技術としては最も多くの暗号がこのカテゴリーに分類されている。

この米国 NIST PQC 標準化プロジェクトを通じて 2022 年 7 月に CRYSTALS-Kyber が標準的な暗号方式として，CRYSTALS-Dilithium および FALCON が標準的な署名方式として選定され，CRYSTALS-Kyber と CRYSTALS-Dilithium については，2024 年 8 月に FIPS 203, FIPS 204 として公開されている [119, 118]。また，CRYSTALS-Kyber と CRYSTALS-Dilithium は 2022 年 9 月に米国国家安全保障局の Commercial National Security Algorithm Suite 2.0 (CNSA2.0) にも選定されている [1]。NIST PQC 標準化プロジェクトの選考プロセスから漏れた方式の中でも，米国以外の公的機関において推奨暗号とされているものが存在する。一例として，FrodoKEM が 2020 年 8 月よりドイツ情報セキュリティ庁 (BSI) の推奨暗号に [141]，2022 年 1 月にはオランダ通信・安全委員会 (NLNCSA) により最も安全な暗号の例として推奨されている [16]。Google 社の Chrome ブラウザには，TLS レイヤーの性能試験目的で搭載された耐量子計算機暗号プロトコル CECPQ1[36] および CECPQ2[130] にそれぞれ NewHope の USENIX 発表バージョン [13] と NTRU が実装されていたが，2023 年 1 月現在ではともに削除されている。IBM 製テープドライブのプロトタイプとして，CRYSTALS-Kyber と CRYSTALS-Dilithium の組み合わせにより暗号化を行うものが制作されている [97]。DNS サーバの一種である PowerDNS において，耐量子計算機性を実現する署名として FALCON のテスト用の実装が行われている [84]。オープンソースライブラリへの導入として，WireGuard VPN protocol への SABER の実装 [90]，WolfSSL への CRYSTALS-Kyber，FALCON の実装 [153]，OpenSSH への Streamlined NTRU Prime の実装 [120] などが存在する他，Open Quantum Safe (OQS) プロジェクトによる liboqs ライブラリには暗号化・鍵交換の方式として CRYSTALS-Kyber，NTRU，SABER，FALCON，FrodoKEM，NTRU-Prime が，署名方式として CRYSTALS-Dilithium と FALCON が実装されている [136]。このように格子に基づく暗号技術の社会実装が徐々に進みつつある。特に，標準化が先行する CRYSTALS-Kyber，CRYSTALS-Dilithium に対するサイドチャネル攻撃とその対策としてマスキング実装が検討されている [144, 150, 48]。

格子に基づく暗号技術の安全性の根拠となる問題としては，先に挙げた LWE 問題，Ring-LWE 問題，NTRU 問題以外にも Compact LWE 問題，Module-LWE 問題，LWR 問題，BDD 問題，SIS 問題他，多くのバリエーションが存在している。一般的な格子問題を解く手法としては，LLL アルゴリズム，BKZ アルゴリズムがよく知られており，LWE 問題については更に SIS 問題や BDD 問題に還元する解析手法が知られている。SVP や LWE/NTRU などの格子問題の解析やそれらの求解アルゴリズムに関する最新研究については [35, 29, 76, 101, 135, 111, 54, 148, 57, 113, 128, 40, 99, 31, 50] を参照。近年，新しい格子問題として格子同型問題 [69] が提案された。（格子同型問題の性質については [28] を参照。）また，格子同型問題の困難性を安全性の根拠とする署名方式 HAWK[65] は，NIST PQC 標準化プロジェクトにおける署名方式の追加公募において，格子に基づく方式の中で第 2 ラウンドにおいて進むことが許された方式である（2024 年 10 月時点）。さらに，量子紛失 LWE サンプリング [58] や，格子問題に対する量子アルゴリズムに関する研究 [44, 49] も近年進展している。

格子問題の困難性をベースとした暗号方式で最初のものは，Ajtai[2] により 1996 年に行われた，SIS 問題が格子問題の最悪時と同等かそれ以上に困難であることの証明およびそれを用いた暗号学的ハッシュ関数の構成である。また，1997 年には Ajtai と Dwork[4] により，unique SVP の最悪困難性を安全性の根拠とした公開鍵暗号が提案されている。この公開鍵暗号方式は翌年，Nguyen らによる解読実験 [116] により必要なパラメータが長大となり実用的でないことが明らかにされたものの，その後の格子に基づく暗号構成の基礎となっている。

1996 年に Hoffstein らによって提案された NTRU 暗号 [86][*18] は，発表当初安全性証明が付けられておらず，攻撃と修正が繰り返されていたが，2011 年 Stehlé ら [145] により方式が修正され，イデアル格子上の問題の困難性に還元可能なことが示されている。一方で，2016 年には subfield attack[7] のような体の構造を使って格子の次元を圧縮する攻撃も提案されており，暗号の構成のためには次元や法の大きさだけでなく，環・体の構造にも注意を払う必要がある。NTRU 格子上の署名方式のサイズ改良 [73]・トラップドア生成 [72] や，NTRU に対する鍵ミスマッチ攻撃の改良 [102]・NTRU 格子の簡約 [24] に関する最新の研究がある。

2005 年に Regev[131] により提案された LWE 問題は，論文発表と同時にそれを暗号の安全性根拠として保障する重要な三つの性質が示された。一つは問題の average-case to worst case reduction，つまりパラメータを固定した際，問題の (秘密ベクトル $\mathbf{s}$ に関する) 平均的な計算量が，最悪計算量 (難しいインスタンスを生成するような $\mathbf{s}$ の集合に対する計算量) と高々多項式倍の違いしか無いことであり，残りの二つは判定 LWE と探索 LWE の等価性，および量子アルゴリズムによる困難な格子問題への還元である。これらの定理を組み合わせることにより，Regev 自身により提案された公開鍵暗号を解読することが平均的に難しいことが示され，その後の様々な LWE ベース暗号の構成の基礎なった。LWE 格子問題への還元に関して，2013 年には古典計算機による還元も示されている [37]。

LWE 問題の欠点である鍵サイズの大きさを改善するため，2010 年には Lyubashevsky ら [106, 107] により Ring-LWE 問題が，2015 年には Langlois ら [96] により Module-LWE 問題が暗号化方式と同時に提案され，LWE 問題における関係と類似の，解読の平均的な困難さが証明されている。一方で，これらの変種とオリジナルの LWE 問題との関係性は自明ではなく，同程度の難しさを持つかどうかは未解決問題である。一般的に Ring(Module)-LWE 問題のインスタンスは LWE 問題のインスタンスとして書きなおすことができるため，LWE 問題は Ring(Module)-LWE 問題よりも困難であるという関係は自明であるが，逆の関係は知られていない。法 q が大きい場合には，Ring-LWE は Module-LWE よりも困難であることが知られている [9]。(Ring/Module-LWE 問題の理論解析の最新研究について [151] を参照。)

実装時の問題として，離散 Gauss 分布を正確に生成することは難しいことが挙げられる。ノイズをある整数区間から一様分布として取った場合でも，格子問題へと量子帰着が可能であることが 2013 年に Döttling ら [61] により示された。この方向性の研究として，Bai ら [23] により提案された，理想的な Gauss 分布を用いた暗号方式とそれを近似的な分布に置き換えた方式の間での安全性の低下を Rényi エントロピーを用いて議論するものがある。

格子に基づく暗号技術は，耐量子計算機暗号としてだけでなく，完全準同型暗号や多重署名などの高機能な暗号方式に応用する研究も数多くある [30, 34, 123, 152, 60, 157, 121, 108, 43, 95, 59, 87, 109, 110]。

また，格子問題の計算機による具体的な求解に関して，2016 年より暗号解読コンテスト LWE Challenge[55] が開催されている。3.1 節に，2024 年 11 月現在の状況について記載した。特に 3.3 節で示された各暗号方式のパラメータから見ると，解が得られている値からは，大きな隔たりがみられる。格子に基づく暗号技術は，各方式毎にパラメータ設定手法に対する制約が異なっていることから，解読コンテストのサイズに基づく解読到達レベルを，具体的な暗号方式の安全性の根拠とすることは，難しいところではあるものの，古典計算機での解読困難性を測る上での検討の一つに値すると考えられる。(最新の BKZ の改良や LWE の解読計算量見積もりについては [149, 154] を参照)

格子に基づく暗号技術の安全性の根拠となる問題は，古典計算機・量子計算機のいずれにおいても現時点で効率的な解読手法は見つかっていないが，格子に基づく暗号技術は未だ研究途上にあり，今後も研究の進捗を注視する必要がある。

[*18] 文献上は 1998 年の国際会議 ANTS だが，初出は CRYPTO1996 の Rump Session である。

第 3 章の参照文献

[1] National Security Agency. Announcing the Commercial National Security Algorithm Suite 2.0. https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF. 2022-09. (2024-12-06 閲覧).

[2] M. Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). STOC. ACM, 1996, pp. 99–108.

[3] M. Ajtai. Generating Hard Instances of the Short Basis Problem. ICALP. Vol. 1644. Lecture Notes in Computer Science. Springer, 1999, pp. 1–9.

[4] M. Ajtai, Cynthia Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. STOC. ACM, 1997, pp. 284–293.

[5] G. Alagic et al. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8413, https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413-upd1.pdf. 2022-07.

[6] M. Albrecht, L. Ducas. Lattice attacks on NTRU and LWE: A history of refinements. London Mathematical Society Lecture Notes 469. Cambridge University Press, 2021, pp. 15–40. Chapter 2.

[7] M. R. Albrecht, S. Bai, L. Ducas. A Subfield Lattice Attack on Overstretched NTRU Assumptions – Cryptanalysis of Some FHE and Graded Encoding Schemes. CRYPTO (1). Vol. 9814. Lecture Notes in Computer Science. Springer, 2016, pp. 153–178.

[8] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, T. Wunderer. Estimate All the {LWE, NTRU} Schemes! SCN. Vol. 11035. Lecture Notes in Computer Science. Springer, 2018, pp. 351–367.

[9] M. R. Albrecht, A. Deo. Large Modulus Ring-LWE $\geq$ Module-LWE. ASIACRYPT (1). Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 267–296.

[10] M. R. Albrecht, L. Ducas, G. Herold, E. Kirshanova, E. W. Postlethwaite, M. Stevens. The General Sieve Kernel and New Records in Lattice Reduction. EUROCRYPT (2). Vol. 11477. Lecture Notes in Computer Science. Springer, 2019, pp. 717–746.

[11] M. R. Albrecht, F. Göpfert, F. Virdia, T. Wunderer. Revisiting the Expected Cost of Solving uSVP and Applications to LWE. ASIACRYPT (1). Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 297–322.

[12] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe. NewHope without reconciliation. (2016). https://eprint.iacr.org/2016/1157.

[13] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe. Post-quantum Key Exchange - A New Hope. USENIX Security Symposium. USENIX Association, 2016, pp. 327–343.

[14] E. Alkim et al. FrodoKEM – Learning with errors key encapsulation. Algorithm specifications and supporting documentation (June 4, 2021). https://frodokem.org/files/FrodoKEM-specification-20210604.pdf. 2021-06. (2024-03-04 閲覧).

[15] E. Alkim et al. NewHope algorithm specifications and supporting documentation Version 1.1 (Updated April 10, 2020). https://newhopecrypto.org/data/NewHope_2020_04_10.pdf. 2020-04. (2024-03-04 閲覧).

[16] General intelligence and security service. Prepare for the threat of quantum computers. https://english.aivd.nl/publications/publications/2022/01/18/prepare-for-the-threat-of-quantumcomputers. 2022-01. (2024-03-04 閲覧).

[17] Annex on FrodoKEM updates. https://frodokem.org/files/FrodoKEM-annex-20230418.pdf. 2024-04. (2024-12-01 閲覧).

[18] R. Avanzi et al. CRYSTALS-Kyber – Algorithm specifications and supporting documentation. https://pq-crystals.org/kyber/data/kyber-specification.pdf. 2017-11. (2024-03-04 閲覧).

[19] R. Avanzi et al. CRYSTALS-Kyber – Algorithm specifications and supporting documentation (version 2.0). https://pq-crystals.org/kyber/data/kyber-specification-round2.pdf. 2019-04. (2024-03-04 閲覧).

[20] R. Avanzi et al. CRYSTALS-Kyber – Algorithm specifications and supporting documentation (version 3.02). https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf. 2021-08. (2024-03-04 閲覧).

[21] S. Bai, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Dilithium algorithm specifications and supporting documentation (Version 3.1). https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf. 2021-02. (2024-03-04 閲覧).

[22] S. Bai, S. D. Galbraith. Lattice Decoding Attacks on Binary LWE. ACISP. Vol. 8544. Lecture Notes in Computer Science. Springer, 2014, pp. 322–337.

[23] S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, R. Steinfeld. Improved Security Proofs in Lattice-Based Cryptography: Using the Rényi Divergence Rather than the Statistical Distance. J. Cryptol. Vol. 31, Num. 2 (2018), pp. 610–640.

[24] H. Bambury, P. Q. Nguyen. Improved Provable Reduction of NTRU and Hypercubic Lattices. PQCrypto (1). Vol. 14771. Lecture Notes in Computer Science. Springer, 2024, pp. 343–370.

[25] A. Banerjee, C. Peikert, A. Rosen. Pseudorandom Functions and Lattices. EUROCRYPT. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 719–737.

[26] M. Barbosa et al. Fixing and Mechanizing the Security Proof of Fiat-Shamir with Aborts and Dilithium. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 358–389.

[27] A. Basso, J. M. Bermudo Mera, J.-P. D'Anvers, A. Karmakar, S. S. Roy, M. Van Beirendonck, F. Vercauteren. SABER: Mod-LWR based KEM (Round 3 Submission). https://www.esat.kuleuven.be/cosic/pqcrypto/saber/files/saberspecround3.pdf. (2024-03-04 閲覧).

[28] B. Bencina, A. Budroni, J.-J. Chi-Domínguez, M. Kulkarni. Properties of Lattice Isomorphism as a Cryptographic Group Action. PQCrypto (1). Vol. 14771. Lecture Notes in Computer Science. Springer, 2024, pp. 170–201.

[29] O. Bernard, A. Lesavourey, TH Nguyen, A. Roux-Langlois. Log-S-unit Lattices Using Explicit Stickelberger Generators to Solve Approx Ideal-SVP. ASIACRYPT (3). Vol. 13793. Lecture Notes in Computer Science. Springer, 2022, pp. 677–708.

[30] W. Beullens, S. Dobson, S. Katsumata, Y.-F. Lai, F. Pintore. Group signatures and more from isogenies and lattices: generic, simple, and efficient. Vol. 91. 6. 2023, pp. 2141–2200.

[31] M. Bolboceanu, Z. Brakerski, D. Sharma. On Algebraic Embedding for Unstructured Lattices. Public Key Cryptography (3). Vol. 14603. Lecture Notes in Computer Science. Springer, 2024, pp. 123–154.

[32] J. W. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, D. Stebila. Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE. CCS. ACM, 2016, pp. 1006–1018.

[33] J. W. Bos et al. CRYSTALS – Kyber: A CCA-Secure Module-Lattice-Based KEM. EuroS&P. IEEE, 2018, pp. 353–367.

[34] C. Boschini, A. Takahashi, M. Tibouchi. MuSig-L: Lattice-Based Multi-signature with Single-Round Online Phase. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 276–305.

[35] K. Boudgoust, E. Gachon, A. Pellet-Mary. Some Easy Instances of Ideal-SVP and Implications on the Partial Vandermonde Knapsack Problem. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 480–509.

[36] M. Braithwaite. Experimenting with post-quantum cryptography. `https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html`. 2023-04. （2024-03-04 閲覧）.

[37] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, D. Stehlé. Classical hardness of learning with errors. STOC. ACM, 2013, pp. 575–584.

[38] Z. Brakerski, V. Vaikuntanathan. Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. CRYPTO. Vol. 6841. Lecture Notes in Computer Science. Springer, 2011, pp. 505–524.

[39] L. G. Bruinderink, A. Hülsing, T. Lange, Y. Yarom. Flush, Gauss, and Reload - A Cache Attack on the BLISS Lattice-Based Signature Scheme. CHES. Vol. 9813. Lecture Notes in Computer Science. Springer, 2016, pp. 323–345.

[40] K. Carrier, T. Debris-Alazard, C. Meyer-Hilfiger, J.-P. Tillich. Reduction from Sparse LPN to LPN, Dual Attack 3.0. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 286–315.

[41] C. Chen, J. Hoffstein, W. Whyte, Z. Zhang. NIST PQ Submission: NTRUEncrypt A lattice based encryption algorithm. `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/NTRUEncrypt.zip`. （2024-03-04 閲覧）.

[42] C. Chen et al. NTRU algorithm specifications and supporting documentation (September 30,2020). `https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/NTRU-Round3.zip`. 2020-09. （2024-03-04 閲覧）.

[43] Y. Chen. sfDualMS: Efficient Lattice-Based Two-Round Multi-signature with Trapdoor-Free Simulation. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 716–747.

[44] Y. Chen, Q. Liu, M. Zhandry. Quantum Algorithms for Variants of Average-Case Lattice Problems via Filtering. EUROCRYPT (3). Vol. 13277. Lecture Notes in Computer Science. Springer, 2022, pp. 372–401.

[45] Y. Chen, P. Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. ASIACRYPT. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, pp. 1–20.

[46] C. Chuengsatiansup, T. Prest, D. Stehlé, A. Wallet, K. Xagawa. ModFalcon: Compact Signatures Based On Module-NTRU Lattices. AsiaCCS. ACM, 2020, pp. 853–866.

[47] C.-M. M. Chung, V. Hwang, M. J. Kannwischer, G. Seiler, C.-J. Shih, B.-Y. Yang. NTT Multiplication for NTT-unfriendly Rings New Speed Records for Saber and NTRU on Cortex-M4 and AVX2. IACR Trans. Cryptogr. Hardw. Embed. Syst. Vol. 2021, Num. 2 (2021), pp. 159–188.

[48] J.-S. Coron, F. Gérard, M. Trannoy, R. Zeitoun. Improved Gadgets for the High-Order Masking of Dilithium. Vol. 2023. 4. 2023, pp. 110–145.

[49] R. Cramer, L. Ducas, B. Wesolowski. Mildly Short Vectors in Cyclotomic Ideal Lattices in Quantum Polynomial Time. J. ACM. Vol. 68, Num. 2 (2021), 8:1–8:26.

[50] R. Cramer, L. Ducas, B. Wesolowski. Short Stickelberger Class Relations and Application to Ideal-SVP. EUROCRYPT (1). Vol. 10210. Lecture Notes in Computer Science. 2017, pp. 324–348.

[51] C. Cremers, S. Düzlü, R. Fiedler, M. Fischlin, C. Janson. BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures. Symposium on Security and Privacy (SP). IEEE, 2021, pp. 1696–1714.

[52] J.-P. D'Anvers. SABER: Module-LWR based KEM. Second PQC Standardization Conference. 2019-08. `https://csrc.nist.gov/Presentations/2019/saber-round-2-presentation`. (2024-03-04 閲覧).

[53] J.-P. D'Anvers, A. Karmakar, S. Sinha Roy, F. Vercauteren. Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM. AFRICACRYPT. Vol. 10831. Lecture Notes in Computer Science. Springer, 2018, pp. 282–305.

[54] D. Dachman-Soled, H. Gong, T. Hanson, H. Kippen. Revisiting Security Estimation for LWE with Hints from a Geometric Perspective. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 748–781.

[55] TU Darmstadt, UC San Diego. LWE Challenge. `https://www.latticechallenge.org/lwe_challenge/challenge.php`. (2024-03-04 閲覧).

[56] TU Darmstadt, UC San Diego. SVP Challenge, Hall Of Fame. `https://www.latticechallenge.org/svp-challenge/halloffame.php`. (2024-03-04 閲覧).

[57] D. Das, A. Joux. Key Recovery Attack on the Partial Vandermonde Knapsack Problem. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 205–225.

[58] T. Debris-Alazard, P. Fallahpour, D. Stehlé. Quantum Oblivious LWE Sampling and Insecurity of Standard Model Lattice-Based SNARKs. STOC. ACM, 2024, pp. 423–434.

[59] J. Devevey, A. Passelègue, D. Stehlé. G+G: A Fiat-Shamir Lattice Signature Based on Convolved Gaussians. ASIACRYPT (7). Vol. 14444. Lecture Notes in Computer Science. Springer, 2023, pp. 37–64.

[60] N. Döttling, D. Kolonelos, R. W. F. Lai, C. Lin, G. Malavolta, A. Rahimi. Efficient Laconic Cryptography from Learning with Errors. EUROCRYPT (3). Vol. 14006. Lecture Notes in Computer Science. Springer, 2023, pp. 417–446.

[61] N. Döttling, J. Müller-Quade. Lossy Codes and a New Variant of the Learning-With-Errors Problem. EUROCRYPT. Vol. 7881. Lecture Notes in Computer Science. Springer, 2013, pp. 18–34.

[62] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Dilithium algorithm specifications and supporting documentation. https://pq-crystals.org/dilithium/data/dilithium-specification.pdf. 2017-11. (2024-03-04 閲覧).

[63] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. IACR Trans. Cryptogr. Hardw. Embed. Syst. Vol. 2018, Num. 1 (2018), pp. 238–268.

[64] L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehle. CRYSTALS-Dilithium: Digital Signatures from Module Lattices. Cryptology ePrint Archive, Paper 2017/633. 2017. https://eprint.iacr.org/2017/633.

[65] L. Ducas, E. W. Postlethwaite, L. N. Pulles, W. P. J. van Woerden. HAWK: Module LIP Makes Lattice Signatures Fast, Compact and Simple. ASIACRYPT (4). Vol. 13794. Lecture Notes in Computer Science. Springer, 2022, pp. 65–94.

[66] L. Ducas, T. Prest. Fast Fourier Orthogonalization. ISSAC. ACM, 2016, pp. 191–198.

[67] L. Ducas, J. Schanck. Security estimation scripts for Kyber and Dilithium. https://github.com/pq-crystals/security-estimates. 2019-03. (2024-03-04 閲覧).

[68] L. Ducas, M. Stevens, W. P. J. van Woerden. Advanced Lattice Sieving on GPUs, with Tensor Cores. EUROCRYPT (2). Vol. 12697. Lecture Notes in Computer Science. Springer, 2021, pp. 249–279.

[69] L. Ducas, W. P. J. van Woerden. On the Lattice Isomorphism Problem, Quadratic Forms, Remarkable Lattices, and Cryptography. EUROCRYPT (3). Vol. 13277. Lecture Notes in Computer Science. Springer, 2022, pp. 643–673.

[70] S. Düzlü, R. Fiedler, M. Fischlin. BUFFing FALCON without Increasing the Signature Size. IACR Cryptol. ePrint Arch. (2024), p. 710.

[71] T. Espitau, P.-A. Fouque, F. Gérard, M. Rossi, A. Takahashi, M. Tibouchi, A. Wallet, Y. Yu. MITAKA: A Simpler, Parallelizable, Maskable Variant of FALCON. EUROCRYPT (3). Vol. 13277. Lecture Notes in Computer Science. Springer, 2022, pp. 222–253.

[72] T. Espitau, T. Thu Quyen Nguyen, C. Sun, M. Tibouchi, A. Wallet. ANTRAG: Annular NTRU trapdoor generation - Making MITAKA as secure as FALCON. ASIACRYPT (7). Vol. 14444. Lecture Notes in Computer Science. Springer, 2023, pp. 3–36.

[73] T. Espitau, M. Tibouchi, A. Wallet, Y. Yu. Shorter Hash-and-Sign Lattice-Based Signatures. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 245–275.

[74] ETSI TR 103 616 V1.1.1 (2021-09) CYBER; Quantum-safe signatures. https://www.etsi.org/deliver/etsi_tr/103600_103699/103616/01.01.01_60/tr_103616v010101p.pdf. 2021-09. (2024-03-04 閲覧).

[75] M. Fahr et al. When Frodo Flips: End-to-End Key Recovery on FrodoKEM via Rowhammer. CCS. ACM, 2022, pp. 979–993.

[76] J. Felderhoff, A. Pellet-Mary, D. Stehlé. On Module Unique-SVP and NTRU. ASIACRYPT (3). Vol. 13793. Lecture Notes in Computer Science. Springer, 2022, pp. 709–740.

[77] A. Fiat, A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. CRYPTO. Vol. 263. Lecture Notes in Computer Science. Springer, 1986, pp. 186–194.

[78] P.-A. Fouque, F. Gérard, M. Rossi, Y. Yu. ℤalcon: An alternative FPA-free NTRU sampler for Falcon. Third PQC Standardization Conference. 2021-06. (2024-03-04 閲覧).

[79] P.-A. Fouque et al. FALCON: Fast-Fourier lattice-based compact signatures over NTRU. Specification v1.0. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/Falcon.zip. (2024-03-04 閲覧).

[80] P.-A. Fouque et al. FALCON: Fast-Fourier lattice-based compact signatures over NTRU. Specification v1.2 – 01/10/2020. https://falcon-sign.info/falcon.pdf. 2020-10. (2024-03-04 閲覧).

[81] FrodoKEM: Learning With Errors Key Encapsulation – Preliminary Standardization Proposal. https://frodokem.org/files/FrodoKEM_standard_proposal_20241205.pdf. 2024-12. (2025-01-27 閲覧).

[82] C. Gentry, C. Peikert, V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC. ACM, 2008, pp. 197–206.

[83] O. Goldreich, S. Goldwasser, S. Halevi. Public-Key Cryptosystems from Lattice Reduction Problems. CRYPTO. Vol. 1294. Lecture Notes in Computer Science. Springer, 1997, pp. 112–131.

[84] M. Grillere, P. Thomassen, N. Wisiol. FALCON-512 in PowerDNS. https://blog.powerdns.com/2022/04/07/falcon-512-in-powerdns/. 2022-04. (2024-03-04 閲覧).

[85] Q. Guo, T. Johansson, A. Nilsson. A Key-Recovery Timing Attack on Post-quantum Primitives Using the Fujisaki-Okamoto Transformation and Its Application on FrodoKEM. CRYPTO (2). Vol. 12171. Lecture Notes in Computer Science. Springer, 2020, pp. 359–386.

[86] J. Hoffstein, J. Pipher, J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. ANTS. Vol. 1423. Lecture Notes in Computer Science. Springer, 1998, pp. 267–288.

[87] D. Hofheinz, K. Hostáková, R. Langrehr, B. Ursu. On Structure-Preserving Cryptography and Lattices. Public Key Cryptography (3). Vol. 14603. Lecture Notes in Computer Science. Springer, 2024, pp. 255–287.

[88] D. Hofheinz, K. Hövelmanns, E. Kiltz. A Modular Analysis of the Fujisaki-Okamoto Transformation. TCC (1). Vol. 10677. Lecture Notes in Computer Science. Springer, 2017, pp. 341–371.

[89] J. Howe, T. Pöppelmann, M. O'Neill, E. O'Sullivan, T. Güneysu. Practical Lattice-Based Digital Signature Schemes. ACM Trans. Embed. Comput. Syst. Vol. 14, Num. 3 (2015), 41:1–41:24.

[90] A. Hülsing, K.-C. Ning, P. S., F. Weber, P. R. Zimmermann. Post-quantum WireGuard. SP. IEEE, 2021, pp. 304–321.

[91] A. Hülsing, J. Rijneveld, J. M. Schanck, P. Schwabe. High-Speed Key Encapsulation from NTRU. CHES. Vol. 10529. Lecture Notes in Computer Science. Springer, 2017, pp. 232–252.

[92] A. Hülsing, J. Rijneveld, J. M. Schanck, P. Schwabe. NTRU-HRSS-KEM Algorithm Specifications And Supporting Documentation (November 30, 2017). https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/NTRU_HRSS_KEM.zip. 2017-11. (2024-03-04 閲覧).

[93] R. Kannan. Improved Algorithms for Integer Programming and Related Lattice Problems. STOC. ACM, 1983, pp. 193–206.

[94] K. Kim, M. Tibouchi, A. Wallet, T. Espitau, A. Takahashi, Y. Yu, S. Guilley. SOLMAE – Algorithm specifications. https://kpqc.or.kr/images/pdf/SOLMAE.pdf. (2024-03-04 閲覧).

[95] R. W. F. Lai, G. Malavolta. Lattice-Based Timed Cryptography. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 782–804.

[96] A. Langlois, D. Stehlé. Worst-case to average-case reductions for module lattices. Des. Codes Cryptogr. Vol. 75, Num. 3 (2015), pp. 565–599.

[97] M. Lantz. World's first quantum computing safe tape drive. https://www.ibm.com/blogs/research/2019/08/crystals/. 2019-08. (2024-03-04 閲覧).

[98] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen. Vol. 261, Num. 4 (1982), pp. 515–534.

[99] X. Lin, M. Suzuki, S. Zhang, T. Espitau, Y. Yu, M. Tibouchi, M. Abe. Cryptanalysis of the Peregrine Lattice-Based Signature Scheme. Public Key Cryptography (1). Vol. 14601. Lecture Notes in Computer Science. Springer, 2024, pp. 387–412.

[100] R. Lindner, C. Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. CT-RSA. Vol. 6558. Lecture Notes in Computer Science. Springer, 2011, pp. 319–339.

[101] H. Liu, Y. Yu. A Non-heuristic Approach to Time-Space Tradeoffs and Optimizations for BKW. ASIACRYPT (3). Vol. 13793. Lecture Notes in Computer Science. Springer, 2022, pp. 741–770.

[102] Z. Liu, V., J. Ding, C. Cheng, Y. Pan. An Improved Practical Key Mismatch Attack Against NTRU. PQCrypto (1). Vol. 14771. Lecture Notes in Computer Science. Springer, 2024, pp. 322–342.

[103] V. Lyubashevsky. CRYSTALS-Dilithium Round 3 presentation. Third PQC Standardization Conference. 2021-06. https://csrc.nist.gov/Presentations/2021/crystals-dilithium-round-3-presentation. (2024-03-04 閲覧).

[104] V. Lyubashevsky. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. ASIACRYPT. Vol. 5912. Lecture Notes in Computer Science. Springer, 2009, pp. 598–616.

[105] V. Lyubashevsky. Lattice Signatures without Trapdoors. EUROCRYPT. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 738–755.

[106] V. Lyubashevsky, C. Peikert, O. Regev. On Ideal Lattices and Learning with Errors over Rings. EUROCRYPT. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 1–23.

[107] V. Lyubashevsky, C. Peikert, O. Regev. On Ideal Lattices and Learning with Errors over Rings. J. ACM. Vol. 60, Num. 6 (2013), 43:1–43:35.

[108] D. Micciancio, M. Schultz. Error Correction and Ciphertext Quantization in Lattice Cryptography. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 648–681.

[109] D. Micciancio, V. Vaikuntanathan. SoK: Learning with Errors, Circular Security, and Fully Homomorphic Encryption. Public Key Cryptography (4). Vol. 14604. Lecture Notes in Computer Science. Springer, 2024, pp. 291–321.

[110] G. De Micheli, D. Kim, D. Micciancio, A. Suhl. Faster Amortized FHEW Bootstrapping Using Ring Automorphisms. Public Key Cryptography (4). Vol. 14604. Lecture Notes in Computer Science. Springer, 2024, pp. 322–353.

[111] G. De Micheli, D. Micciancio, A. Pellet-Mary, N. Tran. Reductions from Module Lattices to Free Module Lattices, and Application to Dequantizing Module-LLL. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 836–865.

[112] D. Moody et al. Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8309, https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf. 2020-07.

[113] G. Mureau, A. Pellet-Mary, G. Pliatsok, A. Wallet. Cryptanalysis of Rank-2 Module-LIP in Totally Real Number Fields. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 226–255.

[114] M. Naehrig, K. E. Lauter, V. Vaikuntanathan. Can homomorphic encryption be practical? CCSW. ACM, 2011, pp. 113–124.

[115] M. Naehrig et al. FrodoKEM – learning with errors key encapsulation. Algorithm specifications and supporting documentation (November 30, 2017). https://frodokem.org/files/FrodoKEM-specification-20171130.pdf. 2017-11. (2024-03-04 閲覧).

[116] P. Q. Nguyen, J. Stern. Cryptanalysis of the Ajtai-Dwork Cryptosystem. CRYPTO. Vol. 1462. Lecture Notes in Computer Science. Springer, 1998, pp. 223–242.

[117] Q. Nguyen. ANTRAG: Simplifying and improving Falcon Without Compromising Security. https://csrc.nist.gov/csrc/media/Presentations/2024/antrag-simplifying-and-improving-falcon/images-media/nguyen-antrag-pqc2024.pdf. 2024-04. (2024-12-30 閲覧).

[118] NIST. Module-Lattice-Based Digital Signature Standard. NIST FIPS 204, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf. 2024-08.

[119] NIST. Module-Lattice-Based Key-Encapsulation Mechanism Standard. NIST FIPS 203, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf. 2024-08.

[120] OpenSSH 8.9 was released on 2022-02-23. https://www.openssh.com/txt/release-8.9. (2024-03-04 閲覧).

[121] J. Pan, B. Wagner, R. Zeng. Lattice-Based Authenticated Key Exchange with Tight Security. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 616–647.

[122] C. Peikert, V. Vaikuntanathan, B. Waters. A Framework for Efficient and Composable Oblivious Transfer. CRYPTO. Vol. 5157. Lecture Notes in Computer Science. Springer, 2008, pp. 554–571.

[123] R. del Pino, S. Katsumata. A New Framework for More Efficient Round-Optimal Lattice-Based (Partially) Blind Signature via Trapdoor Sampling. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 306–336.

[124] T. Plantard, W. Susilo, K. Than Win. A Digital Signature Scheme Based on CVP_{∞}. Public Key Cryptography. Vol. 4939. Lecture Notes in Computer Science. Springer, 2008, pp. 288–307.

[125] T. Pöppelmann, E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. de la Piedra, P. Schwabe, D. Stebila. NewHope algorithm specifications and supporting documentation Version 1.0. https://newhopecrypto.org/data/NewHope_2017_12_21.pdf. 2017-11. (2024-03-04 閲覧).

[126] T. Pöppelmann et al. NewHope algorithm specifications and supporting documentation Version 1.02 (Updated March 15, 2019). https://newhopecrypto.org/data/NewHope_2019_04_10.pdf. 2019-03. (2024-03-04 閲覧).

[127] E. W. Postlethwaite, F. Virdia. On the Success Probability of Solving Unique SVP via BKZ. Public Key Cryptography (1). Vol. 12710. Lecture Notes in Computer Science. Springer, 2021, pp. 68–98.

[128] A. Pouly, Y. Shen. Provable Dual Attacks on Learning with Errors. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 256–285.

[129] T. Prest. FALCON Update (2024). https://csrc.nist.gov/csrc/media/Presentations/2024/falcon/images-media/prest-falcon-pqc2024.pdf. 2024-04. (2024-12-30 閲覧).

[130] The Chromium Projects. CECPQ2. https://www.chromium.org/cecpq2/. (2024-03-04 閲覧).

[131] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. STOC. ACM, 2005, pp. 84–93.

[132] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. J. ACM. Vol. 56, Num. 6 (2009), 34:1–34:40.

[133] O. Regev. The Learning with Errors Problem (Invited Survey). CCC. IEEE Computer Society, 2010, pp. 191–204.

[134] M. Rosca, A. Sakzad, D. Stehlé, R. Steinfeld. Middle-Product Learning with Errors. CRYPTO (3). Vol. 10403. Lecture Notes in Computer Science. Springer, 2017, pp. 283–297.

[135] K. Ryan, N. Heninger. Fast Practical Lattice Reduction Through Iterated Compression. CRYPTO (3). Vol. 14083. Lecture Notes in Computer Science. Springer, 2023, pp. 3–36.

[136] Open Quantum Safe. Algorithms in liboq. https://openquantumsafe.org/liboqs/algorithms/. (2024-03-04 閲覧).

[137] T. Saito, K. Xagawa, T. Yamakawa. Tightly-Secure Key-Encapsulation Mechanism in the Quantum Random Oracle Model. EUROCRYPT (3). Vol. 10822. Lecture Notes in Computer Science. Springer, 2018, pp. 520–551.

[138] C. Saliba. Error correction and reconciliation techniques for lattice-based key generation protocols. Ph. D. Theses. CY Cergy Paris Université, 2022-05. https://theses.hal.science/tel-03718212/file/SALIBA_2022.pdf.

[139] C. Saliba, L. Luzzi, C. Ling. Error Correction for FrodoKEM Using the Gosset Lattice. 2021.

[140] C.-P. Schnorr, M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Math. Program. Vol. 66 (1994), pp. 181–199.

[141] Federal office for information security. BSI – Technical guideline (Cryptographic mechanisms: Recommendations and key lengths). https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile&v=10. 2023-01. (2024-03-04 閲覧).

[142] Selected Algorithms 2022. 2022-07. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. (2024-03-04 閲覧).

[143] E.-Y. Seo, Y.-S. Kim, J.-W. Lee, J.-S. No. Peregrine: Toward Fastest FALCON Based on GPV Framework. (2022). https://eprint.iacr.org/2022/1495.

[144] H. M. Steffen, G. Land, L. J. Kogelheide, T. Güneysu. Breaking and Protecting the Crystal: Side-Channel Analysis of Dilithium in Hardware. PQCrypto. Vol. 14154. Lecture Notes in Computer Science. Springer, 2023, pp. 688–711.

[145] D. Stehlé, R. Steinfeld. Making NTRU as Secure as Worst-Case Problems over Ideal Lattices. EUROCRYPT. Vol. 6632. Lecture Notes in Computer Science. Springer, 2011, pp. 27–47.

[146] D. Stehlé, R. Steinfeld. Making NTRUEncrypt and NTRUSign as Secure as Standard Worst-Case Problems over Ideal Lattices. Cryptology ePrint Archive, Paper 2013/004. 2013. https://eprint.iacr.org/2013/004.

[147] D. Stehlé, R. Steinfeld, K. Tanaka, K. Xagawa. Efficient Public Key Encryption Based on Ideal Lattices. ASIACRYPT. Vol. 5912. Lecture Notes in Computer Science. Springer, 2009, pp. 617–635.

[148] M. J. Steiner. The Complexity of Algebraic Algorithms for LWE. EUROCRYPT (3). Vol. 14653. Lecture Notes in Computer Science. Springer, 2024, pp. 375–403.

[149] L. Wang. Analyzing Pump and Jump BKZ Algorithm Using Dynamical Systems. PQCrypto (1). Vol. 14771. Lecture Notes in Computer Science. Springer, 2024, pp. 406–432.

[150] R. Wang, M. Brisfors, E. Dubrova. A Side-Channel Attack on a Higher-Order Masked CRYSTALS-Kyber Implementation. ACNS (3). Vol. 14585. Lecture Notes in Computer Science. Springer, 2024, pp. 301–324.

[151] Z. Wang, Q. Lai, F.-H. Liu. Ring/Module Learning with Errors Under Linear Leakage – Hardness and Applications. Public Key Cryptography (2). Vol. 14602. Lecture Notes in Computer Science. Springer, 2024, pp. 275–304.

[152] H. Wee, D. J. Wu. Succinct Vector, Polynomial, and Functional Commitments from Lattices. EUROCRYPT (3). Vol. 14006. Lecture Notes in Computer Science. Springer, 2023, pp. 385–416.

[153] wolfSSL. wolfSSL support for Apache httpd and curl (Post-Quantum Edition). `https://github.com/wolfSSL/osp/blob/master/apache-httpd/README_post_quantum.md`. (2024-03-04 閲覧).

[154] W. Xia, L. Wang, G. Wang, D. Gu, B. Wang. A Refined Hardness Estimation of LWE in Two-Step Mode. Public Key Cryptography (3). Vol. 14603. Lecture Notes in Computer Science. Springer, 2024, pp. 3–35.

[155] W. Xia, L. Wang, G. Wang, D. Gu, B. Wang. Refined Strategy for Solving LWE in Two-step Mode. Cryptology ePrint Archive, Paper 2022/1343. 2022. `https://eprint.iacr.org/2022/1343`.

[156] Y. Yu, L. Ducas. Second Order Statistical Behavior of LLL and BKZ. SAC. Vol. 10719. Lecture Notes in Computer Science. Springer, 2017, pp. 3–22.

[157] Y. Yu, H. Jia, X. Wang. Compact Lattice Gadget and Its Applications to Hash-and-Sign Signatures. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 390–420.

[158] 量子耐性暗号研究団. KpqC. `https://kpqc.or.kr/`. (2024-12-06 閲覧).

第 4 章

符号に基づく暗号技術

本章では符号に基づく暗号技術についてまとめる。符号に基づく暗号技術の安全性はシンドローム復号（Syndrome Decoding: SD）問題や Learning Parity with Noise: LPN 問題を解く計算の困難性に依存している。

■準備: 本章で使用する記号・用語を以下にまとめる。以下では，q を素数 p のべきとする。すなわち，ある正整数 k が存在して $q = p^k$ である。以下では log の底が省略されている場合は底を 2 とする。自然対数を用いる場合は ln と書く。

有限体: $\mathbb{F}_q$ で位数が q の有限体を表す。

ハミング重みとハミング距離: V_n を有限体 $\mathbb{F}_q$ 上の n 次元ベクトル空間とする。

- 行ベクトル $\boldsymbol{v} = (v_1, v_2, \ldots, v_n) \in V_n$ のハミング重みとは，非ゼロの成分の数である。すなわち，有限集合 X に対して $|X|$ で X の要素数を表すとき，$\mathsf{HW}(\boldsymbol{v}) = |\{i \mid v_i \neq 0\}|$ である。
- ハミング距離を $d_H(\boldsymbol{x}, \boldsymbol{y}) = \mathsf{HW}(\boldsymbol{x} - \boldsymbol{y})$ で定義する。
- $\mathcal{S}_H(n, w)$ でハミング重みが w の n 次元ベクトル全体の集合を表す。
- $\mathcal{S}_H^{\leq}(n, w)$ でハミング重みが w 以下の n 次元ベクトル全体の集合を表す。

■線形符号: 線形符号とは，誤りが発生する通信路において，メッセージを相手に正しく伝えるための技術である。メッセージを冗長にして（符号化という）送信し，受信時に伝送中に生じた誤りを訂正する（復号という）ことで，正しいメッセージを得ることができる。自然数 n および 素数べき q について，$\mathbb{F}_q$ 上の n 次元ベクトル空間の線形部分空間を $\mathbb{F}_q$ 上の線形符号と呼び，$\mathcal{C}$ で表す。$\mathcal{C}$ の要素を符号語と呼び，n を符号長という。このとき $[\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k]$ を $\mathcal{C}$ の基底とする。k を線形符号の次元と呼ぶ。$\mathbb{F}_q$ 上の線形符号の符号長が n，次元が k であるとき，$[n, k]_q$ -線形符号とよぶ。$[n, k]_q$ -線形符号 $\mathcal{C}$ の最小距離 d とは，2 つの異なる符号語間のハミング距離の最小値である。d は符号 $\mathcal{C}$ の全ての非ゼロ符号語のハミング重みの最小値に一致する。すなわち，$d = \min_{\boldsymbol{c} \in \mathcal{C} \setminus \{\boldsymbol{0}\}} \mathsf{HW}(\boldsymbol{c})$ である。

$[n, k]_q$ -線形符号 $\mathcal{C}$ の生成行列とは，符号 $\mathcal{C}$ の基底ベクトルを行とする行列 $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$ であり，メッセージの符号化に用いられる。メッセージ $\boldsymbol{s} \in \mathbb{F}_q^k$ に対して，$\boldsymbol{sG} \in \mathbb{F}_q^n$ は符号語である。メッセージと符号語は一対一対応させることができる。$[n, k]_q$ -線形符号 $\mathcal{C}$ のパリティ検査行列とは，行列 $\boldsymbol{H} \in \mathbb{F}_q^{r \times n}$ で，$\boldsymbol{c} \in \mathbb{F}_q^n$ に対して，$\boldsymbol{c} \in \mathcal{C}$ ならばかつその時に限り $\boldsymbol{cH}^\top = \boldsymbol{0}$ となるものである。$\boldsymbol{H}$ の行が一次独立であれば，$r = n - k$ である。組織符号化とは，行列 $\boldsymbol{H}$ に対して，行列 $\boldsymbol{S} \in \mathbb{F}_q^{(n-k) \times (n-k)}$ を適用し，$\boldsymbol{SH} = [\boldsymbol{I}_{n-k} \mid \boldsymbol{Z}]$ を得る操作を指す。ここで，$\boldsymbol{Z} \in \mathbb{F}_q^{(n-k) \times k}$ である。

線形符号は，生成行列やパリティ検査行列をうまく設計することで，受信時に符号語に加えられた誤りを訂正することができる。誤り訂正には，復号アルゴリズムが用いられる。送信する符号語を $\boldsymbol{c}$ とし，通信路上で乗った誤りを $\boldsymbol{e}$ とする。受信者側は，受信語として $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ を得る。受信者は，復号アルゴリズムを用いて $\boldsymbol{y}$ から $\boldsymbol{c}$ を得る。受信者がハミング重み t までの誤り $\boldsymbol{e}$ を一意に訂正できるとき，符号の訂正能力が t であるという。一般に，$t \leq \lfloor (d-1)/2 \rfloor$ が成り立つ。復号アルゴリズムには，符号の構造を用いる方式や，パリティ検査行列を用いる方式がある。後者の場

合，受信語 $\boldsymbol{y}$ に対して $\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^\top$ を計算する。$\boldsymbol{s}$ はシンドロームと呼ばれる。$\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top$ となることから，$\boldsymbol{s} \neq \boldsymbol{0}$ であれば，誤りを検出・訂正できる。

本稿では，具体的な線形符号（リード・ソロモン符号，リード・マラー符号，Goppa 符号）の詳細については扱わない。符号理論の教科書や，電子情報通信学会 知識の森 「1 群 2 編 符号理論」 [56] などを参照されたい。

4.1 符号に基づく暗号技術の安全性の根拠となる問題

本節では，SD 問題と LPN 問題およびその困難性について報告する。

4.1.1 SD 問題とその拡張

4.1.1.1 SD 問題

SD 問題とは，解のハミング重みが指定された条件のもとで，$\mathbb{F}_2$ 上の線形方程式を解けるかどうかという問題である。また，$[n,k]_2$ -線形符号において，パリティ検査行列とシンドローム，受信語に乗ったエラーのハミング重みが与えられたときに，エラーを求める問題とみなすことができる。本問題は符号暗号の安全性の根拠として非常に重要である。実際に， 4.3 節で説明する NIST PQC 標準化プロジェクトの第 4 ラウンドの 3 種類の符号暗号いずれの方式においても，SD 問題が安全性の根拠である。また，近年は SD 問題の拡張問題を安全性の根拠とする暗号や署名方式が多数提案されていることから，主要な問題について説明する。

定義 4.1 (SD 問題) $k, w \leq n$ を満たす正の整数 n, k, w に対して，行列 $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ とベクトル $\boldsymbol{s} \in \mathbb{F}_2^{n-k}$ が与えられる。$\mathsf{SD}_{n,k,w}$ 問題は，$\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$ かつ $\mathsf{HW}(\boldsymbol{e}) = w$ を満たすベクトル $\boldsymbol{e} \in \mathbb{F}_2^n$ を求める問題である。

4.1.1.2 SD 問題の拡張

SD 問題の拡張として，Regular-SD 問題，Restricted-SD 問題，Rank-SD 問題について述べる。

■Regular-SD 問題: ベクトル $\boldsymbol{e} \in \mathbb{F}_2^n$ が w-正則であるとは，$\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_w)$ であり，各 $\boldsymbol{e}_i \in \mathbb{F}_2^{n/w}$ がハミング重み $\mathsf{HW}(\boldsymbol{e}_i) = 1$ を満たすことを指す。Regular-$\mathsf{SD}_{n,k,w}$ 問題は，$\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$ かつ $\mathsf{HW}(\boldsymbol{e}) = w$ を満たす w-正則なベクトル $\boldsymbol{e} \in \mathbb{F}_2^n$ を求める問題である。Carozza，Couteau，Joux [16] によると，Regular-SD 問題に基づく署名は，通常の SD 問題に基づく署名と比較して，署名長や署名時間がより短くなる場合があるとのことである。

■Restricted-SD 問題: Restricted-SD 問題は，体を $\mathbb{F}_2$ から $\mathbb{F}_q$ に変更し，解 $\boldsymbol{e}$ の取りうる値を 0 を含む $\mathbb{F}_q$ の部分集合に制限したものである。

定義 4.2 (Restricted-SD 問題) $k, w \leq n$ を満たす正の整数 n, k, w に対して，行列 $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ とベクトル $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$ が与えられる。部分集合 $\mathbb{E} \subseteq \mathbb{F}_q^*$ に対して，$\mathbb{E}_0 = \mathbb{E} \cup \{0\}$ とする。Restricted-$\mathsf{SD}_{n,k,w}$ 問題は，$\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$ かつ $\mathsf{HW}(\boldsymbol{e}) = w$ を満たすベクトル $\boldsymbol{e} \in \mathbb{E}_0^n$ を求める問題である。

Restricted-SD 問題は，NIST PQC 標準化プロジェクト追加署名第 2 ラウンドの CROSS [7] の安全性の根拠である。

■Rank-SD 問題: Rank-SD 問題は，SD 問題を拡大体 $\mathbb{F}_{q^m}$ に拡張し，重みの指標として行列のランクを用いたものである。$\mathbb{F}_{q^m}$ の $\mathbb{F}_q$ 基底を $(b_1, \ldots, b_m)$ とする。ベクトル $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ を考える。各座標 x_i について，

$$x_i = \sum_{j=1}^{m} x_{i,j} b_j$$

となるようなベクトル $(x_{i,1},\ldots,x_{i,m}) \in \mathbb{F}_q^m$ と関連付けることができる。ベクトル $\boldsymbol{x}$ の関連行列 $\boldsymbol{M}$ を，$\boldsymbol{M} = (x_{i,j})_{(i,j)\in[1,n]\times[1,m]}$ で定義する。このとき，$\boldsymbol{x}$ のランク重み $\mathsf{RW}(\boldsymbol{x})$ を $\mathsf{Rank}(\boldsymbol{M})$ で定義する。

定義 4.3 (Rank-SD 問題) $k, w \leq n$ を満たす正の整数 n, k, w に対して，行列 $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ とベクトル $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$ が与えられる。$\mathsf{Rank\text{-}SD}_{n,k,w}$ 問題は，$\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$ かつ $\mathsf{RW}(\boldsymbol{e}) = w$ を満たすベクトル $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ を求める問題である。

Rank-SD 問題は，NIST PQC 標準化プロジェクト追加署名第 2 ラウンドの RYDE [5] の安全性の根拠である。

4.1.2 SD 問題に対する評価

SD 問題の計算の困難性に関して，Berlekamp，McEliece，van Tilborg [9] によって，NP 困難な 3 次元マッチング問題から SD 問題への多項式時間帰着が示されている。これにより，SD 問題が NP 困難であることが判明している。$\mathsf{SD}_{n,k,w}$ 問題や $[n,k]_2$ -線形符号における k/n は符号化レートと呼ばれており，符号化レートが 1/2 付近で SD 問題が最も難しくなることが知られている。また，符号化レートを増加させると，公開鍵に相当する入力行列 $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ のサイズが減少することから，暗号の設計においては，符号化レート 1/2 以上 1 未満の値が採用されることが多い。

以降では，SD 問題の求解手法として最も研究が進んでいる Information Set Decoding: ISD を取り上げる。ISD は，符号化レート 0.42 以上において既存方式の中で漸近計算量が最も小さく，SD 問題の解読チャレンジを通して実時間での計算量解析が進展している。

近年，Carrier，Debris-Alazard，Meyer-Hilfiger，Tillich [18, 17] が提案した Reduction-to-LPN: R-LPN と呼ばれる手法が，符号化レート 0.42 未満において ISD の漸近計算量を下回るとする解析がなされた。R-LPN は，SD 問題を後述する Sparse-LPN 問題や LPN 問題に帰着して解く手法であるが，実時間での計算量に関して評価が不足しているため，本稿では取り上げない。

4.1.2.1 Information Set Decoding

$\mathsf{SD}_{n,k,w}$ 問題と対応する $[n,k]$-線形符号の最小距離を d と置く。2 進符号の場合，Gilbert-Varshamov 限界により，$k/n \approx 1 - H(d/n)$ である[*1]。$w \approx d$ の場合の $\mathsf{SD}_{n,k,w}$ 問題を Full Distance Decoding と呼ぶ。$w \approx d$ のとき，$\mathsf{SD}_{n,k,w}$ 問題は解くのが最も難しくなる。$w \gg d$ のとき，$\mathsf{SD}_{n,k,w}$ 問題には複数の解が存在することが期待され，$w \leq d$ のとき，$\mathsf{SD}_{n,k,w}$ 問題の解の個数の期待値は 1 以下である。暗号利用においては，$w \ll d$ が選ばれ，トラップドアを通して唯一解が存在するように設計される。以降は $w \leq d$ を考える。

$\mathsf{SD}_{n,k,w}$ 問題を総当りで解くには，ハミング重みが w の n 次元ベクトル $\boldsymbol{e}$ を列挙すればよい。そのため，時間計算量は $O\left(\binom{n}{w}\right)$ となる。より効率的な手法として，Prange は Information Set Decoding と呼ばれる手法 [53] を提案した。基本アイデアは以下である:

1. 一様ランダムに $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ の列ベクトルを入れ替え，$\tilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{P}$ とする。（$\boldsymbol{P} \in \mathbb{F}_2^{n\times n}$ は置換行列。）
2. ガウスの消去法と対応する行列 $\boldsymbol{S} \in \mathbb{F}_2^{(n-k)\times(n-k)}$ によって $\tilde{\boldsymbol{H}}$ を $[\boldsymbol{I}_{n-k} \mid \boldsymbol{Z}] = \boldsymbol{S}\tilde{\boldsymbol{H}}$ とする。（組織符号化）
3. シンドローム $\boldsymbol{s} \in \mathbb{F}_2^{n-k}$ に対して，$\boldsymbol{s}' = \boldsymbol{s}\boldsymbol{S}^\top$ を計算する。
4. $\boldsymbol{s}'$ のハミング重みが w ならば，$\boldsymbol{e} = (\boldsymbol{s}', \boldsymbol{0}_k)\boldsymbol{P}^\top$ を出力する。そうでなければ，1. に戻る。

$\mathsf{HW}(\boldsymbol{s}') = w$ ならば，$\mathsf{HW}(\boldsymbol{e}) = w$ である。また，$\boldsymbol{e}\boldsymbol{H}^\top = (\boldsymbol{s}', \boldsymbol{0}_k)\boldsymbol{P}^\top\boldsymbol{H}^\top = (\boldsymbol{s}', \boldsymbol{0}_k)\tilde{\boldsymbol{H}}^\top = (\boldsymbol{s}, \boldsymbol{0}_k)\boldsymbol{S}^\top\tilde{\boldsymbol{H}}^\top = (\boldsymbol{s}, \boldsymbol{0}_k)[\boldsymbol{I}_{n-k}\boldsymbol{Z}]^\top = \boldsymbol{s}$ が成立する。よって，ステップ 4 のチェックが通るならば，$\boldsymbol{e}$ は SD 問題の解である。末尾 k 列が全て 0 であるような $\boldsymbol{e}\boldsymbol{P}$ は $\binom{n-k}{w}$ 通りあるため，ある置換行列に対して解が出力される確率は $\binom{n-k}{w}/\binom{n}{w}$

[*1] ここで $H(p) = -p\log(p) - (1-p)\log(1-p)$。

表 4.1: 確率 1/2 以上で SD 問題を解く場合の漸近計算量（Full Distance Decoding の場合）

	$\log(\text{Time})/n$	$\log(\text{Space})/n$	備考
Pra62（Lee-Brickel）	0.121	–	[53, 39]
Stern89	0.117	0.0135	[54]
MMT11	0.112	0.0530	[43]; [30] によって空間計算量が改良された
BJMM12	0.102	0.0769	[8]
MO15	0.0967	0.0890	[44]
BM17	0.0953	0.0910	[15]; MO15 を最適化したもの
BM18	0.0951	0.0760	[14]; [18, 25] によって時間・空間計算量が修正された
Sieving ISD	0.101	0.0636	[24]

となる。期待計算量は $\mathrm{poly}(n,k)\cdot O\left(\binom{n}{w}/\binom{n-k}{w}\right)$ となり，先ほどの列挙法よりも速くなる。

Stern [54] 以降，空間計算量を犠牲にすることで時間計算量を引き下げる ISD の改良アルゴリズムが多数提案されている。以下では，Both と May [14] による時間計算量の表を，表 4.1 に示す。この表は，時間計算量を最小化した場合の符号化レート k/n の最悪時（1/2 の少し下）の漸近計算量についてまとめられている。したがって，問題のパラメータによっては，表の数値よりも速く解くことが可能となる。

近年は，漸近計算量のみならず，具体的なパラメータに対する $\mathsf{SD}_{n,k,w}$ 問題を求解するために必要なビット計算量を見積もる研究もなされている。Esser，Verbel，Zweydinger，Bellini [29] は，CryptographicEstimators と呼称する符号暗号や多変数多項式暗号のビット計算量を推定するソフトウェアを開発した。Narisada ら [48] は，Becker らの ISD [8] の実用的な改良方式を提案し，NIST PQC 標準化プロジェクト第 4 ラウンドの 3 種類の符号暗号のビット計算量と実時間の計算量を算出した。

■量子アルゴリズム　現在のところ多項式時間で SD 問題を解く量子アルゴリズムは提案されていない。しかし，量子アルゴリズムを利用して，いくつかの古典 ISD を高速化する方法を Kachigar と Tillich [36] が提案している。[*2]2024 年現在，最良の漸近計算量が得られているのは，BJMM 法 [8] の量子アルゴリズムである量子 BJMM 法であり，時間計算量が $2^{0.0587n}$，空間計算量が $2^{0.0188n}$ となっている。

量子回路設計の研究に関しては，量子 Prange 法に対して，Perriello，Barenghi，Pelosi [51] がグローバーのアルゴリズムを用いた量子回路を提案した。Esser ら [28] は，量子 Prange 法に対して，一部の演算を古典コンピュータ上で行う量子と古典のハイブリッド法を提案した。Perriello，Barenghi，Pelosi [52] は，量子 Prange におけるガウスの消去法の量子回路を改良し，NIST PQC 標準化プロジェクト第 4 ラウンドの 3 種類の符号暗号の解読に必要な量子回路の深さを最大で 2^{30} 削減した。Chevignard，Fouque，Schrottenloher [20] は，量子 Prange 法に対して，量子回路の深さを犠牲にすることで量子ビット数を削減する，深さと幅のトレードオフ手法を提案した。Stern の ISD [54] 以降に提案されたリスト探索を伴う ISD については，グローバーのアルゴリズムと量子ウォーク探索を組み合わせた複雑な量子回路が必要になると考えられている [36]。現在のところ，これらの量子 ISD に対する量子回路は提案されていない。

■現状の進展　格子の場合と同様に “Decoding Challenge”（`https://decodingchallenge.org/`）というウェブサイトが作成された。実時間での計算量解析を通じて，符号に基づく暗号技術の信頼性を向上させることが目的である。現在，以下 5 つのカテゴリが用意されている。

[*2] Kirshanova [38] が Kachigar と Tillich の結果 [36] の改良を提案していたが，誤りがあったことが報告されている。そのため，2024 年時点でのベストな量子アルゴリズムは Kachigar と Tillich [36] であると考えられる。

1. $\mathbb{F}_2$ 係数の一様ランダムな線形符号に対する SD 問題
2. $\mathbb{F}_2$ 係数の一様ランダムな線形符号に対するハミング重みが小さい符号語を探索する問題
3. $\mathbb{F}_3$ 係数の一様ランダム線形符号に対する SD 問題
4. Goppa 符号を用いた Niederreiter 暗号の場合の SD 問題（Classic McEliece に対応 4.3.1）
5. QC-MDPC 符号に基づく SD 問題（BIKE や HQC に対応 4.3.2 4.3.3）

各問題に対して研究および解読が進んでおり，2025 年 1 月現在，解読に成功した最も困難な問題は次のとおりである。

- 1. $n = 570, k = n/2$ に対して $w = 70$（成定，福島，清本，2023/04）
- 2. $n = 1280, k = n/2$ の場合に $w = 204$（成定，岡田，上村，相川，福島，清本，2024/09）
- 3. $n = 200, k = \log_3(n)$ の場合に $w = 198$（Esser，May，Zweydinger，2021/12）
- 4. $n = 1409, k = 0.8n$ に対して $w = 26$（成定，古江，相川，福島，清本，2023/11）
- 5. $n = 3602, k = n/2$ に対して $w = 60$（成定，岡田，上村，相川，福島，清本，2025/1）

1 の結果については，Narisada，Fukushima，Kiyomoto [47] を，4 の結果については，Narisada ら [48] を参照されたい。2，3，5 についての詳細は，Decoding Challenge 上に掲載されている各記録の詳細を参照されたい。

4.1.3 LPN 問題とその拡張

4.1.3.1 LPN 問題

LPN 問題とは，$\mathbb{F}_2$ 上の誤差付きの線形方程式を解けるかどうかという問題である。また，$[n, k]_2$ -線形符号において，生成行列と受信語が与えられたときに，メッセージを復号する問題とみなすことができる。1993 年に，Blum，Furst，Kearns，Lipton [11] が困難と思われる問題として挙げ，定式化を行った。第 3 章において，この問題を一般化した LWE 問題を既に扱っている。

Ber_τ でパラメータ τ のベルヌーイ分布を表すことにする。（確率 τ で 1，確率 $1-\tau$ で 0 となる $\mathbb{F}_2$ 上の分布である。）また，自然数 $k \geq 1$ について，Ber_τ^k で，Ber_τ から独立に k 個サンプルを取ったときの $\mathbb{F}_2^k$ 上の分布を表す。

定義 4.4 (LPN 問題) $\mathbb{F}_2^k$ から一様ランダムに選ばれた秘密鍵 $\boldsymbol{s}$ およびエラー比 $\tau \in [0, 1/2)$ に対して，以下の LPN サンプルを出力する LPN オラクルを考える。

$$(\boldsymbol{a}, b) = (\boldsymbol{a}, \boldsymbol{s} \cdot \boldsymbol{a}^\top + e),$$

ここで，$\boldsymbol{a}$ は $\mathbb{F}_2^k$ から一様ランダムに選び，e は分布 Ber_τ に従い選ぶ。LPN オラクルを n 回呼び出すとき，$(\boldsymbol{A}, \boldsymbol{b}) \leftarrow \mathsf{LPN}_{k,\tau}^n$ と表記する。これは，n 個の LPN サンプル $(\boldsymbol{a}_1, b_1), (\boldsymbol{a}_2, b_2), \ldots, (\boldsymbol{a}_n, b_n)$ を行列・ベクトル表示して，

$$\boldsymbol{A} = [\boldsymbol{a}_1^\top \boldsymbol{a}_2^\top \ldots \boldsymbol{a}_n^\top] \in \mathbb{F}_2^{k \times n}, \qquad \boldsymbol{b} = \boldsymbol{s}\boldsymbol{A} + \boldsymbol{e} \in \mathbb{F}_2^n$$

としたものである。n をサンプル数と呼ぶ。$\boldsymbol{e}$ は，n 個の LPN サンプルのエラー e を成分とするベクトルである。$\mathsf{LPN}_{k,\tau}$ 問題とは，LPN オラクルへのアクセスが可能なときに，$\boldsymbol{s}$ を求める問題である。

サンプル数が n の $\mathsf{LPN}_{k,\tau}$ 問題は，$\mathsf{SD}_{n,k,n\tau}$ 問題に変換することができる。変種として，体を $\mathbb{F}_q$ に変更した LPN 問題・仮定が用いられることもある。LPN 問題の安全性仮定について詳しく知りたい方は，2022 年度版の CRYPTREC 耐量子計算機暗号の研究動向調査報告書 [1, Section 3.1] を参照して欲しい。

4.1.3.2 LPN 問題の拡張

以下では LPN 問題の拡張について述べる。

■Exact-LPN 問題　誤差分布として，$\boldsymbol{e} \leftarrow \mathsf{Ber}_\tau^n$ ではなく，ハミング重みが丁度 w のものだけを考える（すなわち $\boldsymbol{e} \leftarrow \mathcal{S}_H(n,w)$）。このように誤差分布を変えた問題を Exact-LPN 問題と呼ぶ。

■Sparse-LPN 問題　一部の暗号方式では，$\boldsymbol{s}$ のハミング重みが小さい，すなわち，疎（スパース，sparse）であることを要求する。Applebaum ら [3] は $\boldsymbol{s}$ を誤差分布である χ^k から選んだ場合の LPN 問題と $\boldsymbol{s}$ を $\mathbb{F}^k$ からランダムに選んだ場合の問題とが等価であることを示している。このように $\boldsymbol{s}$ の分布を変えた問題を Sparse-LPN 問題と呼ぶ。

■Ring-LPN 問題　Heyse，Kiltz，Lyubashevsky，Paar，Pietrzak [35] は，Ring-LPN 問題を定義した。この問題は Ring-LWE 問題（3.1.1.1 節）と同様に定義される。

定義 4.5 (探索版 Ring-LPN 問題) 適当な k 次の $\mathbb{F}_q$ 係数多項式 $f(x)$ を考え，環 $R_q = \mathbb{F}_q[x]/(f(x))$ を固定する。R_q 上の確率分布 χ を固定する。

R_q 上の誤差分布 χ および $s \in R_q$ について，オラクル $\mathcal{O}_{s,\chi}$ を以下で定義する。(1) a を R_q から一様ランダムに選び，(2) e を分布 χ に従い選び，(3) $b = sa + e$ と計算し，(4) $(a,b) \in R_q^2$ を出力する。

探索版 Ring-LPN 問題とは，オラクル $\mathcal{O}_{s,\chi}$ へのアクセスが可能なときに，$s \in R_q$ を求める問題である。

■Module-LPN 問題　Module-LWE 問題（3.1.1.1 節）と同様に，Module-LPN 問題を定義することができる。

定義 4.6 (探索版 Module-LPN 問題) 適当な k_0 次の $\mathbb{F}_q$ 係数多項式 $f(x)$ を考え，環 $R_q = \mathbb{F}_q[x]/(f(x))$ を固定する。R_q 上の確率分布 χ を固定する。

R_q 上の誤差分布 χ および $\boldsymbol{s} \in R_q^k$ について，オラクル $\mathcal{O}_{\boldsymbol{s},\chi}$ を以下で定義する。(1) $\boldsymbol{a}$ を R_q^k から一様ランダムに選び，(2) e を分布 χ に従い選び，(3) $b = \boldsymbol{s} \cdot \boldsymbol{a}^\top + e$ と計算し，(4) $(\boldsymbol{a}, b) \in R_q^{(k+1)}$ を出力する。

探索版 Module-LPN 問題とは，オラクル $\mathcal{O}_{\boldsymbol{s},\chi}$ へのアクセスが可能なときに，$\boldsymbol{s} \in R_q^k$ を求める問題である。

4.1.4 LPN 問題に対する評価

LPN 問題の計算の困難性に関して，サンプル数を固定した場合，NP 困難になることが Berlekamp，McEliece，van Tilborg [9] によって示されている[*3]。また，Håstad [34] により近似版 LPN 問題[*4]の NP 困難性も示されている。しかし，平均時の困難性についてはよく分かっていない。

LPN 問題の古典求解手法として，現在，大別して以下の 5 つのアルゴリズムが知られている。

1. ガウスの消去法に基づく手法 [19]
2. SD 問題における Information Set Decoding に基づく手法 [27]
3. Blum，Kalai，Wasserman [12] の BKW アルゴリズムに基づく手法
4. Arora，Ge [6] の「再線形化」アルゴリズム
5. 2. と 3. を組み合わせたハイブリッド法 [27]

このうち，著者の知る限り漸近的に時間計算量が最も小さい手法は BKW アルゴリズムであり，実用上最も高速な手法はハイブリッド法である。以降で各手法の概要を説明する。

[*3] $\boldsymbol{A}$ および $\boldsymbol{b}$ を与えられたときに，線形方程式 $\boldsymbol{s}\boldsymbol{a}_i^\top = b_i$ を満たす数を最大化する $\boldsymbol{s}$ を探索する問題を考える。
[*4] $\boldsymbol{A}$ および $\boldsymbol{b}$ を与えられたときに，線形方程式 $\boldsymbol{s}\boldsymbol{a}_i^\top = b_i$ を近似度×最大値以上満たす $\boldsymbol{s}$ を探索する問題。

4.1.4.1 ガウスの消去法に基づく手法

ガウスの消去法に基づく手法は，2008 年に Carrijo，Tonicelli，Imai，Nascimento [19] によって初めて提案された LPN 問題に対する多項式空間・指数時間アルゴリズムである。この手法は指数回数の LPN オラクルの呼び出しが必要であるが，k 個の LPN サンプルを格納するメモリがあれば良いので，必要な計算資源が少なく，実装が容易である。アルゴリズムの概要は以下である。

1. LPN オラクルを k 回呼び出す: $(\boldsymbol{A}, \boldsymbol{b}) \leftarrow \mathsf{LPN}_{k,\tau}^{k}$
2. $\boldsymbol{A} \in \mathbb{F}_2^{k \times k}$ が可逆行列なら，$\boldsymbol{s}' = \boldsymbol{b}\boldsymbol{A}^{-1}$ を秘密鍵の候補とする。
3. LPN オラクルを $m = O(k)$ 回呼び出す: $(\boldsymbol{A}', \boldsymbol{b}') \leftarrow \mathsf{LPN}_{k,\tau}^{m}$
4. 閾値 $c \geq m\tau$ に対して，$\mathsf{HW}(\boldsymbol{s}'\boldsymbol{A}' + \boldsymbol{b}') \leq c$ なら，$\boldsymbol{s}'$ を解として出力する。それ以外の場合，1. からやり直す。

本アルゴリズムは，$\boldsymbol{b}$ にエラーが含まれている時とそうでない時のハミング重みの違いを用いて，秘密鍵を抽出する。エラーが全く含まれていない場合，$\boldsymbol{s}'$ は秘密鍵である。このとき，$\mathsf{HW}(\boldsymbol{s}'\boldsymbol{A}' + \boldsymbol{b}')$ の分布は $\mathsf{Ber}_{m,\tau}$ に従う。一方，エラーが含まれているとき，$\boldsymbol{s}'$ は秘密鍵ではない。このとき，$\mathsf{HW}(\boldsymbol{s}'\boldsymbol{A}' + \boldsymbol{b}')$ の分布は $\mathsf{Ber}_{m,1/2}$ に従う。この分布の違いを使えば，適切にパラメータ m, c を設定することで，高い確率で秘密鍵のみを出力できる。本手法の計算量は，k 個の LPN サンプルのエラーが全て 0 である確率が $(1-\tau)^k$ であることから，時間計算量が $\mathrm{poly}(k) \cdot O(2^k)$，空間計算量が $O(k^2)$，サンプル数が $n = O(2^k)$ となる。

4.1.4.2 Information Set Decoding に基づく手法

$\mathsf{LPN}_{k,\tau}$ は任意のサンプル数 n に対して SD 問題（$\mathsf{SD}_{n,k,\tau n}$）に変換できるため，LPN 問題は SD 問題の効率的な求解手法である Information Set Decoding を用いて解くことができる。Esser，Kübler，May [27] によって提案された実用的なアルゴリズムは，ガウスの消去法に基づく手法を拡張したものである。基本アイデアとして，ガウスの消去法に基づく手法は k 個の LPN サンプルのエラーが全て 0 の場合を考えるが，その拡張として n 個の LPN サンプルのうち k 個の LPN サンプルのエラーが全て 0 となる組み合わせを考える。$n = O(k^2)$ に設定することで，高い確率でこのような組み合わせが存在する。ガウスの消去法に基づく手法と比較して，LPN サンプルの数を $O(2^k)$ から $O(k^2)$ まで減らせる点が利点である。アルゴリズムの概要は以下である。

1. LPN オラクルを $n+m$ 回呼び出す: $(\boldsymbol{A}, \boldsymbol{b}) \leftarrow \mathsf{LPN}_{k,\tau}^{n+m}$
2. 集合 $[n] = \{1, \ldots, n\}$ から一様ランダムに k 要素を抽出した部分集合 I を 1 つ選ぶ: $I \leftarrow \mathcal{U}(\binom{[n]}{k})$
3. $(\boldsymbol{A}, \boldsymbol{b})$ から集合 I の要素に対応する k 列を抽出したサンプル $(\boldsymbol{A}_I, \boldsymbol{b}_I)$ に対して，$\boldsymbol{A}_I \in \mathbb{F}_2^{k \times k}$ が可逆行列なら，$\boldsymbol{s}' = \boldsymbol{b}_I \boldsymbol{A}_I^{-1}$ を秘密鍵の候補とする。
4. $(\boldsymbol{A}', \boldsymbol{b}') \leftarrow \mathsf{LPN}_{k,\tau}^{m}$ と閾値 $c \geq m\tau$ に対して，$\mathsf{HW}(\boldsymbol{s}'\boldsymbol{A}' + \boldsymbol{b}') \leq c$ なら，$\boldsymbol{s}'$ を解として出力する。それ以外の場合，2. からやり直す。

この手法のステップ 2 以降では，$\mathsf{LPN}_{k,\tau}^{n}$ 問題に対応する $\mathsf{SD}_{n,k,\tau n}$ 問題を考える。Information Set Decoding における Prange 法 [53] を使い，$\mathsf{SD}_{n,k,\tau n}$ 問題を解く [27]。Prange 法は，$\mathsf{HW}(\boldsymbol{e}_I) = 0$ ならば，$\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{s}\boldsymbol{A}$ かつ $\mathsf{HW}(\boldsymbol{e}) = \tau n$ である $\boldsymbol{e} \in \mathbb{F}_2^n$ を出力する。集合 I は，ISD における Information Set と呼ばれる集合に対応する。上記のアルゴリズムは，任意の ISD に一般化できる:

1. LPN オラクルを $n+m$ 回呼び出す: $(\boldsymbol{A}, \boldsymbol{b}) \leftarrow \mathsf{LPN}_{k,\tau}^{n+m}$
2. $\mathsf{LPN}_{k,\tau}^{n}$ に対して，ISD を使って $\mathsf{SD}_{n,k,\tau n}$ を解く。出力として，解 $\boldsymbol{e} \in \mathbb{F}_2^n$ と Information Set I を得る。なお，残りの m サンプル $\mathsf{LPN}_{k,\tau}^{m}$ は ISD の内部で解の検証に用いられる。
3. $(\boldsymbol{A}_I, \boldsymbol{b}_I)$ に対して，$\boldsymbol{s} = (\boldsymbol{b}_I - \boldsymbol{e}_I)\boldsymbol{A}_I^{-1}$ は高い確率で解である。

ISD には様々な手法があるが，[27] において，MMT 法が実用上最適であることが示されている。MMT 法のような Prange 法より後に提案された ISD は，$\mathsf{HW}(\boldsymbol{e}_I) \leq p$ ならば，$\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{sA}$ かつ $\mathsf{HW}(\boldsymbol{e}) = \tau n$ である $\boldsymbol{e} \in \mathbb{F}_2^n$ を出力する。よって，Information Set にエラーが含まれている場合があるため，ステップ 3 でエラーを打ち消す必要がある。

本手法の計算量は，ISD の手法によって異なる関数 $c(\tau)$ に対して，時間計算量が $2^{c(\tau)k}$，空間計算量が $O(k^3)$，サンプル数が $O(k^2)$ となる。

4.1.4.3 BKW アルゴリズムに基づく手法

BKW アルゴリズム [12] は，LPN 問題に対する最も著名な手法である。基本アイデアは以下である。オラクルからのサンプル $(\boldsymbol{a}, b)$ が $\boldsymbol{a} = (1, 0, \ldots, 0)$ という形であれば，$b = s_1 + e$ となる。このようなサンプルを大量に集めれば，s_1 を多数決法で求めることが出来る。一般に $\boldsymbol{u}_j$ を j 番目の単位ベクトルとして，$(\boldsymbol{u}_j, b)$ という形のサンプルを集めれば s_j を多数決法で求められる。そこで，LPN オラクルからのサンプルを用いて，このようなサンプルを生成することを目指す。s_1 を求める BKW アルゴリズムの概要は以下である。なお，$s_2, \ldots, s_k$ についても同様に求められる。

1. LPN オラクルを $N = 2^{O(k/\log k)}$ 回呼び出す。
2. $\boldsymbol{a} \in \mathbb{F}_2^k$ を長さ $\ell = k/\log k$ の $t = \log k$ 個のブロックに分割する。
3. $i = 1, \ldots, t-1$ および各 $j \in \mathbb{F}_2^\ell$ について，$\boldsymbol{a}$ の接尾辞 $j|0^{(i-1)d}$ の j 毎に，全サンプルに対して 2^ℓ 個のバケツに分類する。
4. 各バケツ内で代表 $(\boldsymbol{a}^*, b^*)$ を 1 つ選び，他のサンプル $(\boldsymbol{a}, b)$ に対して $(\boldsymbol{a} + \boldsymbol{a}^*, b + b^*)$ で置換する。代表 $(\boldsymbol{a}^*, b^*)$ はバケツから除去する。これにより，$\boldsymbol{a}$ の末尾 i ブロックが全て 0 となるサンプルが得られる。
5. $\boldsymbol{a} = (a_1, \ldots, a_\ell, 0, \ldots, 0)$ であるサンプル集合に対してガウスの消去法を行い，$\boldsymbol{a} = (1, 0, \ldots, 0)$ を得る。いま，$b = \boldsymbol{s} \cdot \boldsymbol{a}^\top + e$ に対して，$b = s_1 + e$ である。
6. b について多数決を行い，多い方を s_1 とする。

BKW アルゴリズムでは，ステップ 4 で LPN サンプル同士の加算を実施することに起因してノイズが増加するため，ステップ 5 の e の分布は Ber_τ とは異なる点に注意されたい。

BKW アルゴリズムの計算量は，時間計算量・空間計算量・サンプル数いずれも $2^{O(k/\log k)}$ である。よって，大きな次元の LPN 問題に対しては，メモリ量の増加が課題となる。後述するように，良好なタイムメモリトレードオフ（時間計算量と空間計算量のトレードオフ）を持つ BKW アルゴリズムの研究開発も進められている。

■LF アルゴリズム: Levieil と Fouque [40] は BKW アルゴリズムの一部を変更し LF1 アルゴリズムを提案した。変更点は，BKW アルゴリズムの 5 行目において，$\boldsymbol{a} = (a_1, \ldots, a_\ell, 0, \ldots, 0)$ から $s_1, \ldots, s_\ell$ を総当りで計算することである。

Levieil と Fouque は，LF1 アルゴリズムに一部のヒューリスティクを組み合わせた LF2 アルゴリズムも提案している。報告によれば，$k = 99, \tau = 1/4, n = 10000$ の LPN 問題を CPU: Pentium 4（3GHz），RAM: 1GB のマシンで解くことが可能である。Devadas，Ren，Xiao [22] は LF2 アルゴリズムについて詳細な解析を与え，BKW アルゴリズムとの比較を行っている。Devadas らの報告によれば，LF2 アルゴリズムは BKW アルゴリズムより時間計算量が少ない分，メモリ使用量が増加するとのことである。

■Kirchner の手法: Kirchner [37] は一様ランダムに選ばれた $\boldsymbol{s}$ よりは Ber_τ に従って選ばれる誤りベクトル $\boldsymbol{e}$ の方が，ハミング重みが小さく，取りうる値が少ないことに着目した。そこで，LPN 問題を Sparse-LPN 問題に置き換えた上で問題を解く手法を提案している。一般の $\boldsymbol{s}$ であれば，総当りに必要な回数は 2^ℓ となる。一方，$\boldsymbol{e}$ は疎であることが期待されるため，$\boldsymbol{e}$ の総当りに必要な回数が削減される。

■Ring-LPN 問題への応用: Bernstein と Lange [10] は Levieil と Fouque の高速化手法および Kirchner のアイデアを用いることにより，Ring-LPN 問題の解法が高速化できることを示している。

■Coded-BKW: Guo，Johansson，Löndahl [33] は，covering codes と呼ばれる符号を用いて，Kirchner の手法 [37] および Bernstein と Lange [10] の手法を改良した Coded-BKW を提案した。Kirchner の手法では，BKW アルゴリズムのステップ 5 において，$\boldsymbol{a}$ を covering code の受信語とみなすことで探索空間の圧縮を行い，高速化を行っている。[*5]

Zhang，Jiao，Wang [55] は別の符号を用いて Coded-BKW を改良している。

Bogos と Vaudenay [13] は Coded-BKW の解析が一部欠けていることを分析し，最適化を行いつつ詳細な計算量評価を与えた。

■Dissection-BKW: Esser，Heuer，Kübler，May，Sohler [26] は BKW アルゴリズムのタイムメモリトレードオフ手法である Dissection-BKW を提案した。Dissection-BKW は，Dinur，Dunkelman，Keller，Shamir [23] によって提案された部分和問題に対するタイムメモリトレードオフ手法である Dissection を，BKW アルゴリズムに適用したものである。

4.1.4.4 Arora–Ge アルゴリズム

Arora と Ge [6] は多変数多項式問題で古くから用いられている再線形化と呼ばれる手法を用いて，LPN 問題を解くことを考えた。このアルゴリズムをサンプル数 n の $\mathsf{LPN}_{k,\tau}$ 問題に用いた場合，$w = \tau n$ として，$\mathrm{poly}(k^w)$ 時間で解くことができる。$\mathrm{poly}(k^w) = 2^{O(\tau n \log k)}$ であるから，$\tau = o(k/(n\log^2 k))$ のようにエラーが疎であれば，BKW アルゴリズムよりも効率が良い。実際の符号暗号のパラメータ設定では，エラーをこのように疎に設定することはないため，暗号の攻撃アルゴリズムとして用いるには重要度が低い。

4.1.4.5 Information Set Decoding と BKW を組み合わせたハイブリッド法

Esser，Kübler，May [27] は，BKW アルゴリズムと Information Set Decoding を組み合わせた実用上高速な手法を提案した。ハイブリッド法のアルゴリズムの概要は以下である。

1. 次元削減パラメータ $k_1, k_2, k' = k - k_1 - k_2$ を決定する。
2. $\boldsymbol{a}$ の末尾 k_1 列が 0 である LPN サンプルを一定数集める。
3. 集められたサンプルの k_2 列に対して BKW アルゴリズムを行い，$\boldsymbol{a}$ の末尾 $k_1 + k_2$ 列が 0 となるサンプルを一定数集める。これによって，$s_{k'+1}, \ldots, s_{k'+k_2}$ が求まる。
4. 集められたサンプルからなる LPN 問題を SD 問題に変換し，ISD で解く。これによって，$s_1, \ldots, s_{k'}$ が求まる。
5. 残りの $s_{k'+k_2+1}, \ldots, s_k$ を求める。k_1 の値の大きさに応じて，全探索・ISD・再度ステップ 1 から処理をやり直すといった方策を取る。

本手法の計算量は，Information Set Decoding に基づく手法の計算量と BKW アルゴリズムに基づく手法の計算量との中間である。

報告によれば，$k = 135, \tau = 1/4$ の LPN 問題に対して，$k_1 = 10, k_2 = 99, k' = 26$ のパラメータを使用し，16 コアの CPU および 256GB の RAM を搭載したサーバ 1 台を用いて，5.69 日での求解に成功した。また，$k = 243, \tau = 1/8$ の LPN 問題に対して，$k_1 = 35, k_2 = 0, k' = 208$ のパラメータを使用し，同じサーバ 1 台を用いて，15.07 日での求解に成功した。

[*5] ただし，国際会議でのプレゼンテーションではサンプル数が不足していたとの報告があり，計算量・メモリ・サンプル数の評価は見直されている。詳しくは，[55] および [13] を参照のこと

また，暗号設計に用いられるパラメータを持つ LPN 問題に対して，空間計算量を現実的な値にセキュリティマージンを加えたものに制限（2^{60}bit $=$ 128PB および 2^{80}bit $=$ 128ZB）した時のビット計算量が推定された。報告によれば，$k = 512, \tau = 1/8$ の LPN 問題に対するハイブリッド法のビット計算量は 2^{102} であり，$k = 512, \tau = 1/4$ の LPN 問題に対するビット計算量は 2^{151} である。一方で，この空間計算量の範囲では Coded-BKW [33] は動作しないということである。

4.1.4.6 量子アルゴリズム

現在のところ，多項式時間で LPN 問題を解く量子アルゴリズムは提案されていない。Esser，Kübler，May [27] は，上述する ISD に基づく手法やハイブリッド法に対して，グローバーのアルゴリズムや量子ウォーク探索を用いることで高速化できる点を指摘している。$k = 512, \tau = 1/8$ の LPN 問題に対する量子ハイブリッド法のビット計算量は 2^{69} であり，$k = 512, \tau = 1/4$ の LPN 問題に対するビット計算量は 2^{112} と推定されている。

4.2 符号に基づく代表的な暗号方式

本節では，符号に基づく代表的な暗号方式と署名方式の説明を行う。以下では，$\mathrm{GL}_k(\mathbb{F}_q)$ で k 次の $\mathbb{F}_q$ 要素正則行列全体がなす群を表す。また，S_n で n 次対称群を表す。S_n の要素である置換を $\mathrm{GL}_n(\mathbb{F}_q)$ 中の置換行列と同一視することとする。

4.2.1 McEliece 暗号

McEliece [45] が提案した古典的な暗号方式である。以下では $q = 2$ とする。

- k：安全性パラメータ
- n：サンプルの個数
- τ：誤差パラメータ（例: $\tau n = O(k)$）
- t：線形符号の誤り訂正能力（$t = \Omega(\tau n)$）

鍵生成: 誤り訂正能力が t である $[n, k]_2$-線形符号の生成行列 $\boldsymbol{G} \in \mathbb{F}_2^{k \times n}$ を生成する。$\boldsymbol{S} \leftarrow \mathrm{GL}_k(\mathbb{F}_2)$ を一様ランダムに選ぶ。$\boldsymbol{P} \leftarrow S_n$ を一様ランダムに選ぶ。$\tilde{\boldsymbol{G}} = \boldsymbol{SGP}$ とする。
公開鍵を $\tilde{\boldsymbol{G}}$ とし，秘密鍵を $(\boldsymbol{S}, \boldsymbol{G}, \boldsymbol{P})$ とする。
暗号化: 平文を $\boldsymbol{m} \in \mathbb{F}_2^k$ とする。乱数 $\boldsymbol{e} \leftarrow \mathsf{Ber}_\tau^n$ を選び，暗号文 $\boldsymbol{c} = \boldsymbol{m}\tilde{\boldsymbol{G}} + \boldsymbol{e} \in \mathbb{F}_2^n$ を計算する。
復号: $\hat{\boldsymbol{v}} = \boldsymbol{cP}^{-1}$ を計算する。$\hat{\boldsymbol{v}}$ を線形符号で訂正し $\boldsymbol{m}' \in \mathbb{F}_2^k$ を得る。$\boldsymbol{m} = \boldsymbol{m}'\boldsymbol{S}^{-1}$ を出力する。

復号の正当性は以下で確認される。$\boldsymbol{c} = \boldsymbol{m}\tilde{\boldsymbol{G}} + \boldsymbol{e}$ として，$\hat{\boldsymbol{v}} = \boldsymbol{cP}^{-1}$ を計算すると，

$$\hat{\boldsymbol{v}} = \boldsymbol{m}\tilde{\boldsymbol{G}}\boldsymbol{P}^{-1} + \boldsymbol{e}\boldsymbol{P}^{-1} = \boldsymbol{mSG} + \boldsymbol{e}\boldsymbol{P}^{-1}$$

を得る。$\boldsymbol{mSG}$ はランダム化されたメッセージ $\boldsymbol{mS}$ の符号語であり，$\boldsymbol{e}\boldsymbol{P}^{-1}$ は誤りである。$\boldsymbol{e}\boldsymbol{P}^{-1}$ のハミング重みが t 以下であれば，線形符号の復号により，$\boldsymbol{m}' = \boldsymbol{mS}$ を得る。よって，高い確率で復号に成功する。平文 $\boldsymbol{m}$ および生成行列 $\tilde{\boldsymbol{G}}$ が一様ランダムであれば，暗号文 $\boldsymbol{c} \in \mathbb{F}_2^n$ はランダムな n 次元のベクトルと見分けが付かないと考えられている。一方で，平文 $\boldsymbol{m}$ が零ベクトルのとき，暗号文はランダムなベクトルと区別されてしまう。このことから，オリジナルの McEliece 暗号にはセキュリティ上の課題が存在することがわかる。

4.2.2 Niederreiter 暗号

Niederreiter [49] が 1986 年に提案した。のちに McEliece 暗号と安全性が等価であることが示された。詳しくは [41] を参照のこと。以下では $q = 2$ とする。

- k：安全性パラメータ
- n：サンプルの個数
- t：線形符号の誤り訂正能力

鍵生成: 誤り訂正能力が t である $[n, k]_2$ -線形符号のパリティ検査行列 $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ を生成する。$\boldsymbol{T} \leftarrow \mathrm{GL}_{n-k}(\mathbb{F}_2)$ を一様ランダムに選ぶ。$\boldsymbol{Q} \leftarrow S_n$ を一様ランダムに選ぶ。$\tilde{\boldsymbol{H}} = \boldsymbol{T}\boldsymbol{H}\boldsymbol{Q}$ とする。
公開鍵を $\tilde{\boldsymbol{H}}$ とし，秘密鍵を $(\boldsymbol{T}, \boldsymbol{H}, \boldsymbol{Q})$ とする。
暗号化: 平文を $\boldsymbol{e} \in \mathcal{S}_H(n, t)$ とする。暗号文 $\boldsymbol{d} = \boldsymbol{e}\tilde{\boldsymbol{H}}^\top \in \mathbb{F}_2^{n-k}$ を計算する。
復号: $\hat{\boldsymbol{w}} = \boldsymbol{d}\boldsymbol{T}^{-\top}$ を計算する。$\hat{\boldsymbol{w}}$ を線形符号で訂正し復号し，誤りとして $\boldsymbol{e}'$ を得る。$\boldsymbol{e} = \boldsymbol{e}'\boldsymbol{Q}^{-\top}$ を出力する。

復号の正当性は以下で確認される。$\boldsymbol{d} = \boldsymbol{e}\tilde{\boldsymbol{H}}^\top$ として，$\hat{\boldsymbol{w}} = \boldsymbol{d}\boldsymbol{T}^{-\top}$ を計算すると，

$$\hat{\boldsymbol{v}} = \boldsymbol{e}\tilde{\boldsymbol{H}}^\top\boldsymbol{T}^{-\top} = \boldsymbol{e}\boldsymbol{Q}^\top\boldsymbol{H}^\top\boldsymbol{T}^\top\boldsymbol{T}^{-\top} = \boldsymbol{e}\boldsymbol{Q}^\top\boldsymbol{H}^\top$$

を得る。$\boldsymbol{e}\boldsymbol{Q}^\top$ はランダムに置換されたエラーであり，$\boldsymbol{e}\boldsymbol{Q}^\top\boldsymbol{H}^\top$ はシンドロームである。$\boldsymbol{e}\boldsymbol{Q}^\top$ のハミング重みが t 以下であれば，線形符号の復号により，$\boldsymbol{e}' = \boldsymbol{e}\boldsymbol{Q}^\top$ を得る。よって，高い確率で復号に成功する。平文 $\boldsymbol{e}$ およびパリティ検査行列 $\tilde{\boldsymbol{H}}$ が一様ランダムであれば，暗号文 $\boldsymbol{d} \in \mathbb{F}_2^{n-k}$ はランダムな $n-k$ 次元のベクトルと見分けが付かないと考えられている。また，$\tilde{\boldsymbol{H}}$ が一様ランダムであり，適切な t が選択されていれば，暗号文は統計的にランダムなベクトルと見分けが付かないとされている。一方で，オリジナルの Niederreiter 暗号は適応的選択暗号文攻撃（CCA）に対して安全ではないため，次節で示すより安全な方式が提案されている。

4.2.3 符号版 Lyubashevsky-Peikert-Regev（LPR）暗号

符号版 LPR 暗号は，Lyubashevsky，Peikert，Regev が 2010 年に提案した Ring-LWE 問題に基づく暗号方式 [42] を LPN 問題に基づく方式に変更したものである。以下では $q = 2$ とする。

- k：安全性パラメータ
- $n = n_1 + n_2$：サンプルの個数
- ℓ：平文長
- τ：誤差パラメータ（例: $\tau n = O(\sqrt{k})$）
- t：線形符号の誤り訂正能力（$t = \Omega\big((\tau n)^2\big)$）

鍵生成: 誤り訂正能力が t である $[n_2, \ell]_2$ -線形符号の生成行列 $\boldsymbol{G}_c$ を生成する。$\boldsymbol{A} \leftarrow \mathbb{F}_2^{k\times n_1}$ とする。$\boldsymbol{X} \leftarrow \mathsf{Ber}_\tau^{n_1\times n_2}, \boldsymbol{Y} \leftarrow \mathsf{Ber}_\tau^{k\times n_2}$ とし，$\boldsymbol{B} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{Y} \in \mathbb{F}_2^{k\times n_2}$ とする。
公開鍵を $\tilde{\boldsymbol{G}} = [\boldsymbol{A} \mid \boldsymbol{B}] \in \mathbb{F}_2^{k\times n}$ とし，秘密鍵を $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{X})$ とする。
暗号化: 平文を $\boldsymbol{m} \in \mathbb{F}_2^\ell$ とする。乱数 $\boldsymbol{s} \leftarrow \mathsf{Ber}_\tau^k$ と 乱数 $\boldsymbol{e} \leftarrow \mathsf{Ber}_\tau^n$ を選び，暗号文 $\boldsymbol{c} = \boldsymbol{s}\tilde{\boldsymbol{G}} + \boldsymbol{e} + (\boldsymbol{0}_{n_1}, \boldsymbol{m}\boldsymbol{G}_c) \in \mathbb{F}_2^n$ を計算する。
復号: $\boldsymbol{d} = \boldsymbol{c}\binom{-\boldsymbol{X}}{\boldsymbol{I}_{n_2}}$ を計算する。$\boldsymbol{d}$ を線形符号で訂正し復号すると $\boldsymbol{m}$ を得る。

復号の正当性は以下で確認される。$\boldsymbol{c} = \boldsymbol{s}\tilde{\boldsymbol{G}} + \boldsymbol{e} + (\boldsymbol{0}_{n_1}, \boldsymbol{m}\boldsymbol{G}_c)$ なので，前半部を $\boldsymbol{u} = \boldsymbol{s}\boldsymbol{A} + \boldsymbol{e}_1$，後半部を $\boldsymbol{v} = \boldsymbol{s}\boldsymbol{B} + \boldsymbol{e}_2 + \boldsymbol{m}\boldsymbol{G}_c$ と書く。

$\boldsymbol{d} = \boldsymbol{c}\binom{-\boldsymbol{X}}{\boldsymbol{I}_{n_2}}$ を計算すると，

$$\boldsymbol{d} = \boldsymbol{v} - \boldsymbol{u}\boldsymbol{X} = \boldsymbol{s}\boldsymbol{B} + \boldsymbol{e}_2 + \boldsymbol{m}\boldsymbol{G}_c - \boldsymbol{s}\boldsymbol{A}\boldsymbol{X} - \boldsymbol{e}_1\boldsymbol{X} = \boldsymbol{m}\boldsymbol{G}_c + (\boldsymbol{e}_2 - \boldsymbol{e}_1\boldsymbol{X} + \boldsymbol{s}\boldsymbol{Y})$$

を得る。$\boldsymbol{m}\boldsymbol{G}_c$ は符号語であり，$\boldsymbol{e}_2 - \boldsymbol{e}_1\boldsymbol{X} + \boldsymbol{s}\boldsymbol{Y}$ は誤りベクトルである。よって，$\boldsymbol{e}_2 - \boldsymbol{e}_1\boldsymbol{X} + \boldsymbol{s}\boldsymbol{Y}$ のハミング重みが t 以下であれば，線形符号の復号により，$\boldsymbol{m}$ を得る。高い確率で $\boldsymbol{e}_2 - \boldsymbol{e}_1\boldsymbol{X} + \boldsymbol{s}\boldsymbol{Y}$ のハミング重みが t 以下になるように τ を設定しているため，高い確率で復号に成功する。

4.2.4 CFS 署名

CFS 署名は Courtois，Finiasz，Sendrier が 2001 年に提案した署名である [21]。のちに，安全性仮定が提案パラメータセットでは成り立たないことが示された [31, 32]。しかし後の方式に大きな影響を与えたため，ここに記す。Niederreiter 暗号を思い出すと，秘密鍵を持っている場合，ハミング重みが t 以下の誤りは訂正できる。一方，訂正可能なシンドロームの集合 $\{\boldsymbol{e}\tilde{\boldsymbol{H}} \in \mathbb{F}_2^{n-k} \mid \boldsymbol{e} \in \mathcal{S}_H^{\leq}(n,t)\}$ のサイズは $\mathbb{F}_2^{n-k}$ のサイズに比べれば圧倒的に少ない。

3.2.5 節のように Hash-and-Sign に基づいた構成を考える。メッセージ M のハッシュ値を シンドローム $\boldsymbol{u} \in \mathbb{F}_2^{n-k}$ と捉えた場合，正しく復号できないシンドロームになることが多い。そこで CFS 署名では，ハッシュ値を $\boldsymbol{u} = \mathsf{Hash}(M,i)$ と i をインクリメントしながら計算し，ハッシュ値が $\{\boldsymbol{e}\tilde{\boldsymbol{H}} \in \mathbb{F}_2^{n-k} \mid \boldsymbol{e} \in \mathcal{S}_H^{\leq}(n,t)\}$ に入るものを採用する。

署名鍵と検証鍵: パリティ検査行列 $\tilde{\boldsymbol{H}} \in \mathbb{F}_2^{(n-k)\times n}$ を検証鍵とする。また署名鍵を用いると，ハミング重み t 以下の符号語を訂正できることとする。

署名: 文書 M について，

1. $i = 0$ とする。
2. $\boldsymbol{u} = \mathsf{Hash}(M,i)$ を計算する。
3. ハミング重み t 以下の $\boldsymbol{e}$ で，$\boldsymbol{e}\tilde{\boldsymbol{H}}^\top = \boldsymbol{u}$ となるものを計算する。なければ $i \leftarrow i+1$ としてステップ 2 に戻る。
4. $\sigma = (\boldsymbol{e}, i)$ を出力する。

検証: 文書 M と $\sigma = (\boldsymbol{e}, i)$ について，$\mathsf{HW}(\boldsymbol{e}) \leq t$ と $\boldsymbol{e}\tilde{\boldsymbol{H}}^\top = \mathsf{Hash}(M,i)$ ならば，受理する。そうでないならば，棄却とする。

4.3 符号に基づく主要な暗号方式

本稿では以下の暗号方式を取り上げる。いずれも NIST PQC 標準化プロジェクトにおいて第 4 ラウンドに進んだものである。

1. Classic McEliece: Niederreiter 暗号を採用し，符号の構成が非常に保守的という観点からこれを取り上げる。
2. BIKE: Niederreiter 暗号を採用し，QC-MDPC 符号を用いて鍵を圧縮している，という観点からこれを取り上げる。
3. HQC: 符号版の LPR 暗号を採用，Quasi-Cyclic 符号を用いて鍵を圧縮している，という特徴からこれを取り上げる。

表 4.2: 符号に基づく暗号の分類

文献	暗号化	鍵交換	署名
Classic McEliece [2]	○	○	–
BIKE [4]	○	○	–
HQC [46]	○	○	–

4.3.1 Classic McEliece

- 提案者: Albrecht, Bernstein, Chou, Cid, Gilcher, Lange, Maram, von Maurich, Misoczki, Niederhagen, Paterson, Persichetti, Peters, Schwabe, Sendrier, Szefer, Tjhai, Tomlinson, Wang
- 基本方式の説明: Niederreiter 暗号方式に基づいている。基本符号方式として $\mathbb{F}_2$ 上の Goppa 符号を利用している。（具体的な Goppa 符号の生成方法や符号化および復号の方法については提案方式の仕様書 [2] を参照のこと。）$q = 2^m$ とし，$n \leq q$ を用いる。2 以上の t を $mt < n$ となるように取り，$k = n - mt$ とする。

  鍵生成: 誤り訂正能力が t である Goppa 符号のパリティ検査行列 $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ をランダムに生成する。組織符号化し，$\tilde{\boldsymbol{H}} = [\boldsymbol{I}_{n-k} \mid \boldsymbol{T}]$ とする。公開鍵を $pk = \boldsymbol{T} \in \mathbb{F}_2^{(n-k)\times k}$ とする。符号生成に使ったパラメータを Γ（$\mathbb{F}_q$ 係数の t 次モニック既約多項式と互いに異なる $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_q$）とする。秘密鍵を $sk = \Gamma$ とする。

  暗号化 $\mathsf{Encrypt}(pk, \boldsymbol{e})$：入力を $\boldsymbol{e} \in \mathcal{S}_H(n,t)$ とする。$\tilde{\boldsymbol{H}} = [\boldsymbol{I}_{n-k} \mid \boldsymbol{T}]$ とし，暗号文として $\boldsymbol{c} = \tilde{\boldsymbol{H}}\boldsymbol{e} \in \mathbb{F}_2^{n-k}$ を出力する。

  復号 $\mathsf{Decrypt}(sk, \boldsymbol{c})$：ハミング重み t のベクトル $\boldsymbol{e}$ を復号する。

  1. $\boldsymbol{c}$ に k 個ゼロを加え，$\boldsymbol{v} = (\boldsymbol{c}, \boldsymbol{0}_k) \in \mathbb{F}_2^n$ を考える。
  2. Goppa 符号の復号アルゴリズムを用いて，$\boldsymbol{v}$ と距離 t 以下にある符号語 $\boldsymbol{d}$ を計算する。（なければ $\perp$ を出力する。）
  3. $\boldsymbol{e} = \boldsymbol{v} + \boldsymbol{d}$ とする。
  4. $\mathsf{HW}(\boldsymbol{e}) = t$ かつ $\boldsymbol{c} = \tilde{\boldsymbol{H}}\boldsymbol{e}$ ならば $\boldsymbol{e}$ を出力する。（そうでなければ $\perp$ を出力する。）
- 鍵カプセル化方式の説明: 基本方式を決定性の公開鍵暗号とみなし，藤崎–岡本変換の変種をかけたものとみなせる。以下ではハッシュ関数 $\mathsf{H}\colon \{0,1\}^* \to \{0,1\}^{256}$ を用いる。

  鍵生成: ℓ ビットのシード δ から乱数を生成し，鍵生成を行う。（乱数の生成方法は省略する。）公開鍵は同じく $pk = \boldsymbol{T}$ である。n ビットの一様ランダムな文字列 $\boldsymbol{s}$ を生成する。秘密鍵は $sk = (\Gamma, \boldsymbol{s})$ である。

  鍵カプセル化: 1. $\boldsymbol{e} \leftarrow \mathcal{S}_H(n,t)$ をあるアルゴリズムに従ってランダム生成する。
  2. $\boldsymbol{c} = \mathsf{Encrypt}(pk, \boldsymbol{e})$ を計算する。
  3. $K = \mathsf{H}(1, \boldsymbol{e}, \boldsymbol{c})$ とする。
  4. 暗号文を $\boldsymbol{c}$ とし，セッション鍵 K を出力する。

  デカプセル化: 1. $b = 1$ とする。
  2. 受診した $\boldsymbol{c}$ に対して，$\boldsymbol{e} = \mathsf{Decrypt}(sk, \boldsymbol{c})$ とする。$\boldsymbol{e} = \perp$ であれば，$b = 0, \boldsymbol{e} = \boldsymbol{s}$ と上書きする。
  3. $K = \mathsf{H}(b, \boldsymbol{e}, \boldsymbol{c})$ を計算する。
  4. K を出力する。

  以上より，セッション鍵（共通鍵）K を安全に共有することができる。

パラメータセットとして mceliece348864，mceliece348864f，mceliece460896，mceliece460896f，mceliece6688128，

表 4.3: Classic McEliece のパラメータ。公開鍵長，秘密鍵長，暗号文長の単位はそれぞれ Byte とする。

パラメータ名	(m, n, t)	安全性レベル	公開鍵長	秘密鍵長	暗号文長	復号エラー率
mceliece348864	$(12, 3488, 64)$	レベル 1	$261,120$	$6,492$	96	0
mceliece460896	$(13, 4608, 96)$	レベル 3	$524,160$	$13,608$	156	0
mceliece6688128	$(13, 6688, 128)$	レベル 5	$1,044,992$	$13,932$	208	0
mceliece6960119	$(13, 6960, 119)$	レベル 5	$1,047,319$	$13,948$	194	0
mceliece8192128	$(13, 8192, 128)$	レベル 5	$1,357,824$	$14,120$	208	0

mceliece6688128f, mceliece6960119, mceliece6960119f, mceliece8192128, mceliece8192128f が提案されている。表 4.3 に鍵カプセル化方式のパラメータ，鍵長および暗号文長，想定セキュリティレベル，復号エラー率をまとめた。今回末尾に f が付くものは扱っていないが，鍵長・暗号文長は f 無しのものと同一である。Classic McEliece は公開鍵長が非常に大きく，レベル 5 では 1 メガバイトを超える。一方で，暗号文長は非常に小さく，格子暗号に基づく FIPS 標準（FIPS 203）である ML-KEM [50] の暗号文サイズよりも小さい。例えば，[50, Table 3] によれば，ML-KEM のレベル 1 の公開鍵長は 800 Byte，秘密鍵長は 1632 Byte，暗号文長は 768 Byte となっている。

mceliece348864 の速度に関しては，鍵生成に必要な平均 CPU サイクル数が $60,333,686$ Cycle，鍵カプセル化が $37,585$ Cycle，デカプセル化が $127,668$ Cycle である。参考までに，ML-KEM（Kyber-512）の速度は，鍵生成が $33,428$ Cycle，鍵カプセル化が $49,184$ Cycle，デカプセル化が $40,564$ Cycle である [50]。なお，いずれも Haswell CPU 搭載のサーバ上で AVX 命令を使用した C 言語実装を動作させた時の記録である。他のパラメータに関しては，仕様書を参照されたい。

4.3.2 BIKE

- 提案者: Aragon, Barreto, Bettaieb, Bidoux, Blazy, Deneuville, Gaborit, Gueron, Güneysu, Aguilar Melchor, Misoczki, Persichetti, Sendrier, Tillich, Zémor, Vasseur, Ghosh, Richter-Brokmann
- 基本方式の説明: Niederreiter 暗号方式に基づいている。基本となる符号に QC-MDPC 符号を採用し，公開鍵サイズを圧縮している。そのため，鍵や暗号化は格子暗号の一種の NTRU 暗号と非常に近い形をしている点が特徴である。具体的な符号化および復号の方法については提案方式の仕様書 [4] を参照のこと。以下では，$\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1)$ とする。

  鍵生成: $\boldsymbol{h}_0 \in \mathcal{R}$ および $\boldsymbol{h}_1 \in \mathcal{R}$ を $\mathcal{S}_H(n, w/2)$ から一様ランダムに選ぶ。$\boldsymbol{h} = \boldsymbol{h}_1/\boldsymbol{h}_0 \in \mathcal{R}$ とする。$(\boldsymbol{h}_0, \boldsymbol{h}_1)$ を QC-MDPC 符号のパリティ検査行列とし，$(\mathbf{1}, \boldsymbol{h})$ をその組織符号化したものとみなすことができる。公開鍵を $pk = \boldsymbol{h}$ とし，秘密鍵を $sk = (\boldsymbol{h}_0, \boldsymbol{h}_1)$ とする。

  暗号化 $\mathsf{Encrypt}(pk, (\boldsymbol{e}_0, \boldsymbol{e}_1))$: $(\boldsymbol{e}_0, \boldsymbol{e}_1) \in \mathcal{R}^2$ を $\mathcal{S}_H(2n, t)$ 中のベクトルとみなす。$\boldsymbol{c} = \boldsymbol{e}_0 + \boldsymbol{e}_1\boldsymbol{h} \in \mathcal{R}$ を出力する。

  復号 $\mathsf{Decrypt}(sk, \boldsymbol{c})$: ハミング重み t 以下のベクトル $(\boldsymbol{e}_0, \boldsymbol{e}_1)$ を復号する。

  1. $\boldsymbol{c}\boldsymbol{h}_0$ を計算する。
  2. QC-MDPC 符号の復号アルゴリズムを用いて，$\boldsymbol{c}\boldsymbol{h}_0$ をシンドロームとするベクトル $(\boldsymbol{e}_0, \boldsymbol{e}_1)$ を計算する。
- 鍵カプセル化方式の説明: 基本方式を決定性公開鍵暗号方式とみなす。基本方式とハッシュ関数 $\mathsf{L}\colon \{0,1\}^* \to \{0,1\}^{256}$ を用いて，平文 $\boldsymbol{m} \in \{0,1\}^{256}$ と乱数 $(\boldsymbol{e}_0, \boldsymbol{e}_1)$ に対して暗号化 $(\boldsymbol{c}_0 = \mathsf{Encrypt}(pk, (\boldsymbol{e}_0, \boldsymbol{e}_1)))$ および $\boldsymbol{m}$ のマスキング $(\boldsymbol{c}_1 = \boldsymbol{m} \oplus \mathsf{L}(\boldsymbol{e}_0, \boldsymbol{e}_1))$ とを行う IND-CPA 安全な乱択公開鍵暗号を構成する。鍵カプセル

表 4.4: BIKE のパラメータ。公開鍵長，秘密鍵長，暗号文長の単位はそれぞれ Byte とする。

パラメータ名	(n, w, t)	安全性レベル	公開鍵長	秘密鍵長	暗号文長	復号エラー率
BIKE-Level1	$(12323, 142, 134)$	レベル 1	$1,541$	281	$1,573$	2^{-128}
BIKE-Level3	$(24659, 206, 199)$	レベル 3	$3,083$	419	$3,115$	2^{-192}
BIKE-Level5	$(40973, 274, 264)$	レベル 5	$5,122$	580	$5,154$	2^{-256}

化方式は，この乱択公開鍵暗号に藤崎–岡本変換の変種を適用したものとみなせる。以下ではハッシュ関数 $\mathsf{H}, \mathsf{L}\colon \{0,1\}^* \to \{0,1\}^{256}$ と $\mathsf{G}\colon \{0,1\}^* \to \mathcal{S}_H(2n, t)$ を用いる。

鍵生成: 適切な長さのシード δ から乱数を生成し，鍵生成を行う。公開鍵は同じく $pk = \boldsymbol{h}$ である。ℓ ビットの一様ランダムな文字列 $\boldsymbol{s} \in \{0,1\}^\ell$ を生成する。秘密鍵は $sk = (\boldsymbol{h}_0, \boldsymbol{h}_1, \boldsymbol{s})$ である。

鍵カプセル化:
1. $\boldsymbol{m} \leftarrow \{0,1\}^{256}$ を一様ランダムに選ぶ。
2. $(\boldsymbol{e}_0, \boldsymbol{e}_1) = \mathsf{G}(\boldsymbol{m})$ を計算する。
3. $\boldsymbol{c}_0 = \mathsf{Encrypt}(pk, (\boldsymbol{e}_0, \boldsymbol{e}_1))$ と，$\boldsymbol{c}_1 = \boldsymbol{m} \oplus \mathsf{L}(\boldsymbol{e}_0, \boldsymbol{e}_1)$ を計算する。
4. $K = \mathsf{H}(\boldsymbol{m}, \boldsymbol{c})$ を計算する。
5. 暗号文を $C = (\boldsymbol{c}_0, \boldsymbol{c}_1)$ とし，セッション鍵 K を出力する。

デカプセル化:
1. 受信した C に対して，$(\boldsymbol{e}'_0, \boldsymbol{e}'_1) = \mathsf{Decrypt}(sk, \boldsymbol{c}_0)$ を計算する。
2. 復号に失敗したら，$\perp$ を出力して停止する。
3. $\boldsymbol{m}' = \boldsymbol{c}_1 \oplus \mathsf{L}(\boldsymbol{e}'_0, \boldsymbol{e}'_1)$ を計算する。
4. $(\boldsymbol{e}'_0, \boldsymbol{e}'_1) = \mathsf{G}(\boldsymbol{m}')$ ならば，$K = \mathsf{H}(\boldsymbol{m}', \boldsymbol{c})$ を出力して停止する。
5. そうでなければ，$K = \mathsf{H}(\boldsymbol{s}, \boldsymbol{c})$ を計算し，出力する。

以上より，セッション鍵（共通鍵）K を安全に共有することができる。

表 4.4 に鍵カプセル化方式のパラメータ，鍵長，暗号文長および復号エラー率をまとめた。3 つのパラメータセットがそれぞれレベル 1，3，5 相当として提案された。BIKE-Level1 の速度に関しては，鍵生成が $589,000$ Cycle，鍵カプセル化が $97,000$ Cycle，デカプセル化が $1,135,000$ Cycle である（Skylake CPU 搭載のサーバ，AVX 命令を使用）。他のパラメータに関しては，仕様書を参照されたい。

4.3.3 HQC

- 提案者: Aguilar Melchor，Aragon，Bettaieb，Bidoux，Blazy，Deneuville，Gaborit，Persichetti，Zémor，Bos，Dion，Lacan，Robert，Veron
- 基本方式の説明: 符号版の LPR 暗号に基づき，公開鍵暗号を構成している。
  $\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1)$ とする。$n' = n_1 n_2$ とし，$[n', k]$ 線形符号 $\mathcal{C}$ を採用する。具体的な符号化および復号の方法については提案方式の仕様書 [46] を参照のこと。線形符号 $\mathcal{C}$ の符号化・復号アルゴリズムを $\mathsf{encode}, \mathsf{decode}$ とする。$n \geq n'$ を仮定する。以下では，暗号文の第二要素 $\boldsymbol{v}$ を $\mathcal{R}$ 要素（n ビットベクトル）として扱っているが，実際には n' ビットに縮めて用いる。

  鍵生成: $\boldsymbol{x} \in \mathcal{R}$ および $\boldsymbol{y} \in \mathcal{R}$ を $\mathcal{S}_H(n, w)$ から一様ランダムに選び，$\boldsymbol{h} \leftarrow \mathcal{R}$ に対して公開鍵を $pk = (\boldsymbol{h}, \boldsymbol{s}) \in \mathcal{R}^2$ とし，秘密鍵を $sk = (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{R}^2$ とする。

  暗号化 $\mathsf{Encrypt}(pk, \boldsymbol{m}, \boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{e})$：$\boldsymbol{r}_1 \in \mathcal{R}$ および $\boldsymbol{r}_2 \in \mathcal{R}$ を $\mathcal{S}_H(n, w_r)$ から一様ランダムに選び，$\boldsymbol{e} \in \mathcal{R}$ を $\mathcal{S}_H(n, w_e)$ から一様ランダムに選ぶ。$\boldsymbol{u} = \boldsymbol{r}_1 + \boldsymbol{h} \cdot \boldsymbol{r}_2$ および $\boldsymbol{v} = \mathsf{encode}(m) + \boldsymbol{s} \cdot \boldsymbol{r}_2 + \boldsymbol{e}$ を計算する。$\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{v})$ を暗号文として出力する。

表 4.5: HQC のパラメータ。公開鍵長，秘密鍵長，暗号文長の単位はそれぞれ Byte とする。

パラメータ名	$(n_1, n_2, n, w, w_r = w_e)$	安全性レベル	公開鍵長	秘密鍵長	暗号文長	復号エラー率
hqc-128	$(46, 384, 17669, 66, 75)$	レベル 1	$2,249$	40	$4,497$	2^{-128}
hqc-192	$(56, 640, 35851, 100, 114)$	レベル 3	$4,522$	40	$9,042$	2^{-192}
hqc-256	$(90, 640, 57637, 131, 149)$	レベル 5	$7,245$	40	$14,485$	2^{-256}

復号 $\mathsf{Decrypt}(sk, \boldsymbol{c})$: $\mathsf{decode}(\boldsymbol{v} - \boldsymbol{u} \cdot \boldsymbol{y})$ を出力する。

- 鍵カプセル化方式: 基本方式を乱択な公開鍵暗号とみなし，藤崎–岡本変換の変種を適用したものとみなせる。以下ではハッシュ関数 $\mathsf{H}, \mathsf{H}' \colon \{0,1\}^* \to \{0,1\}^{256}$ を用いる。また，XOF [*6] として $\mathsf{H}_G \colon \{0,1\}^* \to \{0,1\}^*$ も用いる。（第 4 ラウンドで G への入力に $\mathrm{seed} \in \{0,1\}^{128}$ と $\mathrm{salt} \in \{0,1\}^{128}$ が追加された。）

  鍵生成: 同上。ただし $\boldsymbol{h}$ の生成をシード seed から行うこととし，公開鍵を $pk = (\boldsymbol{s}, \mathrm{seed})$ とする。また，秘密鍵にもシードを加え，$sk = (\boldsymbol{x}, \boldsymbol{y}, \mathrm{seed})$ とする。

  鍵カプセル化: 1. $\boldsymbol{m} \leftarrow \mathbb{F}_2^k$ を一様ランダムにとる。

  2. $\mathrm{salt} \leftarrow \mathbb{F}_2^{128}$ を一様ランダムにとる。

  3. $\theta = \mathsf{H}_G(\boldsymbol{m}, \mathrm{seed}, \mathrm{salt})$ を計算する。θ から $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{e}$ を生成する。

  4. $\boldsymbol{c} = \mathsf{Encrypt}(pk, \boldsymbol{m}, \boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{e})$ を計算する。$\boldsymbol{d} = \mathsf{H}'(\boldsymbol{m})$ とする。$K = \mathsf{H}(\boldsymbol{m}, \boldsymbol{c})$ とする。

  5. 暗号文を $C = (\boldsymbol{c}, \boldsymbol{d}, \mathrm{salt})$ とし，セッション鍵 K を出力する。

  デカプセル化: 1. 受信した C に対して，$\boldsymbol{m}' = \mathsf{Decrypt}(sk, \boldsymbol{c})$ を計算する。

  2. $\theta' = \mathsf{H}_G(\boldsymbol{m}', \mathrm{seed}, \mathrm{salt})$ を計算する。θ' から $\boldsymbol{r}_1', \boldsymbol{r}_2', \boldsymbol{e}'$ を生成する。

  3. $\boldsymbol{c}' = \mathsf{Encrypt}(pk, \boldsymbol{m}', \boldsymbol{r}_1', \boldsymbol{r}_2', \boldsymbol{e}')$ を計算する。$\boldsymbol{c} \neq \boldsymbol{c}'$ もしくは $\boldsymbol{d} \neq \boldsymbol{d}'$ ならば $\perp$ を出力して停止する。

  4. $K = \mathsf{H}(\boldsymbol{m}, \boldsymbol{c})$ を出力する。

  以上より，セッション鍵（共通鍵）K を安全に共有することができる。

3 つのパラメータセットがそれぞれレベル 1，3，5 相当として提案された。表 4.5 に鍵カプセル化方式のパラメータ，鍵長，暗号文長および復号エラー率をまとめた。表中では，秘密鍵はシードだけ記憶していることにされており，40 バイトしかない。また公開鍵の $\boldsymbol{h}$ の部分もシードから再生成されることと定義されている点に注意されたい。hqc-128 の速度に関しては，鍵生成が $87,000$ Cycle，鍵カプセル化が $204,000$ Cycle，デカプセル化が $362,000$ Cycle である（Skylake CPU 搭載のデスクトップ PC，AVX 命令を使用）。他のパラメータに関しては，仕様書を参照されたい。

4.4 符号に基づく暗号技術に関するまとめ

基本となる McEliece 暗号方式は，McEliece により 40 年以上前に提案されており，パラメータは改訂されているものの，いまだに破られていない。Classic McEliece などのように，公開鍵や秘密鍵は長いものの，暗号文は短い方式が多い。LPN 問題は学習理論や符号理論から派生した問題であり，SD 問題は LPN 問題の特殊な場合である。誤り確率 τ が十分大きい場合の LPN 問題や，重み w が一定の大きさの SD 問題を確率的多項式時間で効率的に解くことは，量子コンピュータを用いても困難であると予想されている。

共通鍵暗号や公開鍵暗号の分野で多くの方式が LPN 問題や SD 問題に基づいて提案されている。LWE 問題と比較した場合，利点としては，

- $\mathbb{F}_2$ およびその拡大体を基に構成するため，ハードウェア構成との相性が良い点

[*6] eXtendable-Output Function の略。SHAKE128 や SHAKE256 が例として知られている。

- 誤差分布としてベルヌーイ分布やその一般化した分布を用いるため，誤差のサンプリングが容易である点

が挙げられる。一方，欠点として，

- 鍵や暗号文のサイズが大きくなりやすい点
- 符号の復号アルゴリズムが複雑になりがちな点
- 完全準同型暗号といった発展的な応用が少ない点

が挙げられる。暗号方式のパラメータ設定の際には，4.1 節で挙げたさまざまなアルゴリズムを考慮する必要がある。アルゴリズムの高速化について盛んに研究されており，動向を注視する必要がある。また，符号に基づく暗号技術の信頼性を向上させるためには，理論面における研究だけではなく，実時間の計算量に関する研究も重要である。公開鍵や秘密鍵を圧縮しようと特殊な符号を採用したり，距離の定義を変える提案も多くある。これらは解読攻撃を受けることも多く，評価が確定していない暗号・署名方式については注視が必要である。

第 4 章の参照文献

[1] CRYPTREC 暗号技術調査 WG (耐量子計算機暗号). CRYPTREC 耐量子計算機暗号の研究動向調査報告書. CRYPTREC GL-2004-2022, https://www.cryptrec.go.jp/report/cryptrec-tr-2001-2022.pdf. 2023-03.

[2] M. R. Albrecht et al. Classic McEliece: conservative code-based cryptography. https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/mceliece-Round4.tar.gz. 2022-10. (2024-03-05 閲覧).

[3] B. Applebaum, D. Cash, C. Peikert, A. Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. CRYPTO. Vol. 5677. Lecture Notes in Computer Science. Springer, 2009, pp. 595–618.

[4] N. Aragon et al. BIKE: Bit flipping key encapsulation (Round 4 submission). https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/BIKE-Round4.zip. 2022-10. (2024-03-05 閲覧).

[5] N. Aragon et al. RYDE. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/ryde-submission.zip. 2024-11. (2024-11-15 閲覧).

[6] S. Arora, R. Ge. New Algorithms for Learning in Presence of Errors. ICALP (1). Vol. 6755. Lecture Notes in Computer Science. Springer, 2011, pp. 403–415.

[7] M. Baldi et al. CROSS (Codes and Restricted Objects Signature Scheme). https://cross-crypto.com/. 2024-11. (2024-11-15 閲覧).

[8] A. Becker, A. Joux, A. May, A. Meurer. Decoding Random Binary Linear Codes in $2^{n/20}$: How $1 + 1 = 0$ Improves Information Set Decoding. EUROCRYPT. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 520–536.

[9] E. R. Berlekamp, R. J. McEliece, H. C. A. van Tilborg. On the inherent intractability of certain coding problems (Corresp.) IEEE Trans. Inf. Theory. Vol. 24, Num. 3 (1978), pp. 384–386.

[10] D. J. Bernstein, T. Lange. Never Trust a Bunny. RFIDSec. Vol. 7739. Lecture Notes in Computer Science. Springer, 2012, pp. 137–148.

[11] A. Blum, M. L. Furst, M. J. Kearns, R. J. Lipton. Cryptographic Primitives Based on Hard Learning Problems. CRYPTO. Vol. 773. Lecture Notes in Computer Science. Springer, 1993, pp. 278–291.

[12] A. Blum, A. Kalai, H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM. Vol. 50, Num. 4 (2003), pp. 506–519.

[13] S. Bogos, S. Vaudenay. Optimization of LPN Solving Algorithms. ASIACRYPT (1). Vol. 10031. Lecture Notes in Computer Science. 2016, pp. 703–728.

[14] L. Both, A. May. Decoding Linear Codes with High Error Rate and Its Impact for LPN Security. PQCrypto. Vol. 10786. Lecture Notes in Computer Science. Springer, 2018, pp. 25–46.

[15] L. Both, A. May. Optimizing BJMM with nearest neighbors: Full decoding in $2^{2n/21}$ and McEliece security. Workshop on Coding and Cryptography. 2017. https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/paper/bjmm+.pdf.

[16] E. Carozza, G. Couteau, A. Joux. Short Signatures from Regular Syndrome Decoding in the Head. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 532–563.

[17] K. Carrier, T. Debris-Alazard, C. Meyer-Hilfiger, J.-P. Tillich. Reduction from Sparse LPN to LPN, Dual Attack 3.0. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 286–315.

[18] K. Carrier, T. Debris-Alazard, C. Meyer-Hilfiger, J.-P. Tillich. Statistical Decoding 2.0: Reducing Decoding to LPN. ASIACRYPT (4). Vol. 13794. Lecture Notes in Computer Science. Springer, 2022, pp. 477–507.

[19] J. Carrijo, R. Tonicelli, H. Imai, A. C. A. Nascimento. A Novel Probabilistic Passive Attack on the Protocols HB and HB^{+}. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. Vol. 92-A, Num. 2 (2009), pp. 658–662.

[20] C. Chevignard, P.-A. Fouque, A. Schrottenloher. Reducing the Number of Qubits in Quantum Information Set Decoding. ASIACRYPT (8). Vol. 15491. Lecture Notes in Computer Science. Springer, 2024, pp. 299–329.

[21] N. T. Courtois, M. Finiasz, N. Sendrier. How to Achieve a McEliece-Based Digital Signature Scheme. ASIACRYPT. Vol. 2248. Lecture Notes in Computer Science. Springer, 2001, pp. 157–174.

[22] S. Devadas, L. Ren, H. Xiao. On Iterative Collision Search for LPN and Subset Sum. TCC (2). Vol. 10678. Lecture Notes in Computer Science. Springer, 2017, pp. 729–746.

[23] I. Dinur, O. Dunkelman, N. Keller, A. Shamir. Efficient Dissection of Composite Problems, with Applications to Cryptanalysis, Knapsacks, and Combinatorial Search Problems. CRYPTO. Vol. 7417. Lecture Notes in Computer Science. Springer, 2012, pp. 719–740.

[24] L. Ducas, A. Esser, S. Etinski, E. Kirshanova. Asymptotics and Improvements of Sieving for Codes. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 151–180.

[25] A. Esser. Revisiting Nearest-Neighbor-Based Information Set Decoding. IMACC. Vol. 14421. Lecture Notes in Computer Science. Springer, 2023, pp. 34–54.

[26] A. Esser, F. Heuer, R. Kübler, A. May, C. Sohler. Dissection-BKW. CRYPTO (2). Vol. 10992. Lecture Notes in Computer Science. Springer, 2018, pp. 638–666.

[27] A. Esser, R. Kübler, A. May. LPN Decoded. CRYPTO (2). Vol. 10402. Lecture Notes in Computer Science. Springer, 2017, pp. 486–514.

[28] A. Esser, S. Ramos-Calderer, E. Bellini, J. I. Latorre, M. Manzano. Hybrid Decoding – Classical-Quantum Trade-Offs for Information Set Decoding. PQCrypto. Vol. 13512. Lecture Notes in Computer Science. Springer, 2022, pp. 3–23.

[29] A. Esser, J. A. Verbel, F. Zweydinger, E. Bellini. SoK: CryptographicEstimators – a Software Library for Cryptographic Hardness Estimation. AsiaCCS. ACM, 2024.

[30] A. Esser, F. Zweydinger. New Time-Memory Trade-Offs for Subset Sum – Improving ISD in Theory and Practice. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 360–390.

[31] J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, J.-P. Tillich. A distinguisher for high rate McEliece cryptosystems. ITW. IEEE, 2011, pp. 282–286.

[32] J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, J.-P. Tillich. A Distinguisher for High-Rate McEliece Cryptosystems. IEEE Trans. Inf. Theory. Vol. 59, Num. 10 (2013), pp. 6830–6844.

[33] Q. Guo, T. Johansson, C. Löndahl. Solving LPN Using Covering Codes. J. Cryptol. Vol. 33, Num. 1 (2020), pp. 1–33.

[34] J. Håstad. Some optimal inapproximability results. J. ACM. Vol. 48, Num. 4 (2001), pp. 798–859.

[35] S. Heyse, E. Kiltz, V. Lyubashevsky, C. Paar, K. Pietrzak. Lapin: An Efficient Authentication Protocol Based on Ring-LPN. FSE. Vol. 7549. Lecture Notes in Computer Science. Springer, 2012, pp. 346–365.

[36] G. Kachigar, J.-P. Tillich. Quantum Information Set Decoding Algorithms. PQCrypto. Vol. 10346. Lecture Notes in Computer Science. Springer, 2017, pp. 69–89.

[37] P. Kirchner. Improved Generalized Birthday Attack. Cryptology ePrint Archive, Paper 2011/377. 2011. https://eprint.iacr.org/2011/377.

[38] E. Kirshanova. Improved Quantum Information Set Decoding. PQCrypto. Vol. 10786. Lecture Notes in Computer Science. Springer, 2018, pp. 507–527.

[39] P. J. Lee, E. F. Brickell. An Observation on the Security of McEliece's Public-Key Cryptosystem. EUROCRYPT. Vol. 330. Lecture Notes in Computer Science. Springer, 1988, pp. 275–280.

[40] É. Levieil, P.-A. Fouque. An Improved LPN Algorithm. SCN. Vol. 4116. Lecture Notes in Computer Science. Springer, 2006, pp. 348–359.

[41] Y. Li, R. H. Deng, X. Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. IEEE Trans. Inf. Theory. Vol. 40, Num. 1 (1994), pp. 271–273.

[42] V. Lyubashevsky, C. Peikert, O. Regev. On Ideal Lattices and Learning with Errors over Rings. EUROCRYPT. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 1–23.

[43] A. May, A. Meurer, E. Thomae. Decoding Random Linear Codes in $\tilde{\mathcal{O}}(2^{0.054n})$. ASIACRYPT. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, pp. 107–124.

[44] A. May, I. Ozerov. On Computing Nearest Neighbors with Applications to Decoding of Binary Linear Codes. EUROCRYPT (1). Vol. 9056. Lecture Notes in Computer Science. Springer, 2015, pp. 203–228.

[45] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report. Vol. 44 (1978), pp. 114–116.

[46] C. Aguilar Melchor et al. Hamming Quasi-Cyclic (HQC) – Fourth round version (Updated version 01/10/2022). https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/HQC-Round4.zip. 2022-10. (2024-03-05 閲覧).

[47] S. Narisada, K. Fukushima, S. Kiyomoto. Multiparallel MMT: Faster ISD Algorithm Solving High-Dimensional Syndrome Decoding Problem. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. Vol. 106, Num. 3 (2023), pp. 241–252.

[48] S. Narisada, S. Uemura, H. Okada, H. Furue, Y. Aikawa, K. Fukushima. Solving McEliece-1409 in One Day – Cryptanalysis with the Improved BJMM Algorithm. ISC (2). Vol. 15258. Lecture Notes in Computer Science. Springer, 2024, pp. 3–23.

[49] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. Problemy Upravleniia i Teorii Informatsii (Problems of Control and Information Theory). Vol. 15, Num. 2 (1986), pp. 157–166.

[50] NIST. Module-Lattice-Based Key-Encapsulation Mechanism Standard. NIST FIPS 203, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf. 2024-08.

[51] S. Perriello, A. Barenghi, G. Pelosi. A Complete Quantum Circuit to Solve the Information Set Decoding Problem. QCE. IEEE, 2021, pp. 366–377.

[52] S. Perriello, A. Barenghi, G. Pelosi. Improving the Efficiency of Quantum Circuits for Information Set Decoding. ACM Transactions on Quantum Computing. Vol. 4, Num. 4 (2023). https://doi.org/10.1145/3607256.

[53] E. Prange. The use of information sets in decoding cyclic codes. IRE Trans. Inf. Theory. Vol. 8, Num. 5 (1962), pp. 5–9.

[54] J. Stern. A method for finding codewords of small weight. Coding Theory and Applications. Vol. 388. Lecture Notes in Computer Science. Springer, 1988, pp. 106–113.

[55] B. Zhang, L. Jiao, M. Wang. Faster Algorithms for Solving LPN. EUROCRYPT (1). Vol. 9665. Lecture Notes in Computer Science. Springer, 2016, pp. 168–195.

[56] 電子情報通信学会. 知識ベース 知識の森 1 群 (信号・システム) 2 編 (符号理論). https://www.ieice-hbkb.org/portal/01-2/01_02/. (2024-03-05 閲覧).

第 5 章

多変数多項式に基づく暗号技術

多変数公開鍵暗号（Multivariate Public Key Cryptosystems）における暗号方式の特徴は，有限体上の多変数多項式を用いた連立方程式

$$\begin{cases} p_1(x_1, x_2, \ldots, x_n) = 0, \\ p_2(x_1, x_2, \ldots, x_n) = 0, \\ \qquad\qquad \vdots \\ p_m(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

の求解問題（MP 問題）を解く計算の困難性が安全性の根拠として必要ということである。連立線形方程式は多項式時間で求解可能であるから，多変数公開鍵暗号に現れる MP 問題における多項式の最大次数は 2 次以上に限定される。本報告書では，多変数公開鍵暗号の多くの方式で採用されている双極型システムを中心に解説する。

5.1 多変数多項式に基づく暗号技術の安全性の根拠となる問題

$\mathbb{F}_q$ で位数 q の有限体を表し，$\mathbf{x} = (x_1, x_2, \ldots, x_n)$ で（代数的に独立な）変数の集合を表すものとする。$\mathbf{x}$ に関する $\mathbb{F}_q$ 上の多変数多項式の組，すなわち，多変数多項式 $p_i(\mathbf{x})$ $(i = 1, \ldots, m)$ により，$P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ と表されるものを（$\mathbb{F}_q$ 上の）多変数多項式系と呼ぶことにする。この多変数多項式系 $P(\mathbf{x})$ は代入評価により，$\mathbb{F}_q^n$ から $\mathbb{F}_q^m$ への写像を構成する。この（多変数多項式）写像を $P : \mathbb{F}_q^n \to \mathbb{F}_q^m$ と表すことにする。

5.1.1 MP 問題（MQ 問題）

MP 問題は次のように定義される。

MP 問題 多変数多項式系 $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ と $\mathbf{d} = (d_1, d_2, \ldots, d_m) \in \mathbb{F}_q^m$ に対して，変数 $\mathbf{x}$ に関する連立方程式

$$\begin{cases} p_1(x_1, x_2, \ldots, x_n) = d_1, \\ p_2(x_1, x_2, \ldots, x_n) = d_2, \\ \qquad\qquad \vdots \\ p_m(x_1, x_2, \ldots, x_n) = d_m \end{cases} \tag{5.1}$$

の解（が存在するなら）少なくとも 1 つ求めよ。

連立方程式 (5.1) の右辺の各 d_i を左辺に移項して $p_i(\mathbf{x})$ に吸収させることができるので，右辺を 0 として MP 問題を表現する場合もある。MP 問題において，$P(\mathbf{x})$ の全ての成分 $p_i(\mathbf{x})$ が 1 次以下となる場合，MP 問題は単に線形方程式を解く問題となり，ガウスの消去法などで m, n に関し多項式時間で求解することが可能である。よって，MP 問題

を考える場合は通常，各 $p_i(\mathbf{x})$ の次数は 2 以上であると仮定する。特に，$p_i(\mathbf{x})$ の次数が全て 2 となるとき，MP 問題は MQ 問題と呼ばれる。$\mathbb{F}_q = \mathbb{F}_2$ の場合，MQ 問題は NP 完全であることが知られている [24]。

MQ 問題を解くコンテストとして Fukuoka MQ challenge が知られている。扱われている MQ 問題は，有限体は $q = 2, 31, 256$ の 3 種類と m, n に関しては $m = 2n$，$n \approx 1.5m$ の 2 種類の計 6 種類である。投稿され解かれた問題の (m, n) の値の最大は表 5.1 のようになっている。

表 5.1: Fukuoka MQ challenge で解かれた MQ 問題のパラメータの最大値（2024/9/30 時点）

タイプ	I	II	III	IV	V	VI
$\mathbb{F}_q$	$\mathbb{F}_2$	$\mathbb{F}_{31}$	$\mathbb{F}_{256}$	$\mathbb{F}_2$	$\mathbb{F}_{31}$	$\mathbb{F}_{256}$
(m, n)	$m = 2n$	$m = 2n$	$m = 2n$	$n \approx 1.5m$	$n \approx 1.5m$	$n \approx 1.5m$
(m, n) の最大	$(166, 83)$	$(74, 37)$	$(76, 38)$	$(76, 114)$	$(20, 30)$	$(22, 33)$

5.1.2 MP 問題を解く計算の計算量

MP 問題 (5.1) は，右辺の $\mathbf{d}$ を左辺に移行し，$P(\mathbf{x})$ の中に吸収させてしまうことにより，

$$P(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})) = \mathbf{0}_m \tag{5.2}$$

の形の求解問題に変形できる。MP 問題 (5.2) に対する一般的な解き方として，総当たり法や XL [37], Gröbner 基底攻撃 [15] が知られている。

XL は，$(\prod_{j=1}^{k} x_{i_j})\, p_j(\mathbf{x}) = 0$ $(j = 1, \ldots, m)$ の形の方程式をたくさん集め，連立線型方程式の簡約操作を用いて MP 問題の解を求める。MP 問題の解の x_n の値を求める基本的な手順は以下のようになる。

1. $(\prod_{j=1}^{k} x_{i_j})\, p_j(\mathbf{x}) = 0$ $(j = 1, \ldots, m)$ の形の方程式をたくさん集める。
2. これらの方程式内に現れる単項式を新たな変数で置き直し，連立線型方程式を立てる。
3. 立てた連立線型方程式を簡約化することで，x_n^ℓ $(\ell = 0, 1, 2, \ldots)$ 以外の変数を消去する。
4. 得られた 1 変数 x_n に関する多項方程式を解いて x_n の値を求める。

この手順で得られた x_n の値を (5.2) に代入すると，$x_1, \ldots, x_{n-1}$ に関する MP 問題が得られる。この MP 問題に対して，上の手順と同様のことを行うと，今度は x_{n-1} の値を得ることができる。これを繰り返せば，最終的に MP 問題の解のすべての成分が得られる。十分大きい正の整数 D を取り，D 次以下の $(\prod_{j=1}^{k} x_{i_j})\, p_j(\mathbf{x}) = 0$ $(j = 1, \ldots, m)$ の形の方程式をすべて集めると，連立線型方程式が退化し，必ず解を持つようにすることができる。

Gröbner 基底攻撃は，イデアルの Gröbner 基底 [1] を計算して，MP 問題の解を求める。MP 問題 (5.2) の $p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})$ の列を延長するように $p_{m+1}(\mathbf{x}) = x_1^q - x_1, \ldots, p_{m+n}(\mathbf{x}) = x_n^q - x_n$ とおき，イデアル $I \subset \mathbb{F}_q[\mathbf{x}]$ を $I = \langle p_1(\mathbf{x}), \ldots, p_{m+n}(\mathbf{x}) \rangle$ とおく。イデアル I の（ある項順序に関する）Gröbner 基底が計算できたとして，それを $g_1(\mathbf{x}), \ldots, g_\ell(\mathbf{x}) \in \mathbb{F}_q[\mathbf{x}]$ とすると，MP 問題 (5.2) の解集合と，方程式 $(g_1(\mathbf{x}), \ldots, g_\ell(\mathbf{x})) = \mathbf{0}_\ell$ の解集合は一致する。項順序を辞書式順序にした場合，I の Gröbner 基底は，

$$g_1(x_1, \ldots, x_n), \ldots, g_{i_2-1}(x_1, \ldots, x_n), g_{i_2}(x_2, \ldots, x_n), \ldots, g_{i_3}(x_2, \ldots, x_n), \ldots, g_{\ell-1}(x_{n-1}, x_n), g_\ell(x_n)$$

という風に，i が大きくなるにつれ，g_i の変数の個数が（広義単調に）減るという形にできる。すると，$g_\ell(x_n) = 0$ を解いて x_n の値を求めることができ，さらに，$g_{\ell-1}(x_{n-1}, x_n) = 0$ などに求めた x_n の値を代入することで，x_{n-1} に関する 1 変数の多項方程式が得られ，x_{n-1} の値を求めることができる。これを繰り返すことで，すべての x_i の値が

特定でき，MP 問題 (5.2) の解を求められる。これが Gröbner 基底攻撃の基本戦略である。Gröbner 基底の効率的計算方法としては，F4/F5 アルゴリズム [18, 19] が有名である。

Gröbner 基底攻撃と XL の攻撃計算量は，アルゴリズム内に現れる（最も計算が重い）連立線型方程式の簡約操作の計算量で見積もられる。よって，攻撃計算量を求めるには，アルゴリズム内に現れる行列のサイズを知る必要があるが，それには攻撃アルゴリズム中に現れる多項式の次数の上限を見積もる必要がある。この上限の見積もり方について説明する。各 $p_i(\mathbf{x})$ $(i = 1, \ldots, m+n)$ に対し，その最高次斉次部分を $p_i^h(\mathbf{x})$ （d_i 次斉次多項式）と表し，$\mathbb{F}_q[\mathbf{x}]$ の斉次イデアル J を

$$J = \langle p_1^h(\mathbf{x}), \ldots, p_{m+n}^h(\mathbf{x}) \rangle$$

で定める。$d \geq 0$ に対し， $\mathbb{F}_q[\mathbf{x}]_d$ で d-次斉次多項式のなす $\mathbb{F}_q[\mathbf{x}]$ の部分ベクトル空間を表し，$J_d := J \cap \mathbb{F}_q[\mathbf{x}]_d$ とする。次数環 $\mathbb{F}_q[\mathbf{x}]/J = \bigoplus_{d=0}^{\infty} \mathbb{F}_q[\mathbf{x}]_d/J_d$ の Hilbert 級数は

$$\mathrm{HS}_{\mathbb{F}_q[\mathbf{x}]/J}(t) = \sum_{d=0}^{\infty} \dim_{\mathbb{F}_q}(\mathbb{F}_q[\mathbf{x}]_d/J_d)\, t^d \in \mathbb{Z}[[t]] \quad \text{(形式的べき級数)}$$

で定義される。J の Krull-次元が 0，すなわち，J が $\mathbb{F}_q[\mathbf{x}]$ の極大イデアルとなるとき，$\mathrm{HS}_{\mathbb{F}_q[\mathbf{x}]/J}(t)$ は多項式となる。このとき，$d_{\mathrm{reg}} = \deg(\mathrm{HS}_{\mathbb{F}_q[\mathbf{x}]/J}(t)) + 1$ とおき，これを正則性の次数 (degree of regularity) と呼ぶ。これ以外にも，Gröbner 基底計算と関係のある不変量として，solving degree d_{sol} や first fall degree d_{ff} などが存在する [12, 17]。これらの不変量 d_{reg}，d_{sol}，d_{ff} はいずれも Gröbner 基底計算中に現れる多項式の次数の上限を評価する値である。一般に，これらの不変量を求めることは Gröbner 基底計算と同程度困難であろうと考えられている。d をこれら不変量のうちの 1 つとしたとき，Gröbner 基底攻撃の計算量は以下のようになる [5]：

$$\mathcal{O}\left(\binom{n+d}{d}^{\omega} \right). \tag{5.3}$$

ここで，$2 \leq \omega \leq 3$ は行列の簡約操作のアルゴリズムにより定まる定数である。

任意の $S(t) \in \mathbb{Z}[[t]]$ に対し，$[S(t)]_+ \in \mathbb{Z}_{>0}[[t]]$ で，$S(t)$ の最初に現れる非正係数の次数以降（この項も含む）を切り捨てた多項式を表すことにする。もし，

$$\mathrm{HS}_{\mathbb{F}_q[\mathbf{x}]/J}(t) = \left[\frac{\prod_{i=1}^{m+n}(1-t^{d_i})}{(1-t)^n} \right]_+ = \left[\left(\prod_{i=1}^{n}(1-t^{d_i}) \right) \left(\frac{1-t^q}{1-t} \right)^n \right]_+$$

を満たすならば，$p_1(\mathbf{x}), \ldots, p_{m+n}(\mathbf{x})$ は半正則であるという。任意の m, n に対して, $p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})$ の係数をランダムに選ぶと，多くの場合に $p_1(\mathbf{x}), \ldots, p_{m+n}(\mathbf{x})$ は半正則となることが実験的に知られている。半正則であれば，正則性の次数 d_{reg} は容易に計算可能である。

XL で解く連立線形方程式の行列部分は疎行列である。実際，$p_i(\mathbf{x})$ $(i = 1, \ldots, m)$ が含む単項式の個数の最大を L とすると，行列の各行の非零成分の個数も L 個以下となる。従って，

$$d_{\mathrm{XL}} = \deg\left(\left[\frac{\prod_{i=1}^{m}(1-t^{d_i})}{(1-t)^{n+1}} \right]_+ \right) + 1$$

とおくと，q がある程度大きい場合，XL の計算量は以下のようになる：

$$\mathcal{O}\left(\binom{n+d_{\mathrm{XL}}}{d_{\mathrm{XL}}}^2 L \right).$$

5.1.3 MinRank 問題

MinRank 問題 正の整数 r と行列 $\mathrm{M}_1, \ldots, \mathrm{M}_k \in \mathbb{F}_q^{m \times n}$ に対し，$\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q$ で，$(\alpha_1, \ldots, \alpha_k) \neq (0, \ldots, 0)$ かつ

$$\mathrm{Rank}\left(\sum_{i=1}^{k} \alpha_i \mathrm{M}_i\right) \leq r$$

なるものを求めよ。(Rank(M) は行列 M のランクを表す。)

MinRank 問題は HFE_v^- や Rainbow など様々な方式の安全性に関わっている。また，MinRank 問題を解く計算の困難性をベースとした署名方式などがいくつか提案されている [14, 6, 34, 2]。MinRank 問題は MP 問題に帰着できることが知られている [29, 20, 4]。

例えば，Support minor modeling [4] では以下のように MinRank 問題が MP 問題に帰着される。$\alpha_1, \ldots, \alpha_k$ が MinRank 問題の解であるとするならば，$(\mathrm{S}, \mathrm{C}) \in \mathbb{F}_q^{m \times r} \times \mathbb{F}_q^{r \times n}$ で

$$\mathrm{S}\,\mathrm{C} = \sum_{i=1}^{k} \alpha_i \mathrm{M}_i \tag{5.4}$$

なるものが存在する。$\mathbf{r}_j$ を $\sum_{i=1}^{k} \alpha_i \mathrm{M}_i$ の第 j 行とすると，(5.4) より $\mathbf{r}_j$ は C の行ベクトルが張る空間に属する。よって，行列 $\mathrm{C}'_j \in \mathbb{F}_q^{(r+1) \times n}$ を

$$\mathrm{C}'_j = \begin{pmatrix} \mathbf{r}_j \\ \mathrm{C} \end{pmatrix}$$

で定めると，各 $j = 1, \ldots, m$ に対して，$\mathrm{Rank}\,\mathrm{C}'_j \leq r$ を満たす。従って，C'_j の任意の $(r+1) \times (r+1)$ 小行列の行列式 $= 0$ という関係式が得られるが，このような関係式は j と小行列を動かすことにより $m \begin{pmatrix} n \\ r+1 \end{pmatrix}$ 個存在する。r 個の元からなる $T\,(\subset \{1, 2, \ldots, n\})$ に対して，T に属する列番号からなる C の $r \times r$ 小行列を C_T と表し，さらにその行列式を c_T と表すと，C'_j の任意の $(r+1) \times (r+1)$ 小行列の行列式は，$\alpha_1, \ldots, \alpha_k$ と c_T ($T \subset \{1, \ldots, m\}$, T の元の個数は r) に関する多項式で表すことができる。これらの変数の個数は $k + \begin{pmatrix} n \\ r \end{pmatrix}$ である。つまり，MinRank 問題は $k + \begin{pmatrix} n \\ r \end{pmatrix}$ 個の変数の $m \begin{pmatrix} n \\ r+1 \end{pmatrix}$ 個の方程式からなる MP 問題に帰着される。

5.1.4 IP 問題，EIP 問題

IP（Isomorphism of Polynomials）問題は以下のように定義される。

IP 問題 S, T をそれぞれ，$\mathbb{F}_q^n$, $\mathbb{F}_q^m$ 上のアフィン同型写像とする。多変数多項式系 $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ に対し，多変数多項式系 $\tilde{P}(\mathbf{x})$ を合成により，$\tilde{P}(\mathbf{x}) = T \circ P(\mathbf{x}) \circ S$ で定める。このとき，$P(\mathbf{x})$, $\tilde{P}(\mathbf{x})$ の情報から S, T を求めよ。

IP 問題において，S や T の行列成分やベクトル成分をすべて独立な変数と見た場合，等式 $\tilde{P}(\mathbf{x}) = T \circ P(\mathbf{x}) \circ S$ は連立多項式方程式と見ることができる。すなわち，IP 問題は MP 問題に変換される。

多変数多項式系のクラス $\mathcal{C}$ を 1 つ固定する。ここで多変数多項式系のクラスとは多変数多項式系の集合 $\mathbb{F}_q[\mathbf{x}]^m$ の部分集合のことである。このとき，(クラス $\mathcal{C}$ に関する) EIP（Extended Isomorphism of Polynomials）問題は以下のように定義される。

EIP 問題 多変数多項式系 $\tilde{P}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ は，$\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T とクラス $\mathcal{C}$ に属する多変数多項式系 $P(\mathbf{x})$ により，$\tilde{P}(\mathbf{x}) = T \circ P(\mathbf{x}) \circ S$ で表されるとする。このとき，分解 $\tilde{P}(\mathbf{x}) = T' \circ P'(\mathbf{x}) \circ S'$ で，S', T' は $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像，$P'(\mathbf{x}) \in \mathcal{C}$ なるものを見つけよ。

$\mathcal{C} = \{P(\mathbf{x})\}$ に関する EIP 問題が通常の IP 問題であるから，EIP 問題は IP 問題の拡張である。5.2 節で述べるように，EIP 問題は双極型システムで構成される暗号化方式，署名方式の鍵復元攻撃に対する安全性に関わる。EIP 問題を解く方法はクラス $\mathcal{C}$ の取り方（あるいは方式）に依存する。

5.2 多変数多項式に基づく代表的な暗号方式の説明

5.2.1 双極型システム

IP 問題ベース [32] や MinRank 問題ベース [14, 2, 6, 34] の方式も存在するが，多変数公開鍵暗号の多くの方式が MP 問題をベースとして構成されている。中でも双極型システム [15] と呼ばれる構成方法が多く利用されているため，この構成方法について説明する。（1 次多項式で構成されてなくても）多変数多項式系 $P(\mathbf{x})$ によっては，多くの $\mathbf{d} \in \mathbb{F}_q^m$ に対して MP 問題が効率的に計算できる場合がある。例えば, $n = m$ とし, $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ が三角型多変数多項式系である，すなわち，

$$
\begin{aligned}
&p_1(\mathbf{x}) = x_1, \\
&p_2(\mathbf{x}) = x_2 + g_2(x_1) \quad (g_2(x_1) \in \mathbb{F}_q[x_1]), \\
&p_3(\mathbf{x}) = x_3 + g_3(x_1, x_2) \quad (g_2(x_1, x_2) \in \mathbb{F}_q[x_1, x_2]), \\
&\qquad \vdots \\
&p_m(\mathbf{x}) = x_m + g_m(x_1, \ldots, x_{m-1}) \quad (g_m(x_1, \ldots, x_{m-1}) \in \mathbb{F}_q[x_1, \ldots, x_{m-1}])
\end{aligned}
$$

の形で表されるとすると，任意の $\mathbf{d} \in \mathbb{F}_q^m$ に対して $P(\mathbf{x}) = \mathbf{d}$ の（唯 1 つの）解が，x_1 から逐次的に求められることが分かる。このことはすなわち，多変数多項式系のクラス $\mathcal{C}$ を三角型多変数多項式系の全体で定めると，任意の $P \in \mathcal{C}$ に対して，$P(\mathbf{x}) = \mathbf{d}$ $(\mathbf{d} \in \mathbb{F}_q^m)$ の解が効率的に計算可能ということである。

双極型システムでは，まず，MP 問題が効率的に計算できる多変数多項式系のクラス $\mathcal{C}_{\text{cent}}$ を見つけ固定する。（例えば，$\mathcal{C}_{\text{cent}}$ として三角型多変数多項式系の集合を取れる。）$G(\mathbf{x}) \in \mathcal{C}_{\text{cent}}$ と $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T をそれぞれ任意にとり，これらを合成した多変数多項式系 $F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ をトラップドア付き一方向関数として利用するのが，双極型システムのアイデアである。ただし，$F(\mathbf{x})$ が実際にトラップドア付き一方向関数となるかどうかは $\mathcal{C}_{\text{cent}}$ のとり方に依存する。

双極型システムの鍵生成は次のように行う。

鍵生成

1. $G(\mathbf{x}) \in \mathcal{C}_{\text{cent}}$ をランダムに選ぶ。
2. $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T をランダムに選ぶ。
3. $F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ とする。

このとき，公開鍵は $F(\mathbf{x})$，秘密鍵は $G(\mathbf{x}), S, T$ となる。$F(\mathbf{x})$ はその係数集合が公開鍵として保管される。また，$G(\mathbf{x})$ を（この方式の）**中心写像**とよぶ。中心写像のクラス $\mathcal{C}_{\text{cent}}$ は 2 次の多変数多項式系で構成されることが多い。これは，公開鍵長（や秘密鍵長）を出来るだけ小さくするためである。双極型システムは暗号化方式，署名方式両方の構成に用いることができる。

暗号化方式の暗号化・復号は次のように行う。

暗号化 平文 $M \in \mathbb{F}_q^n$ に対し，$C = F(M)$ を計算する。C が暗号文となる。

復号 暗号文 $C \in \mathbb{F}_q^m$ に対し，(1) $B_1 = T^{-1}(C)$, (2) $G(B_2) = B_1$ なる B_2 を計算, (3) $M' = S^{-1}(B_2)$ の順に計算する。M' が平文と一致する。

復号が成功するためには，$G(\mathbf{x})$（あるいは $F(\mathbf{x})$）が単射である必要がある。単射の条件を少し緩めて，「$G(\mathbf{x})$（あるいは $F(\mathbf{x})$）の逆像の個数が十分少ない」とすることもできる。この場合，M' が複数得られることになるので，ハッシュ値などを用いて平文 M と一致する M' を特定する。

双極型システムの署名方式の署名生成・検証は次のように行う。

署名生成 メッセージ（のハッシュ値） $M \in \mathbb{F}_q^m$ に対し，(1) $B_1 = T^{-1}(M)$, (2) $G(B_2) = B_1$ なる B_2 を計算, (3) $\sigma = S^{-1}(B_2)$ の順に計算する。σ が署名となる。

検証 署名 $\sigma \in \mathbb{F}_q^n$ に対し，$M' = F(\sigma)$ を計算する。$M = M'$ ならば署名を受理，それ以外は棄却する。

署名生成がいつでも実行できるためには，どのような $M \in \mathbb{F}_q^m$ に対しても，$B_2 = G^{-1}(B_1)$ の計算ができる，すなわち，$G(\mathbf{x})$（あるいは $F(\mathbf{x})$）が全射である必要がある。

双極型システムでは，中心写像のクラス $\mathcal{C}_{\text{cent}}$ の取り方を変えることで幅広い方式の構成が可能である。例えば，$\mathcal{C}_{\text{cent}} = \{$ 三角型多変数多項式系 $\}$ とすると暗号化方式が得られる。双極型システムにおいては，$\mathcal{C}_{\text{cent}}$ に関する EIP 問題がその安全性に大きく関わってくる。実際，EIP 問題を解けた場合，$F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ の代わりに分解 $F(\mathbf{x}) = T' \circ G'(\mathbf{x}) \circ S'$ を用いても，暗号化方式における復号および，署名方式における署名生成（偽造）が実行可能となる。EIP 問題の困難性はクラス $\mathcal{C}$ の選び方に依存するので，$\mathcal{C}$ の選び方に応じて個々に解析される必要がある。例えば，$\mathcal{C}_{\text{cent}} = \{$ 三角型多変数多項式系 $\}$ としたときの EIP 問題は効率的に解けることが知られている [25]。

双極型システムの代表的な構成法として，simple field 法と big field 法がある。Simple field 方式は中心写像の構成に $\mathbb{F}_q$ 以外の有限体を利用しない。Big field 方式は中心写像の構成に $\mathbb{F}_q$ の n 次拡大体 $\mathbb{F}_{q^n}$ を利用する。Big field 方式は中心写像を構成しやすいが，Gröbner 基底攻撃が効果的となる場合が多いという性質を持つ。5.2.3 節では，big field 方式の代表として署名方式 HFE および HFE$_v^-$，5.2.4 節では，simple field 方式の代表として署名方式 UOV について説明する。

5.2.2 双極型システムの modifier

Modifier [36, 15] は双極型システムの方式からその変種方式を構成する。様々な Modifier があり，それぞれに安全性を強化したり，効率性を向上させたりといった効果がある。以下では代表的な modifier を 4 つ紹介する。5.2.3 節で説明する HFEv^- 方式では，2 つの modifier（マイナス手法と External Perturbation）が利用されている。

5.2.2.1 マイナス手法 “ − ”

マイナス手法は，公開鍵 $F(\mathbf{x})$ のいくつかの成分を削除する方法である。すなわち，$F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ と表されるとき，r 個の成分を削除し，$\tilde{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_{m-r}(\mathbf{x}))$ を新たな公開鍵とする方式である。$F(\mathbf{x})$ に対して有効な秘密鍵復元攻撃がある場合でも，$\tilde{F}(\mathbf{x})$ は $F(\mathbf{x})$ よりも情報が欠落しているため，$\tilde{F}(\mathbf{x})$ に対しては同じ秘密鍵復元攻撃が適用できなくなる可能性があり，安全性強化につながる。暗号化方式では公開鍵の単射性が失われたり，復号が困難になるといった理由により，マイナス手法はあまり用いられない。署名方式では，$F(\mathbf{x})$ に対する署名生成を利用して，$\tilde{F}(\mathbf{x})$ に対する署名生成ができる。

5.2.2.2 プラス手法 “+”

プラス手法は，中心写像 $G(\mathbf{x})$ にランダムな多項式を成分として加える方法である。すなわち，中心写像 $G(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_m(\mathbf{x}))$ に対し，r 個のランダムな多項式 $g_{m+1}(\mathbf{x}), \ldots, g_{m+r}(\mathbf{x})$ を用意し，$\tilde{G}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_{m+r}(\mathbf{x}))$ を新たな中心写像とする方式である。中心写像に特定の構造を持たない多項式が混ざることにより，秘密鍵復元攻撃の成功率を下げることができる。署名方式では公開鍵の全射性が失われたり，署名生成が困難になるといった理由により，プラス手法はあまり用いられない。暗号化方式では，復号において $\tilde{G}(\mathbf{x})$ の逆写像を計算しなければならないが，その計算に $G(\mathbf{x})$ の逆写像計算が利用できる。

5.2.2.3 External Perturbation “v”

この modifier は，元々の変数 $\mathbf{x} = (x_1, \ldots, x_n)$ に新たな変数 $\mathbf{v} = (x_{n+1}, \ldots, x_{n+v})$（vinegar 変数）を加える方法である。この modifier は主に署名方式で利用される。署名方式を定める中心写像のクラスを $\mathcal{C}$ とする。新たな中心写像のクラス $\mathcal{C}'$ を多項式写像 $G(\mathbf{x}, \mathbf{v}) : \mathbb{F}_q^{n+v} \to \mathbb{F}_q^m$ で，任意の $\mathbf{v}_0 \in \mathbb{F}_q^v$ に対し，$G(\mathbf{x}, \mathbf{v}_0) \in \mathcal{C}$ なるもの全体として定める。すると，$G(\mathbf{x}, \mathbf{v}) \in \mathcal{C}'$ に対し，$G(\mathbf{x}, \mathbf{v}) = \mathbf{d}$ $(\mathbf{d} \in \mathbb{F}_q^m)$ の解が次のように得られる。

1. $\mathbf{v}_0 \in \mathbb{F}_q^v$ をランダムに選ぶ。
2. $G(\mathbf{x}, \mathbf{v}_0) = \mathbf{d}$ を $\mathbf{x}$ に関して解く。（解を $\mathbf{x}^*$ とする。）
3. $(\mathbf{x}, \mathbf{v}) = (\mathbf{x}^*, \mathbf{v}_0)$ を出力。

この計算を利用して，新たな署名方式が構成できる。Vinegar 変数は $\mathcal{C}$ とは無関係な変数なので追加することで安全性強化が期待できる。

5.2.2.4 Internal Perturbation “I”

この modifier は，中心写像 $G(\mathbf{x})$ にノイズを加えて安全性を強化する方法である。変数 $\mathbf{z} = (z_1, \ldots, z_w)$ と多項式写像 $H(\mathbf{z}) : \mathbb{F}_q^w \to \mathbb{F}_q^m$，および，アフィン写像 $S : \mathbb{F}_q^n \to \mathbb{F}_q^w$ を用意する。また，$H(\mathbf{z})$ の像 $W \subset \mathbb{F}_q^m$ が分かっており，W に属する元の個数は十分少ないと仮定する。新たな中心写像 $\tilde{G}(\mathbf{x}) : \mathbb{F}_q^n \to \mathbb{F}_q^m$ を $G(\mathbf{x}) + H(S(\mathbf{x}))$ で定める。このとき，$\tilde{G}(\mathbf{x}) = \mathbf{d}$ $(\mathbf{d} \in \mathbb{F}_q^m)$ の解が次のように得られる。

1. $\mathbf{w}_0 \in W$ をランダムに選ぶ。
2. 次の方程式の解 $\mathbf{x}^*$ を求める。

$$\begin{cases} G(\mathbf{x}) = \mathbf{d} - \mathbf{w}_0, \\ H(S(\mathbf{x})) = \mathbf{w}_0. \end{cases}$$

   もし，解が得られなかったら 1. に戻る。
3. $\mathbf{x}^*$ を出力。

よって，この $\tilde{G}(\mathbf{x})$ を中心写像として方式が構成できる。

5.2.3 HFE 方式，HFE v^- 方式

5.2.3.1 暗号化方式 HFE

$K = \mathbb{F}_{q^n}$ を $\mathbb{F}_q$ の n 次拡大体とし，$\mathbb{F}_q$-線形同型写像 $\phi : \mathbb{F}_q^n \xrightarrow{\sim} K$ を 1 つ固定する。D を正の整数として，K 上の 1 変数多項式

$$\mathcal{G}(X) = \sum_{0 \le i \le j}^{q^i+q^j \le D} \alpha_{i,j} X^{q^i+q^j} + \sum_{0 \le i}^{q^i \le D} \beta_i X^{q^i} + \gamma \qquad (\alpha_{i,j},\ \beta_i,\ \gamma \in K)$$

をとる。$\mathcal{G}(X)$ の形の 1 変数多項式は HFE 多項式と呼ばれる。このとき，多変数多項式写像 $G : \mathbb{F}_q^n \to \mathbb{F}_q^n$ を $G = \phi^{-1} \circ \mathcal{G} \circ \phi$ と定めると，対応する多変数多項式系 $G(\mathbf{x})$ の成分は全て 2 次多項式となる。$\mathbf{d} \in \mathbb{F}_q^n$ に対して，$G(\mathbf{x}) = \mathbf{d}$ が解を持つならば，この解は全て効率的に計算することができる。実際，次の手順で計算できる。

1. $B = \phi(\mathbf{d}) \in K$ を計算する。
2. $A = \mathcal{G}^{-1}(B)$ を Cantor-Zassenhaus アルゴリズムなどの因数分解アルゴリズムを用いて計算する。
3. $\phi^{-1}(A)$ を計算する。

但し，Step 2 の計算が効率的に実行できるためには D をある程度小さくとる必要がある。上記のことを踏まえると，$\alpha_{i,j},\ \beta_i,\ \gamma \in K$ を動かしてできる $G(\mathbf{x})$ のなすクラス $\mathcal{C}_{\mathrm{HFE}}$ に対し，$\mathcal{C}_{\mathrm{cent}} = \mathcal{C}_{\mathrm{HFE}}$ の双極型システムは暗号化方式を構成する。この暗号化方式を HFE [32] と呼ぶ。HFE 自体は 1999 年，Kipnis と Shamir により効果的な攻撃が発見されている [29]。その後，HFE から派生した変種方式がいくつか提案されており，以下の HFE_v^- もその 1 つである。

5.2.3.2 署名方式 HFE_v^-

HFE_v^- [32, 33] は，暗号化方式 HFE を署名方式に応用したものである。HFE と同様に $\mathbb{F}_q$ の n 次拡大体 $K = \mathbb{F}_{q^n}$ をとり，$\mathbb{F}_q$-線形同型写像 $\phi : \mathbb{F}_q^n \to K$ を固定する。正の整数 a $(a < n)$ と v を固定する。まず，$\mathcal{G}(X)$ は次のように変更される。

$$\mathcal{G}(X) = \sum_{0 \le i \le j}^{q^i+q^j \le D} \alpha_{i,j} X^{q^i+q^j} + \sum_{0 \le i}^{q^i \le D} \beta_i(x_{n+1}, \ldots, x_{n+v}) X^{q^i} + \gamma(x_{n+1}, \ldots, x_{n+v}) \qquad (\alpha_{i,j} \in K). \tag{5.5}$$

ここで, $\beta_i(x_{n+1}, \ldots, x_{n+v}), \gamma(x_{n+1}, \ldots, x_{n+v})$ は共に $\mathbb{F}_q^v$ から K への多項式写像であり, $\beta_i(x_{n+1}, \ldots, x_{n+v})$ は 1 次多項式, $\gamma(x_{n+1}, \ldots, x_{n+v})$ は 2 次多項式である。多変数多項式系 $G(\mathbf{x})$ は, 多変数多項式写像 $G = \phi^{-1} \circ \mathcal{G} \circ (\phi \times id_v) : \mathbb{F}_q^{n+v} \to \mathbb{F}_q^n$ により定める。$\alpha_{i,j} \in K$ と $\beta_i(x_{n+1}, \ldots, x_{n+v})$, $\gamma(x_{n+1}, \ldots, x_{n+v})$ を動かしてできる $G(\mathbf{x})$ のなすクラスを $\mathcal{C}_{\mathrm{HFE}_v^-}$ と定める。基本的には，これを $\mathcal{C}_{\mathrm{cent}} = \mathcal{C}_{\mathrm{HFE}_v^-}$ として構成される双極型システムを考えるのであるが，双極型システムを若干変更する。S は $\mathbb{F}_q^{n+v}$ 上のアフィン同型写像のままでよいが，T は $\mathbb{F}_q^n$ から $\mathbb{F}_q^{n-a}$ への最大ランクのアフィン写像と変更する。公開鍵は通常の双極型システムと同じように，$F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ と定める。よって，F は $\mathbb{F}_q^{n+v}$ から $\mathbb{F}_q^{n-a}$ への多変数多項式写像となる。メッセージ（のハッシュ値） $M \in \mathbb{F}_q^{n-a}$ に対する署名 $\sigma = F^{-1}(M)$ は以下のよう計算される。

1. $\mathbf{c} = T^{-1}(M) \in \mathbb{F}_q^n$ （の 1 つ）を計算する。
2. $B = \phi(\mathbf{c}) \in K$ を計算する。
3. $B' \in \mathbb{F}_q^v$ をランダムに選び， $A = \mathcal{G}^{-1}(B \,\|\, B')$ を Cantor-Zassenhaus アルゴリズムなどを用いて計算する。$\mathcal{G}^{-1}(B \,\|\, B')$ が存在しない場合は，B' の選択からやり直す。
4. $\mathbf{e} = \phi^{-1}(A)$ を計算する。
5. $\sigma = S^{-1}(\mathbf{e})$ を計算する。

HFE_v^- と同じ構造を持つ署名方式 GeMSS [13] は NIST PQC 標準化プロジェクト 第 3 ラウンドに選ばれたが，効率的な攻撃法が提案されたため [35]，第 4 ラウンドに進むことはできなかった。

5.2.4 署名方式 UOV

5.2.4.1 UOV の概要

UOV [28, 10] は，双極型システムを用いた署名方式である。UOV の中心写像は，決まったいくつかの変数に値を代入することで 1 次式に変形でき，連立線形方程式の求解手法を用いて，効率的に署名生成が可能である。v, m を正の整数とし，$n = v + m$ とする。2 次多項式からなる多変数多項式系 $G(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_m(\mathbf{x}))$ を次の形で与える。

$$g_k(\mathbf{x}) = \sum_{\substack{1 \le i \le v \\ v+1 \le j \le n}} \alpha_{i,j}^{(k)} x_i x_j + \sum_{1 \le i \le j \le v} \beta_{i,j}^{(k)} x_i x_j + \sum_{1 \le i \le n} \gamma_i^{(k)} x_i + \eta^{(k)} \quad (k = 1, \ldots, m).$$

ここで，$\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in \mathbb{F}_q$ である。$G(\mathbf{x})$ の形で定義される多変数多項式写像を **UOV 多項式写像**と呼ぶ。$g_k(\mathbf{x})$ の 2 次多項式部分を $\tilde{g}_k(\mathbf{x})$ とすると，

$$\tilde{g}_k(\overbrace{0, \ldots, 0}^{v}, \overbrace{*, \ldots, *}^{m}) = 0 \tag{5.6}$$

となることが，UOV 多項式写像の特徴である。$x_1, \ldots, x_v$ をビィネガ変数，$x_{v+1}, \ldots, x_n$ をオイル変数と呼ぶ。$G(\mathbf{x})$ のヴィネガ変数に（ランダムな）値を代入すると，(5.6) によりオイル変数に関する 1 次式が得られる。$G(\mathbf{x})$ の逆写像は，連立線形方程式の求解手法を用いて効率的に計算できる。具体的に，任意の $\mathbf{c} = (c_1, \ldots, c_m) \in \mathbb{F}_q^m$ に対し，$\mathbf{b} = G^{-1}(\mathbf{c})$（の一つ）が以下のように計算できる。

1. $b_1, \ldots, b_v \in \mathbf{F}_q$ をランダムにとる。
2. $g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})$ に $(x_1, \ldots, x_v) = (b_1, \ldots, b_v)$ を代入して得られる $x_{v+1}, \ldots, x_n$ に関する 1 次式をそれぞれ $\bar{g}_1(x_{v+1}, \ldots, x_n)$, $\ldots, \bar{g}_m(x_{v+1}, \ldots, x_n)$ とする。連立線形方程式
$$\begin{cases} \bar{g}_1(x_{v+1}, \ldots, x_n) = c_1 \\ \qquad\qquad \vdots \\ \bar{g}_m(x_{v+1}, \ldots, x_n) = c_m \end{cases}$$
の解を計算し，それを $b_{v+1}, \ldots, b_n$ と置く。もし解がなければ Step 1 に戻る。
3. $\mathbf{b} = (b_1, \ldots, b_n)$.

$\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in \mathbb{F}_q$ を動かしてできる $G(\mathbf{x})$ の集合を $\mathcal{C}_{\mathrm{UOV}}$ としたとき，$\mathcal{C}_{\mathrm{cent}} = \mathcal{C}_{\mathrm{UOV}}$ として構成される双極型システムの署名方式を UOV と呼ぶ。但し，通常の双極型システムで使用するアフィン同型写像 T は，UOV の安全性には貢献しないので必要ない。すなわち，秘密鍵は $G(\mathbf{x}) \in \mathcal{C}_{\mathrm{UOV}}$ と $\mathbb{F}_q^n$ 上のアフィン同型写像 S で，公開鍵は $F(\mathbf{x}) = G(\mathbf{x}) \circ S$ となる。$F(\mathbf{x})$ の成分の 2 次多項式部分を $\tilde{f}_1(\mathbf{x}), \ldots, \tilde{f}_m(\mathbf{x})$ とし，部分空間 $O\,(\subset \mathbb{F}_q^n)$ を

$$O = S^{-1}(\{(\overbrace{0, \ldots, 0}^{v}, \mathbf{a}) \in \mathbb{F}_q^n \mid \mathbf{a} \in \mathbb{F}_q^m\})$$

とすると，(5.6) により $\tilde{f}_i(\mathbf{o}) = 0 \ \ (i = 1, \ldots, m,\ \mathbf{o} \in O)$ を満たす。このような性質を持つ部分空間をオイル空間という。逆に，多変数 2 次多項式系 $H(\mathbf{x})$ が o 次元のオイル空間 $O\,(\subset \mathbb{F}_q^n)$ を持ち，$o \ge m$ を満たすならば，$H(\mathbf{x})$ は UOV の公開鍵として使用できる。

5.2.4.2 UOV の公開鍵長の削減

双極型システムの公開鍵 $F(\mathbf{x})$ は，通常，その係数集合の形で記述され，その中でも，2 次多項式部分の係数集合が公開鍵の大部分を占める。$P(\mathbf{x})$ を多変数 2 次多項式系とし，$\tilde{p}_1(\mathbf{x}), \ldots, \tilde{p}_m(\mathbf{x})$ をその 2 次多項式部分とすると，ある行列 $\mathrm{P}_1, \ldots, \mathrm{P}_m \in \mathbb{F}_q^{n\times n}$ により，

$$\tilde{p}_i(\mathbf{x}) = \mathbf{x}\,\mathrm{P}_i\,\mathbf{x}^\top \quad (i = 1, \ldots, m)$$

と表すことができる。一般に，行列 $\mathrm{P} = (p_{ij}) \in \mathbb{F}_q^{h\times h}$ に対し，上三角行列 $\mathrm{upper}(\mathrm{P}) = (c_{ij}) \in \mathbb{F}_q^{h\times h}$ を

$$c_{ij} = \begin{cases} p_{ii} & i = j, \\ p_{ij} + p_{ji} & i < j, \\ 0 & i > j \end{cases}$$

で定義すると，$\tilde{p}_i(\mathbf{x}) = \mathbf{x}\,\mathrm{P}_i\,\mathbf{x}^\top = \mathbf{x}\,\mathrm{upper}(\mathrm{P}_i)\,\mathbf{x}^\top$ が成り立つ。特に，P_i はすべて上三角行列で選ぶことができる。

UOV の中心写像 $G(\mathbf{x})$ の 2 次多項式部分に対応する上三角行列を $\mathrm{G}_1, \ldots, \mathrm{G}_m$ とし，公開鍵 $F(\mathbf{x})$ の 2 次多項式部分に対応する上三角行列を $\mathrm{F}_1, \ldots, \mathrm{F}_m$ とすると，

$$\mathrm{G}_i = \begin{pmatrix} \mathrm{G}_{i,1} & \mathrm{G}_{i,2} \\ \mathbf{0}_{o\times v} & \mathbf{0}_{o\times o} \end{pmatrix}, \quad \mathrm{F}_i = \begin{pmatrix} \mathrm{F}_{i,1} & \mathrm{F}_{i,2} \\ \mathbf{0}_{o\times v} & \mathrm{F}_{i,3} \end{pmatrix} \quad (\mathrm{G}_{i,1}, \mathrm{F}_{i,1} \in \mathbb{F}_q^{v\times v},\ \mathrm{G}_{i,2}, \mathrm{F}_{i,2} \in \mathbb{F}_q^{v\times o},\ \mathrm{F}_{i,3} \in \mathbb{F}_q^{o\times o})$$

の形で表すことができる。ここで，$\mathrm{G}_{i,1}, \mathrm{G}_{i,3}, \mathrm{F}_{i,1}, \mathrm{F}_{i,3}$ は上三角行列である。今，アフィン同型写像 S の線形部分を表す行列 S （線形写像は $\mathbf{x} \mapsto \mathbf{x}\,\mathrm{S}$ の形）が

$$\mathrm{S} = \begin{pmatrix} \mathbf{I}_v & \mathbf{0}_{v\times o} \\ \mathrm{S}_0 & \mathbf{I}_o \end{pmatrix} \quad (\mathrm{S}_0 \in \mathbb{F}_q^{o\times v})$$

の形で表される場合に限定する。（但し，$\mathbf{I}_\ell$ は ℓ 次単位行列を表す。）すると，$\mathrm{F}_i = \mathrm{upper}(\mathrm{S}\,\mathrm{G}_i\,\mathrm{S}^\top)$ となるので，次の関係が成り立つ。

$$\begin{aligned} &\cdot\ \mathrm{G}_{i,1} = \mathrm{F}_{i,1},\ \ \mathrm{G}_{i,2} = \mathrm{F}_{i,2} - (\mathrm{F}_{i,1} + \mathrm{F}_{i,1}^\top)\,\mathrm{S}_0^\top, \\ &\cdot\ \mathrm{F}_{i,3} = \mathrm{upper}(-\mathrm{S}_0\,\mathrm{F}_{i,1}^\top\,\mathrm{S}_0^\top + \mathrm{S}_0\,\mathrm{F}_{i,2}) = \mathrm{upper}(-\mathrm{S}_0\,\mathrm{F}_{i,1}\,\mathrm{S}_0^\top + \mathrm{S}_0\,\mathrm{F}_{i,2}). \end{aligned} \tag{5.7}$$

これより，$\mathrm{F}_{i,1}$ は任意の上三角行列, $\mathrm{F}_{i,2}$ は任意の行列で選べることが分かる。そこで，$\mathrm{F}_{i,1}, \mathrm{F}_{i,2}$ の成分すべてを公開鍵として記述する代わりに，$\mathrm{F}_{i,1}, \mathrm{F}_{i,2}$ を疑似乱数生成器を用いて構成することにして，そのシードのみを公開鍵として記述することにより公開鍵長を削減できる。このようにして，UOV の公開鍵の 2 次多項式部分は，シードと (5.7) で求められた $\mathrm{F}_{i,3}$ $(i = 1, \ldots, m)$ だけで記述できる。

一般に，双極型システムの公開鍵長は n, m に関して，$\mathcal{O}(mn^2)$ の増大度を持ち，大きくなりやすい。UOV では，上の公開鍵の記述方法を使うことにより，公開鍵長の増大度は $\mathcal{O}(m^3)$ となり，UOV のパラメータが $2m < n$ で選ばれることを踏まえると，一般の双極型システムの公開鍵の記述方法よりも，公開鍵長を削減できる。この削減方法は，5.3 節で記述する（UOV の変種である）QR-UOV や MAYO にも利用されている。

5.2.4.3 署名方式 Rainbow

署名方式 Rainbow [16] は UOV を多層化して作られる。正の整数 $t, v_1, o_1, \ldots, o_t$ に対し，$v_{i+1} = v_i + o_i$ により，$v_2, \ldots, v_{t+1}$ を順次定める。また，$i = 1, \ldots, t$ に対し，$S_i = \{1, \ldots, v_i\}$, $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ とおく。S_i の元の個数は v_i で，O_i の元の個数は o_i である。変数の個数を $n = v_{t+1}$，式数を $m = n - v_1$ とする多変数多項式系 $G(\mathbf{x}) = (g_{v_1+1}(\mathbf{x}), \ldots, g_n(\mathbf{x}))$ を次の形で与える：

$$g_k(x_1, \ldots, x_n) = \sum_{i\in O_h, j\in S_h} \alpha_{i,j}^{(k)} x_i x_j + \sum_{i,j\in S_h,\, i\le j} \beta_{i,j}^{(k)} x_i x_j + \sum_{i\in S_{h+1}} \gamma_i^{(k)} x_i + \eta^{(k)} \quad (k = v_1{+}1, \ldots, n).$$

但し，h は k が属する層番号，すなわち，“$k \in O_h$” で定まる整数 $1 \leq h \leq t$ である。$\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in \mathbb{F}_q$ を動かしてできる $G(\mathbf{x})$ のなすクラスを $\mathcal{C}_{\text{Rainbow}}$ と定め，これを Rainbow の中心写像のクラスとする。任意の $\mathbf{c} = (c_1, \ldots, c_m) \in \mathbb{F}_q^m$ に対し，$\mathbf{b} = G^{-1}(\mathbf{c})$（の一つ）が以下のように計算できる。

1. $b_1, \ldots, b_{v_1} \in \mathbb{F}_q$ をランダムにとる。
2. $h = 1, 2, \ldots, t$ に対し，以下を実行：

   $g_{v_h+1}(\mathbf{x}), \ldots, g_{v_{h+1}}(\mathbf{x})$ に $(x_1, \ldots, x_{v_h}) = (b_1, \ldots, b_{v_h})$ を代入して得られる $x_{v_h+1}, \ldots, x_{v_{h+1}}$ に関する 1 次式をそれぞれ $\bar{g}_{v_h+1}(x_{v_h+1}, \ldots, x_{v_{h+1}}), \ldots, \bar{g}_{v_{h+1}}(x_{v_h+1}, \ldots, x_{v_{h+1}})$ とする。連立線形方程式

   $$\begin{cases} \bar{g}_{v_h+1}(x_{v_h+1}, \ldots, x_{v_{h+1}}) = c_{v_h+1} \\ \qquad\qquad \vdots \\ \bar{g}_{v_{h+1}}(x_{v_h+1}, \ldots, x_{v_{h+1}}) = c_{v_{h+1}} \end{cases}$$

   の解を計算し，それを $b_{v_h+1}, \ldots, b_{v_{h+1}}$ と置く。もし解がなければ Step 1 に戻る。
3. $\mathbf{b} = (b_1, \ldots, b_n)$.

上のアルゴリズムの Step 2 の第 h ループ内では，S_h を添字とする変数をヴィネガ変数，O_h を添字とする変数をオイル変数とし，UOV の中心写像と同じ逆写像計算を行っている。このことから，Rainbow を UOV の多層化と見ることができる。Rainbow は NIST PQC 標準化プロジェクト 第 3 ラウンドに選ばれたが，Rainbow の EIP 問題を解くことにより，小さなサイズの UOV への攻撃に帰着する攻撃が提案され [7, 11]，その結果，level I, III, V として提案されていたパラメータがその安全性レベルに到達しないことになり（安全性レベル 143 bits が 69 bits に，207 bits が 157bits に，272 bits が 206 bits に下がった），第 4 ラウンドに進むことはできなかった。

5.2.5 MPC-in-the-Head

5.2.5.1 秘匿マルチパーティ計算

Ishai らによって導入された MPC-in-the-Head [27] は，秘匿マルチパーティ計算からゼロ知識証明を構成し，さらに Fiat-Shamir 変換により署名方式が構成できる。本来，MPC-in-the-Head の枠組みは広く，多変数公開鍵暗号に限定された技術ではないが，多変数公開鍵暗号においては，MQ 問題に付随する MPC-in-the-Head と，MinRank 問題に付随する MPC-in-the-Head が重要であるため，この 2 つの場合に限定して説明する。

$\mathcal{R} \subset \{0,1\}^* \times \{0,1\}^*$ を関係とする。命題 a に対して，$(a, x) \in \mathcal{R}$ であるとき，x は a の証拠（witness）であるという。ここでは，$\mathcal{R}$ として，MQ 問題（あるいは，MinRank 問題）のインスタンスを命題とし，その解を証拠とする関係に限定する。N 組のパーティ $\mathcal{P}_1, \ldots, \mathcal{P}_N$ が存在するとする。加法構造を持つ代数系の元 b に対し，分散 $[\![b]\!]$ は

$$[\![b]\!] = ([\![b]\!]_1, \ldots, [\![b]\!]_N) \text{ であり，} [\![b]\!]_1 + \cdots + [\![b]\!]_N = b$$

なるものを意味するとする。命題 a に対し，以下のような性質を持つ秘匿マルチパーティ計算 f を考える。

- 秘密情報 x の分散 $[\![x]\!]$ に対し，$\mathcal{P}_j$ は $[\![x]\!]_j$ を入力として受け取る。
- f は ‘受理’ か ‘棄却’ を返す。$(a, x) \in \mathcal{R}$ ならば，‘受理’ が返される。
- $N - 1$ 組以下のパーティのビューが集まっても x の情報は全く漏れない。

命題が以下のような MQ 問題のインスタンスである場合を考える。

$$\begin{cases} \mathbf{x}\,\mathrm{A}_1\,\mathbf{x}^\top + \mathbf{x}\,\mathbf{b}_1^\top = y_1 \\ \mathbf{x}\,\mathrm{A}_2\,\mathbf{x}^\top + \mathbf{x}\,\mathbf{b}_2^\top = y_2 \\ \qquad\qquad \vdots \\ \mathbf{x}\,\mathrm{A}_m\,\mathbf{x}^\top + \mathbf{x}\,\mathbf{b}_m^\top = y_m \end{cases} \tag{5.8}$$

ここで，$\mathrm{A}_1, \ldots, \mathrm{A}_m \in \mathbb{F}_q^{n\times n}$，$\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{F}_q^n$ である。この場合の（MQOM [21] で利用されている）秘匿マルチパーティ計算 f_{MQ} の入力は，この MQ 問題の解 $\mathbf{x}^*$ の分散 $[\![\mathbf{x}^*]\!]$ である。η を正の整数とする。正の整数 n_1, n_2 を $n_1 n_2 \geq n$ なるように選んでおく。また，$u_1, \ldots, u_{n_2} \in \mathbb{F}_q$ を相異なる元として固定しておく。このとき，f_{MQ} の計算手順は以下の通りである。

秘匿マルチパーティ計算 f_{MQ}

1. ランダムオラクル $\mathcal{O}_R$ により $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{q^\eta}$ が作られ，全パーティに送信する。
2. 各パーティ $\mathcal{P}_j$ は $[\![z]\!]_j = \sum_{i=1}^m \gamma_i \left([\![y_i]\!]_j - [\![\mathbf{x}^*]\!]_j \, \mathbf{b}_i^\top\right)$ を計算する。ここで，$[\![y_i]\!] = (y_i, 0, 0, \ldots, 0)$ である。
3. 各パーティ $\mathcal{P}_j$ は $[\![\mathbf{w}]\!]_j = [\![\mathbf{x}^*]\!]_j \left(\sum_{i=1}^m \gamma_i \, \mathrm{A}_i^\top\right)$ を計算する。
4. ヒントオラクル $\mathcal{O}_H$ により，$a_1, \ldots, a_{n_2} \in \mathbb{F}_{q^\eta}$，$Q'(u) \in \mathbb{F}_{q^\eta}[u]$ の分散 $[\![a_1]\!], \ldots, [\![a_{n_2}]\!]$，$[\![Q'(u)]\!]$ が作られ，対応するパーティに配布される。ここで，$a_1, \ldots, a_{n_2}$ はランダムに選ばれ，$Q'(u)$ は次のように定められる。まず，$n_1 - 1$ 次以下の多項式 $X_\ell(u) \in \mathbb{F}_q[u]$，$W_\ell(u) \in \mathbb{F}_{q^\eta}[u]$ $(\ell = 1, \ldots, n_2)$ を
$$\begin{cases} X_\ell(u_1) = x^*_{(\ell-1)n_1+1} \\ \quad\vdots \\ X_\ell(u_{n_1}) = x^*_{(\ell-1)n_1+n_1} \end{cases} \qquad \begin{cases} W_\ell(u_1) = w_{(\ell-1)n_1+1} \\ \quad\vdots \\ W_\ell(u_{n_1}) = w_{(\ell-1)n_1+n_1} \end{cases}$$
を満たす補間多項式によって計算する。次に，$\tilde{W}_\ell(u) \in \mathbb{F}_{q^\eta}[u]$ $(\ell = 1, \ldots, n_2)$ を
$$\tilde{W}_\ell(u) = W_\ell(u) + a_j \, (u - u_1)(u - u_2) \cdots (u - u_{n_1})$$
とし，$Q(u) \in \mathbb{F}_{q^\eta}[u]$ を $Q(u) = \sum_{\ell=1}^{n_2} X_\ell(u) \, \tilde{W}_\ell(u)$ で定める。最後に，q_0 を $Q(u)$ の定数項とし，$Q(u) = u \cdot Q'(u) + q_0$ で $Q'(u)$ を定める。
5. 各パーティ $\mathcal{P}_j$ は $n_1 - 1$ 次以下の多項式 $[\![X_\ell]\!]_j(u) \in \mathbb{F}_q[u]$，$[\![W_\ell]\!]_j(u) \in \mathbb{F}_{q^\eta}[u]$ $(\ell = 1, \ldots, n_2)$ を
$$\begin{cases} [\![X_\ell]\!]_j(u_1) = [\![x^*_{(\ell-1)n_1+1}]\!]_j \\ \quad\vdots \\ [\![X_\ell]\!]_j(u_{n_1}) = [\![x^*_{(\ell-1)n_1+n_1}]\!]_j \end{cases} \qquad \begin{cases} [\![W_\ell]\!]_j(u_1) = [\![w_{(\ell-1)n_1+1}]\!]_j \\ \quad\vdots \\ [\![W_\ell]\!]_j(u_{n_1}) = [\![w_{(\ell-1)n_1+n_1}]\!]_j \end{cases}$$
を満たす補間多項式によって計算する。（$[\![X_\ell]\!](u)$, $[\![W_\ell]\!](u)$ は，それぞれ $X_\ell(u)$, $W_\ell(u)$ の分散となる。）
6. 各パーティ $\mathcal{P}_j$ は $[\![\tilde{W}_\ell]\!]_j(u) \in \mathbb{F}_{q^\eta}[u]$ $(\ell = 1, \ldots, n_2)$ を
$$[\![\tilde{W}_\ell]\!]_j(u) = [\![W_\ell]\!]_j(u) + [\![a_\ell]\!]_j \, (u - u_1)(u - u_2) \cdots (u - u_{n_1})$$
で定める。（$[\![\tilde{W}_\ell]\!](u)$ は，$\tilde{W}_\ell(u)$ の分散となる。）
7. 各パーティ $\mathcal{P}_j$ は $[\![q_0]\!]_j = n_1^{-1} \cdot \left([\![z]\!]_j - \sum_{i=1}^{n_1} u_i \, [\![Q']\!]_j(u_i)\right)$ を計算する。
8. ランダムオラクル $\mathcal{O}_R$ により $r \in \mathbb{F}_{q^\eta} \backslash \{u_1, \ldots, u_{n_1}\}$ を取り，全パーティに送信する。
9. 各パーティ $\mathcal{P}_j$ は $[\![c_\ell]\!]_j = [\![\tilde{W}_\ell]\!]_j(r)$ $(\ell = 1, \ldots, n_2)$ を計算する。
10. 全パーティは $[\![c_\ell]\!]$ $(\ell = 1, \ldots, n_2)$ を共有し，$c_\ell \in \mathbb{F}_{q^\eta}$ $(\ell = 1, \ldots, n_2)$ を計算する。
11. 各パーティ $\mathcal{P}_j$ は $[\![v]\!]_j = r \cdot [\![Q']\!]_j(r) + [\![q_0]\!]_j - \sum_{\ell=1}^{n_2} c_\ell \, [\![X_\ell]\!]_j(r)$ を計算する。
12. 全パーティは $[\![v]\!]$ を共有し，v を計算する。
13. $v = 0$ なら ‘受理’，それ以外は ‘棄却’ を出力する。

この計算について補足する。この計算では，$\mathbf{x}^*$ が (5.8) の解であることを確認する代わりに，
$$\sum_{i=1}^m \gamma_i \left(y_i - \mathbf{x}^* \, \mathrm{A}_i \, {\mathbf{x}^*}^\top - \mathbf{x}^* \, \mathbf{b}_i^\top\right) = 0$$

であることを確認している。$\mathbf{x}^*$ が (5.8) の解でなくても，この等式は $1/q^\eta$ の確率で成り立つ。等式を書き直すと，

$$\sum_{i=1}^{m}\gamma_i\,(y_i-\mathbf{x}^*\,\mathbf{b}_i^\top)=\sum_{i=1}^{m}\gamma_i\,(\mathbf{x}^*\,\mathrm{A}_i\,\mathbf{x}^{*\top})=\mathbf{x}^*\left(\sum_{i=1}^{m}\gamma_i\,\mathrm{A}_i\right)\mathbf{x}^{*\top}=\langle\mathbf{x}^*,\,\mathbf{w}\rangle,\qquad\left(\mathbf{w}=\mathbf{x}^*\left(\sum_{i=1}^{m}\gamma_i\,\mathrm{A}_i^\top\right)\right)$$

となる。よって，$z=\sum_{i=1}^{m}\gamma_i\,(y_i-\mathbf{x}^*\,\mathbf{b}_i^\top)$ とおくならば，$z=\langle\mathbf{x}^*,\,\mathbf{w}\rangle$ を確かめればよい。これは，

$$z=\sum_{i=1}^{n_1}\sum_{\ell=1}^{n_2}X_\ell(u_i)\tilde{W}_\ell(u_i)=\sum_{i=1}^{n_1}Q(u_i)\tag{5.9}$$

と同値である。$v=0$ であれば，Schwartz-Zippel の補題により，高い確率で (5.9) が満たされることになる。η を大きくすることで，この確率を 1 に近づけることができる。

5.2.5.2 ゼロ知識証明への変換

MPC-in-the-Head では秘匿マルチパーティ計算からゼロ知識証明を構成する。このゼロ知識証明は，命題 a に対して証拠 x を知っているかどうかを検証するものである。秘匿マルチパーティ計算 f_{MQ} に対応するゼロ知識証明の基本設計は以下の通りである。

f_{MQ} に対応するゼロ知識証明

1. 証明者は，証拠 $\mathbf{x}^*$ の分散 $[\![\mathbf{x}^*]\!]$ を作成する。そして，すべての $j\in\{1,\ldots,N\}$ に対し，$[\![\mathbf{x}^*]\!]_j$ のコミットメントを作成し，検証者に送る。
2. 検証者は，ランダムに $\gamma_1,\ldots,\gamma_m\in\mathbb{F}_{q^\eta}$ を生成し，これらをチャレンジとして証明者に送る。
3. 証明者は，$a_1,\ldots,a_{n_2}$, $Q'(u)$ を生成し，これらの分散 $[\![a_1]\!],\ldots,[\![a_{n_2}]\!]$, $[\![Q']\!](u)$ を作成する。そして，すべての $j\in\{1,\ldots,N\}$ に対し，$[\![a_1]\!]_j,\ldots,[\![a_{n_2}]\!]_j$, $[\![Q']\!]_j(u)$ のコミットメントを作成し，検証者に送る。
4. 検証者は，ランダムに $r\in\mathbb{F}_{q^\eta}\backslash\{u_1,\ldots,u_{n_1}\}$ を生成し，これらをチャレンジとして証明者に送る。
5. 証明者は，（全てのパーティの計算を “頭の中で” 行い，）$[\![c_1]\!],\ldots,[\![c_{n_2}]\!]$，および，$[\![v]\!]$ を作成する。そして，これらを検証者に送る。
6. 検証者は，ランダムに $i^*\in\{1,2,\ldots,N\}$ を生成し，これをチャレンジとして証明者に送る。
7. 証明者は，すべての $j\in\{1,2,\ldots,N\}\backslash\{i^*\}$ に対し，$[\![\mathbf{x}]\!]_j$, $[\![a_1]\!]_j,\ldots,[\![a_{n_2}]\!]_j$, $[\![Q']\!]_j(u)$ を検証者に開示する。
8. 検証者は，以下を検証し，全て正しければ ‘受理’ を，それ以外は ‘棄却’ を出力する。
   - ✓ すべての $j\in\{1,2,\ldots,N\}\backslash\{i^*\}$ に対し，$[\![\mathbf{x}^*]\!]_j$ のコミットメント，および，$[\![a_1]\!]_j,\ldots,[\![a_{n_2}]\!]_j$, $[\![Q']\!]_j(u)$ のコミットメントが正しいこと
   - ✓ すべての $j\in\{1,2,\ldots,N\}\backslash\{i^*\}$ に対し，$[\![\mathbf{x}^*]\!]_j$, $[\![a_1]\!]_j,\ldots,[\![a_{n_2}]\!]_j$, $[\![Q']\!]_j(u)$ から $\mathcal{P}_j$ と同じ計算を行ったとき，$[\![c_1]\!]_j,\ldots,[\![c_{n_2}]\!]_j$, $[\![v]\!]_j$ の計算結果が一致すること
   - ✓ $v=0$ であること

このゼロ知識証明について補足する。f_{MQ} における全てのパーティのビューは（もともと開示されるものを除き）コミットメントが作成され，検証者に送られている。また，f_{MQ} においてランダムオラクルが介入する部分は，検証者のチャレンジに置き換えられている。そして，1 つのパーティ以外のすべてのパーティに対するビューが開示され，そのビューから各パーティと同じ計算を行って得られる結果と開示情報が一致すること，および，$v=0$ であることにより検証者は ‘受理’ を行っている。このゼロ知識証明の健全性誤差（soundness error）ε は約 $1/N$ である。このゼロ知識証明を τ 回繰り返すことにより，全体の健全性誤差を ε^τ にすることができる。

このゼロ知識証明に Fiat-Shamir 変換を施すことで署名方式 MQOM [21] が構成できる。MQOM ついては，5.3.4 節で詳しく述べる。また，5.3.5 節で詳しく述べる MiRitH は MinRank 問題に付随する秘匿マルチパーティ計算から構成される署名方式である。

5.3 多変数多項式に基づく主要な暗号方式

多変数公開鍵暗号で標準化が有力視されるのは効率的な検証と短い署名長を持つ署名方式の UOV である。但し，UOV は公開鍵長が大きくなりやすいという性質を持つため，公開鍵長の削減手法を取り入れている UOV の変種である QR-UOV と MAYO も標準化の有力候補である。

また，MPC-in-the-Head では，MQ 問題に関するマルチパーティ計算に基づく署名方式 MQOM と，MinRank 問題に関するマルチパーティ計算に基づく署名方式 MiRitH が注目されている。

表 5.2: 多変数多項式に基づく暗号の分類

文献	暗号化	鍵交換	署名
UOV [28, 10]			○
QR-UOV [22, 23]			○
MAYO [8, 9]			○
MQOM [21]			○
MiRitH [3]			○

5.3.1 署名方式 UOV

5.3.1.1 UOV の概要

5.2.4.1 節で UOV の基本アルゴリズムは述べたため，この節でのアルゴリズムの記述は割愛する。NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出された UOV [10] のアルゴリズムには，さらに，5.2.4.2 節で述べた公開鍵長の削減手法が取り入れられている。

5.3.1.2 UOV のパラメータ選択

UOV の設計に必要なパラメータは，q, m, n である。NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出されたドキュメント [10] では，以下のように UOV のパラメータ見積もりが公開されている。

(q, m, n)	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ
$(256, 44, 112)$	レベル 1	$43,576$ Bytes	48 Bytes	128 Bytes
$(16, 64, 160)$	レベル 1	$66,576$ Bytes	48 Bytes	96 Bytes
$(256, 72, 184)$	レベル 3	$189,232$ Bytes	48 Bytes	200 Bytes
$(256, 96, 244)$	レベル 5	$446,992$ Bytes	48 Bytes	260 Bytes

5.3.2 署名方式 QR-UOV

5.3.2.1 QR-UOV の概要

QR-UOV [22, 23] は UOV の変種である。$\mathbb{F}_{q^\ell}$ 上の行列の集合 $\mathbb{F}_{q^\ell}^{n' \times n'}$ は，$\mathbb{F}_q$ 上の行列の集合 $\mathbb{F}_q^{n'\ell \times n'\ell}$ の部分集合と見なすことができる。この部分集合に属する行列は，$\mathbb{F}_q^{n'\ell \times n'\ell}$ の元として表示するよりも，$\mathbb{F}_{q^\ell}^{n' \times n'}$ の元として表示する方が，サイズを $1/\ell$ 倍に圧縮できる。QR-UOV は，この性質を利用して UOV の公開鍵長を削減している。

ℓ, V, M を正の整数とし，$v = \ell \cdot V$, $m = \ell \cdot M$, $n = v + m$ とする。次数 ℓ の既約多項式 $f \in \mathbb{F}_q[t]$ を取り，$\mathbb{F}_q$ の拡大体 $E_f = \mathbb{F}_q[t]/(f)$ に，$\mathbb{F}_q$-基底を $1, t, t^2, \ldots, t^{\ell-1}$ で入れ，$\mathbb{F}_q$ 上のベクトル空間として $\mathbb{F}_q^\ell$ と同一視する。任意の

$g \in \mathbb{F}_q[t]$ に対し，写像 $E_f \ni x \mapsto xg \in E_f$ は $\mathbb{F}_q$ 上の線形写像となる。よって，この写像は $\mathbb{F}_q$ 上の $\ell \times \ell$ 行列として表すことができる。この行列を $\Phi_g^f \in \mathbb{F}_q^{\ell\times\ell}$ と表し，$\mathcal{A}_f = \{\Phi_g^f \,|\, g \in E_f\} (\subset \mathbb{F}_q^{\ell\times\ell})$ とおく。$\phi : E_f \to \mathbb{F}_q$ を非自明な $\mathbb{F}_q$-線形写像で固定し，$\mathrm{W} = (\phi(t^{i+j-2}))_{ij} \in \mathbb{F}_q^{\ell\times\ell}$ とすると，任意の $\mathrm{X} \in \mathcal{A}_f$ に対して，$\mathrm{WX} \in \mathbb{F}_q^{\ell\times\ell}$ は対称行列になることが知られている（[22, Theorem 1]）。正の整数 a, b に対し，$\mathcal{A}_f^{a,b}$ を以下のような形で表される $a\ell \times b\ell$ 行列の集合とする：

$$\begin{pmatrix} \mathrm{X}_{11} & \mathrm{X}_{12} & \cdots & \mathrm{X}_{1b} \\ \mathrm{X}_{21} & \mathrm{X}_{22} & \cdots & \mathrm{X}_{2b} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{X}_{a1} & \mathrm{X}_{a2} & \cdots & \mathrm{X}_{ab} \end{pmatrix} \quad (X_{11}, X_{12}, \ldots, \mathrm{X}_{ab} \in \mathcal{A}_f).$$

$\mathrm{W}^{(a)} \in \mathbb{F}_q^{a\ell\times a\ell}$ を W が主対角線に a 個並ぶ対角行列とし，$\mathrm{W}^{(a)}\mathcal{A}_f^{a,b} = \{\mathrm{W}^{(a)}\,\mathrm{X} \,|\, \mathrm{X} \in \mathcal{A}_f^{a,b}\}$ $(\subset \mathbb{F}_q^{a\ell\times b\ell})$ とする。$\mathbb{F}_q^{a\ell\times b\ell}$ に属する一般の行列を記述するには，$\mathbb{F}_q$ の元が $a\,b\,\ell^2$ 個必要であるが，それが $\mathrm{W}^{(a)}\mathcal{A}_f^{a,b}$ に属する場合は，$a\,b\,\ell$ 個で記述できることに注意してほしい。

QR-UOV では，公開鍵 $F(\mathbf{x})$ の成分として 2 次斉次多項式を用いる。QR-UOV は双極型システムであるため，5.2.4.2 節で述べたように，$F(\mathbf{x})$ は，m 個の行列 $(\in \mathbb{F}_q^{n\times n})$ を用いて記述することができる。QR-UOV では，これらの行列がすべて $\mathrm{W}^{(V+M)}\mathcal{A}_f^{V+M,V+M}$ に属するような $F(\mathbf{x})$ だけを用いる。但し，これらの行列は上三角行列の形にはできないので，対称行列で記述する。この影響で，有限体の位数 q は奇数にする必要がある。$\mathrm{W}^{(V+M)}\mathcal{A}_f^{V+M,V+M}$ に属する行列は，一般の $\mathbb{F}_q^{n\times n}$ に属する行列よりも小さいサイズで記述できるため，オリジナルの UOV よりも QR-UOV の公開鍵の方が小さいサイズで記述できる。また，5.2.4.2 節で述べた UOV に対して適用できる公開鍵長削減手法は，QR-UOV に対しても適用可能である。

安全性パラメータを λ とし，以下の関数を用意する。

- $\mathrm{Expand}_{\mathrm{sk}}$： 任意の 2λ-ビット列から 1 個の $\mathcal{A}_f^{V,M}$ に属する行列を生成する疑似乱数生成関数
- $\mathrm{Expand}_{\mathrm{pk}}$： 任意の 2λ-ビット列から m 個の $\mathrm{W}^{(V)}\mathcal{A}_f^{V,V}$ に属する対称行列と，m 個の $\mathrm{W}^{(V)}\mathcal{A}_f^{V,M}$ に属する行列を生成する疑似乱数生成関数
- $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$： 暗号学的ハッシュ関数

鍵生成

1. $\mathrm{seed}_{\mathrm{pk}}, \mathrm{seed}_{\mathrm{sk}} \in \{0,1\}^{2\lambda}$ をランダムに選ぶ。
2. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathrm{W}^{(V)}\mathcal{A}_f^{V,V}$ (対称行列)，$\mathrm{P}_{i,2} \in \mathrm{W}^{(V)}\mathcal{A}_f^{V,M}$ $(i = 1, \ldots, m)$ を得る.
3. $\mathrm{Expand}_{\mathrm{sk}}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\mathrm{S}_0 \in \mathcal{A}_f^{V,M}$ を得る。
4. $\mathrm{P}_{i,3} = -\mathrm{S}_0^\top \mathrm{P}_{i,1}\mathrm{S}_0 + \mathrm{P}_{i,2}^\top \mathrm{S}_0 + \mathrm{S}_0^\top \mathrm{P}_{i,2} \in \mathbb{F}_q^{m\times m}$ $(i = 1, \ldots, m)$ を計算する。

公開鍵は $\mathrm{pk} = (\mathrm{seed}_{\mathrm{pk}}, \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$，秘密鍵は $\mathrm{sk} = \mathrm{seed}_{\mathrm{sk}}$ である。次に，署名生成である。メッセージを $M \in \{0,1\}^*$ とする。

署名生成

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$ を取り出す.
2. sk から $\mathrm{seed}_{\mathrm{sk}}$ を取り出す.
3. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathrm{W}^{(V)}\mathcal{A}_f^{V,V}$ (対称行列)，$\mathrm{P}_{i,2} \in W^{(V)}\mathcal{A}_f^{V,M}$ $(i = 1, \ldots, m)$ を得る.
4. $\mathrm{Expand}_{\mathrm{sk}}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\mathrm{S}_0 \in \mathcal{A}_f^{V,M}$ を得る。
5. $\mathrm{G}_i = -\mathrm{P}_{i,1}\mathrm{S}_0 + \mathrm{P}_{i,2} \in \mathbb{F}_q^{v\times m}$ $(i = 1, \ldots, m)$ を計算する。
6. $\mathrm{U} = \begin{pmatrix} \mathbf{I}_v & \mathbf{0}_{v\times m} \\ -\mathrm{S}_0^\top & \mathbf{I}_m \end{pmatrix} \in \mathbb{F}_q^{n\times n}$ とおく。

7. $\mathbf{y} = (y_1, \ldots, y_v) \in \mathbb{F}_q^v$ をランダムに選ぶ。
8. $\mathrm{L} = (2\,(\mathbf{y}\,\mathrm{G}_1)^\top, \ldots, 2\,(\mathbf{y}\,\mathrm{G}_m)^\top) \in \mathbb{F}_q^{m \times m}$ を計算する。（縦ベクトルを m 列並べて行列を作る。）
9. $\mathbf{u} = (\mathbf{y}\,\mathrm{P}_{1,1}\,\mathbf{y}^\top, \ldots, \mathbf{y}\,\mathrm{P}_{m,1}\,\mathbf{y}^\top) \in \mathbb{F}_q^m$ を計算する。
10. $\mathrm{salt} \in \{0,1\}^\lambda$ をランダムに選び，$\mathbf{t} = \mathcal{H}(M \,\|\, \mathrm{salt})$ を計算する。
11. 連立線形方程式 $\mathbf{x}\,\mathrm{L} = \mathbf{t} - \mathbf{u}$ の解を計算し，解 $\mathbf{x} = (y_{v+1}, \ldots, y_n) \in \mathbb{F}_q^m$ を得る。もし解がなければ，Step 10 に戻る。
12. $\mathbf{s} = (y_1, \ldots, y_n)\,\mathrm{U}$ を計算する.

$\sigma = (\mathrm{salt}, \mathbf{s})$ が署名となる。最後に検証である。

検証

1. pk から $(\mathrm{seed}_{\mathrm{pk}},\ \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$ を取り出す.
2. σ から $(\mathrm{salt}, \mathbf{s})$ を取り出す.
3. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathrm{W}^{(V)}\mathcal{A}_f^{V,V}$ (対称行列)，$\mathrm{P}_{i,2} \in \mathrm{W}^{(V)}\mathcal{A}_f^{V,M}$ $(i = 1, \ldots, m)$ を得る.
4. $\mathrm{F}_i = \begin{pmatrix} \mathrm{P}_{i,1} & \mathrm{P}_{i,2} \\ \mathrm{P}_{i,2}^\top & \mathrm{P}_{i,3} \end{pmatrix}$ $(i = 1, \ldots, m)$ とおく。
5. $\mathbf{t} = \mathcal{H}(M \,\|\, \mathrm{salt})$ を計算する.
6. $\mathbf{t}' = (\mathbf{s}\,\mathrm{F}_1\,\mathbf{s}^\top, \ldots, \mathbf{s}\,\mathrm{F}_m\,\mathbf{s}^\top)$ を計算する.
7. $\mathbf{t} = \mathbf{t}'$ ならば ‘受理’ を，それ以外は ‘棄却’ を返す。

5.3.2.2 QR-UOV のパラメータ選択

QR-UOV の設計に必要なパラメータは，λ, q, v, m, ℓ である。NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出されたドキュメント [23] では，以下のように QR-UOV のパラメータ見積もりが公開されている。

(λ, q, v, m, ℓ)	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ
$(128, 7, 740, 100, 10)$	レベル 1	20,657 Bytes	32 Bytes	331 Bytes
$(128, 31, 165, 60, 3)$	レベル 1	23,657 Bytes	32 Bytes	157 Bytes
$(128, 31, 600, 70, 10)$	レベル 1	12,282 Bytes	32 Bytes	435 Bytes
$(128, 127, 156, 54, 3)$	レベル 1	24,271 Bytes	32 Bytes	200 Bytes
$(192, 7, 1100, 140, 10)$	レベル 3	55,173 Bytes	48 Bytes	489 Bytes
$(192, 31, 246, 87, 3)$	レベル 3	71,007 Bytes	48 Bytes	232 Bytes
$(192, 31, 890, 100, 10)$	レベル 3	34,423 Bytes	48 Bytes	643 Bytes
$(192, 127, 228, 78, 3)$	レベル 3	71,915 Bytes	48 Bytes	292 Bytes
$(256, 7, 1490, 190, 10)$	レベル 5	135,439 Bytes	64 Bytes	662 Bytes
$(256, 31, 324, 114, 3)$	レベル 5	158,453 Bytes	64 Bytes	306 Bytes
$(256, 31, 1120, 120, 10)$	レベル 5	58,564 Bytes	64 Bytes	807 Bytes
$(256, 127, 306, 105, 3)$	レベル 5	173,708 Bytes	64 Bytes	392 Bytes

5.3.3 署名方式 MAYO

5.3.3.1 MAYO の概要

MAYO [8, 9] は UOV の変種である。公開鍵を作る基となる変数の個数が少ない多変数多項式系 $P(\mathbf{x})$ を用意しておき，検証者は，検証時（あるいはそれ以前）に $P(\mathbf{x})$ から MAYO の公開鍵 $F(\mathbf{x})$ を構成する。そのため，MAYO

の公開鍵は，$F(\mathbf{x})$ の係数集合ではなく，$P(\mathbf{x})$ の係数集合となる。これにより，MAYO は，オリジナルの UOV に比べて公開鍵長を小さくすることができる。

m, v, o, k を正の整数とし，$o < m$，$n = v + o$ とする。2 次斉次多変数多項式写像 $P(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})) : \mathbb{F}_q^n \to \mathbb{F}_q^m$ に対し，ある o 次元部分空間 $O\,(\subset \mathbb{F}_q^n)$ があり，

$$P(\mathbf{o}) = \mathbf{0}_m \quad (\mathbf{o} \in O)$$

を満たすとする。5.2.4.1 節の言葉を使えば，O はオイル空間である。もし $o \geq m$ であれば，$P(\mathbf{x})$ は UOV の公開鍵として使用できるが，$o < m$ なのでそれはできない。$P^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k) : \mathbb{F}_q^{kn} \to \mathbb{F}_q^m$ を次のようにおく。

$$P^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \sum_{i=1}^{k} P(\mathbf{x}_i)\,\mathrm{E}_{i,i} + \sum_{1 \leq i < j \leq k} P'(\mathbf{x}_i, \mathbf{x}_j)\,\mathrm{E}_{i,j} \tag{5.10}$$

ここで，$\mathrm{E}_{i,j} \in \mathbb{F}_q^{m \times m}$ $(1 \leq i \leq j \leq k)$ は正則行列であり，$P'(\mathbf{x}, \mathbf{y})$ は $P(\mathbf{x} + \mathbf{y}) - P(\mathbf{x}) - P(\mathbf{y})$ で定まる双線形写像である。すると，

$$P^\star(\mathbf{o}_1, \ldots, \mathbf{o}_k) = \mathbf{0}_m \quad (\mathbf{o}_1, \ldots, \mathbf{o}_k \in O)$$

を満たすので，O^k が $P^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ のオイル空間となる。$ko = \dim_{\mathbb{F}_q} O^k$ なので，$ko \geq m$ を満たせば，$P^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ は UOV の公開鍵として使用できる。$\mathrm{E}_{i,j}$ $(1 \leq i \leq j \leq n)$ をシステムパラメータとしておけば，$P^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ は (5.10) により，$P(\mathbf{x})$ だけから構成できる。公開鍵を $P(\mathbf{x})$ の係数集合だけで記述することで，公開鍵のサイズを小さくした UOV が MAYO である。さらに，5.2.4.2 節で述べた UOV に対して適用できる公開鍵長削減手法は，MAYO に対しても適用可能である。

安全性パラメータを λ とし，以下の行列，関数を用意する。

- 正則行列 $\mathrm{E}_{i,j} \in \mathbb{F}_q^{m \times m}$ $(1 \leq i \leq j \leq k)$
- $\mathrm{Expand}_{\mathrm{sk}}$： 任意の λ-ビット列から 1 個の $\mathbb{F}_q^{o \times v}$ に属する行列を生成する疑似乱数生成関数
- $\mathrm{Expand}_{\mathrm{pk}}$： 任意の λ-ビット列から m 個の $\mathbb{F}_q^{v \times v}$ に属する上三角行列と，m 個の $\mathbb{F}_q^{v \times o}$ に属する行列を生成する疑似乱数生成関数
- $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$： 暗号学的ハッシュ関数

鍵生成

1. $\mathrm{seed}_{\mathrm{pk}}$, $\mathrm{seed}_{\mathrm{sk}} \in \{0,1\}^\lambda$ をランダムに選ぶ。
2. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathbb{F}_q^{v \times v}$（上三角行列），$\mathrm{P}_{i,2} \in \mathbb{F}_q^{v \times o}$ $(i = 1, \ldots, m)$ を得る.
3. $\mathrm{Expand}_{\mathrm{sk}}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\mathrm{R} \in \mathbb{F}_q^{o \times v}$ を得る。
4. $\mathrm{P}_{i,3} = \mathrm{upper}(-\mathrm{R}\,\mathrm{P}_{i,1}\mathrm{R}^\top - \mathrm{R}\,\mathrm{P}_{i,2}) \in \mathbb{F}_q^{o \times o}$ $(i = 1, \ldots, m)$ を計算する。

公開鍵は $\mathrm{pk} = (\mathrm{seed}_{\mathrm{pk}}, \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$，秘密鍵は $\mathrm{sk} = \mathrm{seed}_{\mathrm{sk}}$ である。次に，署名生成である。メッセージを $M \in \{0,1\}^*$ とする。

署名生成

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$ を取り出す.
2. sk から $\mathrm{seed}_{\mathrm{sk}}$ を取り出す.
3. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathbb{F}_q^{v \times v}$（上三角行列），$\mathrm{P}_{i,2} \in \mathbb{F}_q^{v \times o}$ $(i = 1, \ldots, m)$ を得る.
4. $\mathrm{Expand}_{\mathrm{sk}}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\mathrm{R} \in \mathbb{F}_q^{o \times v}$ を得る。
5. $\mathrm{F}_i = (\mathrm{P}_{i,1} + \mathrm{P}_{i,1}^\top)\,\mathrm{R}^\top + \mathrm{P}_{i,2} \in \mathbb{F}_q^{v \times o}$ $(i = 1, \ldots, m)$ を計算する。

6. $\mathrm{U} = \begin{pmatrix} \mathbf{I}_v & \mathbf{0}_{v\times o} \\ \mathrm{R} & \mathbf{I}_o \end{pmatrix} \in \mathbb{F}_q^{n\times n}$ とおく。
7. $\mathbf{y}_1, \ldots, \mathbf{y}_k \in \mathbb{F}_q^v$ （横ベクトル）をランダムに選ぶ。
8. $\mathrm{L}_i = \sum_{j=1}^{i}((\mathbf{y}_j\,\mathrm{F}_1)^\top, \ldots, (\mathbf{y}_j\,\mathrm{F}_m)^\top)\,\mathrm{E}_{j,i} + \sum_{j=i+1}^{k}((\mathbf{y}_j\,\mathrm{F}_1)^\top, \ldots, (\mathbf{y}_j\,\mathrm{F}_m)^\top)\,\mathrm{E}_{i,j} \in \mathbb{F}_q^{o\times m}$ $(i = 1, \ldots, k)$ を計算する.
9. $\mathrm{L} = \begin{pmatrix} \mathrm{L}_1 \\ \vdots \\ \mathrm{L}_k \end{pmatrix} \in \mathbb{F}_q^{ko\times m}$ とおく。
10. $\mathbf{u} = \sum_{i=1}^{k}(\mathbf{y}_i\,\mathrm{P}_{1,1}\,\mathbf{y}_i^\top, \ldots, \mathbf{y}_i\,\mathrm{P}_{m,1}\,\mathbf{y}_i^\top)\,\mathrm{E}_{i,i} + \sum_{1\le i<j\le k}(\mathbf{y}_i\,(\mathrm{P}_{1,1}+\mathrm{P}_{1,1}^\top)\,\mathbf{y}_j^\top, \ldots, \mathbf{y}_i\,(\mathrm{P}_{m,1}+\mathrm{P}_{m,1}^\top)\,\mathbf{y}_j^\top)\,\mathrm{E}_{i,j} \in \mathbb{F}_q^m$ を計算する。
11. $\mathrm{salt} \in \{0,1\}^\lambda$ をランダムに選び，$\mathbf{t} = \mathcal{H}(M \,\|\, \mathrm{salt})$ を計算する。
12. 連立線形方程式 $(\mathbf{x}_1, \ldots, \mathbf{x}_k)\,\mathrm{L} = \mathbf{t} - \mathbf{u}$ の解を計算し，解 $(\mathbf{x}_1, \ldots, \mathbf{x}_k) = (\mathbf{z}_1, \ldots, \mathbf{z}_k) \in \mathbb{F}_q^{ko}$ を得る。もし解がなければ，Step 11 に戻る。
13. $\mathbf{s} = ((\mathbf{y}_1, \mathbf{z}_1)\,\mathrm{U}, \ldots, (\mathbf{y}_k, \mathbf{z}_k)\,\mathrm{U})$ を計算する.

$\sigma = (\mathrm{salt}, \mathbf{s})$ が署名となる。最後に検証である。

検証

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \{\mathrm{P}_{i,3}\}_{i=1,\ldots,m})$ を取り出す.
2. σ から $(\mathrm{salt}, (\mathbf{s}_1, \ldots, \mathbf{s}_k))$ を取り出す.
3. $\mathrm{Expand}_{\mathrm{pk}}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{P}_{i,1} \in \mathbb{F}_q^{v\times v}$ （上三角行列），$\mathrm{P}_{i,2} \in \mathbb{F}_q^{v\times o}$ $(i = 1, \ldots, m)$ を得る.
4. $\mathrm{P}_i = \begin{pmatrix} \mathrm{P}_{i,1} & \mathrm{P}_{i,2} \\ \mathbf{0}_{v\times o} & \mathrm{P}_{i,3} \end{pmatrix}$ $(i = 1, \ldots, m)$ とおく。
5. $\mathbf{t} = \mathcal{H}(M \,\|\, \mathrm{salt})$ を計算する.
6. $\mathbf{t}' = \sum_{i=1}^{k}(\mathbf{s}_i\mathrm{P}_1\,\mathbf{s}_i^\top, \ldots, \mathbf{s}_i\mathrm{P}_k\,\mathbf{s}_i^\top)\,\mathrm{E}_{i,i} + \sum_{1\le i<j\le k}(\mathbf{s}_i(\mathrm{P}_1+\mathrm{P}_1^\top)\,\mathbf{s}_j^\top, \ldots, \mathbf{s}_i(\mathrm{P}_k+\mathrm{P}_k^\top)\,\mathbf{s}_j^\top)\,\mathrm{E}_{i,j}$ を計算する.
7. $\mathbf{t} = \mathbf{t}'$ ならば ‘受理’ を，それ以外は ‘棄却’ を返す。

5.3.3.2 MAYO のパラメータ選択

MAYO のパラメータは，NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出されたドキュメント [9] に記載されていたが，(MAYO に限定されない) under-defined な MQ 問題に対する解読手法 [26] が適用できることが分かり，パラメータの修正が必要となった。しかし，これは致命的な攻撃ではなく，MAYO は NIST PQC 標準化プロジェクト追加署名第 2 ラウンドへの進出が決まっている。MAYO の Round 2 ドキュメントで修正パラメータが公開される予定であるが，本稿の執筆時点（2025 年 2 月 19 日時点）では公開されていない。

5.3.4 署名方式 MQOM

5.3.4.1 MQOM の概要

MQOM [21] は，5.2.5.1 節で説明した MQ 問題に関する秘匿マルチパーティ計算 f_{MQ} から MPC-in-the-Head で構成された署名方式である。5.2.5.2 節で述べたように，f_{MQ} はゼロ知識証明に変換することができる。さらに，Fiat-Shamir 変換により署名方式が構成できる。以下では，5.2.5.1 節の設定や記号を用いる。安全性パラメータを λ とし，以下の関数を用意する。

- Expand： 任意の λ-ビット列を入力とする疑似乱数生成関数

- $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3 : \{0,1\}^* \to \{0,1\}^{2\lambda}$ ：暗号学的ハッシュ関数
- Commit：コミットメント関数

鍵生成

1. $\mathrm{seed}_{\mathrm{pk}},\ \mathrm{seed}_{\mathrm{sk}} \in \{0,1\}^\lambda$ をランダムに選ぶ。
2. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{A}_i \in \mathbb{F}_q^{n\times n}$（上三角行列），$\mathbf{b}_i \in \mathbb{F}_q^n$ $(i = 1, \ldots, m)$ を得る．
3. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\mathbf{x}^* \in \mathbb{F}_q^n$ を得る。
4. $y_i = \mathbf{x}^* \, \mathrm{A}_i \, \mathbf{x}^{*\top} + \mathbf{x}^* \, \mathbf{b}_i^\top \quad (i = 1, \ldots, m)$ を計算する。

公開鍵は $\mathrm{pk} = (\mathrm{seed}_{\mathrm{pk}}, \mathbf{y} = (y_1, \ldots, y_m))$，秘密鍵は $\mathrm{sk} = \mathrm{seed}_{\mathrm{sk}}$ である。次に，署名生成である。メッセージを $M \in \{0,1\}^*$ とする。

署名生成

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \mathbf{y})$ を取り出す。
2. sk から $\mathrm{seed}_{\mathrm{sk}}$ を取り出す。
3. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{A}_i \in \mathbb{F}_q^{n\times n}$（上三角行列），$\mathbf{b}_i \in \mathbb{F}_q^n$ $(i = 1, \ldots, m)$ を得る．
4. $\mathrm{salt} \in \{0,1\}^{2\lambda},\ \mathrm{mseed} \in \{0,1\}^\lambda$ をランダムに選ぶ。
5. $\mathrm{Expand}(\mathrm{salt}, \mathrm{mseed})$ の計算により，$\mathrm{rseed}^{[e]} \in \{0,1\}^\lambda$ $(e = 1, \ldots, \tau)$ を得る．
6. $e = 1, \ldots, \tau$ に対して以下を行う：
   - 6-1. $\mathrm{Expand}(\mathrm{salt}, \mathrm{rseed}^{[e]})$ の計算により，$\mathrm{seed}_j^{[e]} \in \{0,1\}^\lambda$ $(j = 1, \ldots, N)$ を得る．
   - 6-2. すべての $j = 1, \ldots, N$ に対し，$\mathrm{Expand}(\mathrm{salt}, \mathrm{seed}_j^{[e]})$ の計算により以下を得る：
     - $j < N$ ならば，$[\![\mathbf{x}^{*[e]}]\!]_j, [\![a_1^{[e]}]\!]_j, \ldots, [\![a_{n_2}^{[e]}]\!]_j, [\![Q'^{[e]}]\!]_j(u)$
     - $j = N$ ならば，$[\![a_1^{[e]}]\!]_N, \ldots, [\![a_{n_2}^{[e]}]\!]_N$
   - 6-3. $[\![\mathbf{x}^{*[e]}]\!]_N = \mathbf{x}^* - \sum_{j=1}^{N-1} [\![\mathbf{x}^{*[e]}]\!]_j$ を計算する。
   - 6-4. コミットメントを計算する：

$$\mathrm{com}_j^{[e]} = \begin{cases} \mathrm{Commit}(\mathrm{salt}, e, j, \mathrm{seed}_j^{[e]}) & j = 1, \ldots, N-1 \\ \mathrm{Commit}(\mathrm{salt}, e, N, \mathrm{seed}_N^{[e]}, [\![\mathbf{x}^{*[e]}]\!]_N) & j = N \end{cases}$$

7. $h_1 = \mathcal{H}_1(M, \mathrm{salt}, \mathrm{com}_1^{[1]}, \ldots, \mathrm{com}_N^{[\tau]})$ を計算する。
8. $\mathrm{Expand}(h_1)$ の計算により，$\gamma_1^{[e]}, \ldots, \gamma_m^{[e]}$ $(e = 1, \ldots, \tau)$ を得る。
9. $e = 1, \ldots, \tau$ に対して以下を行う：
   - 9-1. f_{MQ} のヒントオラクルと同じ計算により，$\mathbf{x}^*, \{\mathrm{A}_i, \mathbf{b}_i\}_{i=1,\ldots,m}, \gamma_1^{[e]}, \ldots, \gamma_m^{[e]}, a_1^{[e]}, \ldots, a_{n_2}^{[e]}$（但し，$a_i^{[e]} = \sum_{j=1}^N [\![a_i^{[e]}]\!]_j$）から，$Q'^{[e]}(u) \in \mathbb{F}_{q^\eta}[u]$ を計算する。
   - 9-2. $[\![Q'^{[e]}]\!]_N(u) = Q'^{[e]}(u) - \sum_{j=1}^{N-1} [\![Q'^{[e]}]\!]_j(u)$ を計算する。
   - 9-3. コミットメント $\mathrm{com}'^{[e]}_N = \mathrm{Commit}(\mathrm{salt}, e, 0, [\![Q'^{[e]}]\!]_N(u))$ を計算する。
10. $h_2 = \mathcal{H}_2(M, \mathrm{salt}, h_1, \mathrm{com}'^{[1]}_N, \ldots, \mathrm{com}'^{[\tau]}_N)$ を計算する。
11. $\mathrm{Expand}(h_2)$ の計算により，$r^{[1]}, \ldots, r^{[\tau]} \in \mathbb{F}_{q^\eta}$ を得る。
12. $e = 1, \ldots, \tau$ に対して以下を行う：
    - 12-1. f_{MQ} における全パーティと同じ計算により，$[\![\mathbf{x}^{*[e]}]\!], [\![a_1^{[e]}]\!], \ldots, [\![a_{n_2}^{[e]}]\!], [\![Q'^{[e]}]\!](u)$ から，$[\![\mathrm{broad}^{[e]}]\!] = ([\![c_1^{[e]}]\!], \ldots, [\![c_{n_2}^{[e]}]\!], [\![v^{[e]}]\!])$ を計算する。
    - 12-2. $\mathrm{broad}^{[e]} = (c_1^{[e]}, \ldots, c_{n_2}^{[e]}, v^{[e]}) = \sum_{j=1}^N [\![\mathrm{broad}^{[e]}]\!]$ を計算する。
13. $h_3 = \mathcal{H}_3(M, \mathrm{salt}, h_2, [\![\mathrm{broad}^{[1]}]\!], \ldots, [\![\mathrm{broad}^{[\tau]}]\!])$ を計算する。
14. $\mathrm{Expand}(h_3)$ の計算により，$i^{*[1]}, \ldots, i^{*[\tau]} \in \{1, \ldots, N\}$ を得る。

15. $\mathrm{view}^{[e]}$ $(e=1,\ldots,\tau)$ を以下のようにおく：

$$\mathrm{view}^{[e]} = \begin{cases} (\{\mathrm{seed}_j^{[e]}\}_{j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}},\ [\![\mathbf{x}^{*[e]}]\!]_N,\ [\![Q'^{[e]}]\!]_N(u)) & i^{*[e]}\neq N \\ (\{\mathrm{seed}_j^{[e]}\}_{j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}}) & i^{*[e]}=N \end{cases}$$

16. $\sigma = (\mathrm{salt},\ h_1,\ h_2,\ h_3,\ \{\mathrm{view}^{[e]},\ \mathrm{broad}^{[e]},\ \mathrm{com}_{i^{*[e]}}^{[e]},\ \mathrm{com}'^{[e]}_N\}_{e=1,\ldots,\tau})$ とおく。

σ が署名となる。最後に検証である。

検証

1. pk から $(\mathrm{seed}_{\mathrm{pk}},\ \mathbf{y})$ を取り出す。
2. σ から $(\mathrm{salt},\ h_1,\ h_2,\ h_3,\ \{\mathrm{view}^{[e]},\ \mathrm{broad}^{[e]},\ \mathrm{com}_{i^{*[e]}}^{[e]},\ \mathrm{com}'^{[e]}_N\}_{e=1,\ldots,\tau})$ を取り出す。
3. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{A}_i\in\mathbb{F}_q^{n\times n}$（上三角行列），$\mathbf{b}_i\in\mathbb{F}_q^n$ $(i=1,\ldots,m)$ を得る.
4. $\mathrm{Expand}(h_1)$ の計算により，$\gamma_1^{[e]},\ldots,\gamma_m^{[e]}$ $(e=1,\ldots,\tau)$ を得る。
5. $\mathrm{Expand}(h_2)$ の計算により，$r^{[1]},\ldots,r^{[\tau]}\in\mathbb{F}_{q^\eta}$ を得る。
6. $\mathrm{Expand}(h_3)$ の計算により，$i^{*[1]},\ldots,i^{*[\tau]}\in\{1,\ldots,N\}$ を得る。
7. $e=1,\ldots,\tau$ に対して以下を行う：
   - 7-1. $\mathrm{broad}^{[e]}$ から $(c_1^{[e]},\ldots,c_{n_2}^{[e]},v^{[e]})$ を取り出す。
   - 7-2. $\mathrm{view}^{[e]}$ から以下を取り出す：
     - $i^{*[e]}\neq N$ ならば，$\{\mathrm{seed}_j^{[e]}\}_{j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}},\ [\![\mathbf{x}^{*[e]}]\!]_N,\ [\![Q'^{[e]}]\!]_N(u)$
     - $i^{*[e]}=N$ ならば，$\{\mathrm{seed}_j^{[e]}\}_{j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}}$
   - 7-3. すべての $j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}$ に対し，$\mathrm{Expand}(\mathrm{salt},\ \mathrm{seed}_j^{[e]})$ の計算により以下を得る：
     - $j<N$ ならば，$[\![\mathbf{x}^{*[e]}]\!]_j,\ [\![a_1^{[e]}]\!]_j,\ldots,[\![a_{n_2}^{[e]}]\!]_j,\ [\![Q'^{[e]}]\!]_j(u)$
     - $j=N$ ならば，$[\![a_1^{[e]}]\!]_N,\ldots,[\![a_{n_2}^{[e]}]\!]_N$
   - 7-4. すべての $j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}$ に対し，f_{MQ} における $\mathcal{P}_j$ と同じ計算より，$[\![\mathbf{x}^{*[e]}]\!]_j,\ [\![a_1^{[e]}]\!]_j,\ldots,[\![a_{n_2}^{[e]}]\!]_j,\ [\![Q'^{[e]}]\!]_j(u),\ c_1^{[e]},\ldots,c_{n_2}^{[e]}$ から，$[\![\mathrm{broad}^{[e]}]\!]_j=([\![c_1^{[e]}]\!]_j,\ldots,[\![c_{n_2}^{[e]}]\!]_j,\ [\![v^{[e]}]\!]_j)$ を計算する。
   - 7-5. $[\![\mathrm{broad}^{[e]}]\!]_{i^{*[e]}}=\mathrm{broad}^{[e]}-\sum_{j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}}[\![\mathrm{broad}^{[e]}]\!]_j$ を計算する。
   - 7-6. すべての $j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}$ に対し，以下のようにコミットメントを計算する：
     - $j<N$ ならば，$\mathrm{com}_j^{[e]}=\mathrm{Commit}(\mathrm{salt},\ e,\ j,\ \mathrm{seed}_j^{[e]})$
     - $j=N$ ならば，$\mathrm{com}_j^{[e]}=\mathrm{Commit}(\mathrm{salt},\ e,\ N,\ \mathrm{seed}_N^{[e]},\ [\![\mathbf{x}^{*[e]}]\!]_N)$
       $\mathrm{com}''^{[e]}_j=\mathrm{Commit}(\mathrm{salt},\ e,\ 0,\ [\![Q'^{[e]}]\!]_N(u))$
8. $h_1'=\mathcal{H}_1(M,\ \mathrm{salt},\ \mathrm{com}_1^{[1]},\ldots,\mathrm{com}_N^{[\tau]})$ を計算する。
9. $h_2'=\mathcal{H}_2(M,\ \mathrm{salt},\ h_1',\ \mathrm{com}'^{[1]}_N,\ldots,\mathrm{com}'^{[\tau]}_N)$ を計算する。
10. $h_3'=\mathcal{H}_3(M,\ \mathrm{salt},\ h_2',\ [\![\mathrm{broad}^{[1]}]\!],\ldots,[\![\mathrm{broad}^{[\tau]}]\!])$ を計算する。
11. $i^{*[e]}\neq N$ なる $e\in\{1,\ldots,\tau\}$ で，$\mathrm{com}''^{[e]}_N\neq\mathrm{com}'^{[e]}_N$ となるものがあれば，‘棄却’ を返す。
12. $v^{[e]}\neq 0$ なる $e\in\{1,\ldots,\tau\}$ があれば，‘棄却’ を返す。
13. $(h_1',h_2',h_3')\neq(h_1,h_2,h_3)$ ならば，‘棄却’ を返す。それ以外は ‘受理’ を返す。

5.3.4.2 MQOM のパラメータ選択

MQOM の設計に必要なパラメータは，$\lambda,q,m,n,n_1,n_2,\eta,N,\tau$ である。NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出されたドキュメント [21] では，さらに効率性向上のテクニック（hypercube optimization，seed tree など）が追加されており，それを踏まえて以下のように MQOM のパラメータの見積もりが公開されている。

$(\lambda, q, m(=n), n_1, n_2, \eta, N, \tau)$	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ（平均）
$(128, 31, 49, 5, 10, 10, 256, 20)$	レベル 1	47 Bytes	78 Bytes	6,348 Bytes
$(128, 31, 49, 5, 10, 6, 32, 35)$	レベル 1	59 Bytes	102 Bytes	6,575 Bytes
$(128, 251, 43, 4, 11, 5, 256, 22)$	レベル 1	47 Bytes	78 Bytes	7,621 Bytes
$(128, 251, 43, 4, 11, 4, 32, 34)$	レベル 1	59 Bytes	102 Bytes	7,809 Bytes
$(192, 31, 77, 6, 13, 11, 256, 30)$	レベル 3	73 Bytes	122 Bytes	13,837 Bytes
$(192, 31, 77, 6, 13, 7, 32, 51)$	レベル 3	92 Bytes	160 Bytes	14,257 Bytes
$(192, 251, 68, 5, 14, 7, 256, 30)$	レベル 3	73 Bytes	122 Bytes	16,590 Bytes
$(192, 251, 68, 5, 14, 4, 32, 52)$	レベル 3	92 Bytes	160 Bytes	17,161 Bytes
$(256, 31, 106, 6, 18, 10, 256, 42)$	レベル 5	99 Bytes	166 Bytes	24,147 Bytes
$(256, 31, 106, 6, 18, 8, 32, 66)$	レベル 5	125 Bytes	218 Bytes	24,926 Bytes
$(256, 251, 93, 6, 16, 7, 256, 41)$	レベル 5	99 Bytes	166 Bytes	28,917 Bytes
$(256, 251, 93, 6, 16, 5, 32, 66)$	レベル 5	125 Bytes	218 Bytes	29,919 Bytes

5.3.5 署名方式 MiRitH

5.3.5.1 MiRitH の概要

MiRitH [3] は，MinRank 問題に関する秘匿マルチパーティ計算から MPC-in-the-Head で構成された署名方式である。MinRank 問題は 5.1.3 節でも述べたが，次のように表現することもできる。

MinRank 問題（別バージョン） 正の整数 r と行列 $\mathrm{M}_0, \ldots, \mathrm{M}_k \in \mathbb{F}_q^{m \times n}$ に対し，$\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q$ で，

$$\mathrm{Rank}\left(\mathrm{M}_0 + \sum_{i=1}^{k} \alpha_i \mathrm{M}_i\right) \leq r$$

なるものを求めよ。

もし，$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k) \in \mathbb{F}_q^k$ と $\mathrm{K} \in \mathbb{F}_q^{r \times (n-r)}$ が存在し，

$$\left(\mathrm{M}_0 + \sum_{i=1}^{k} \alpha_i \mathrm{M}_i\right) \cdot \begin{pmatrix} \mathbf{I}_{n-r} \\ \mathrm{K} \end{pmatrix} = \mathbf{0}_{m \times (n-r)} \tag{5.11}$$

となるならば，$\boldsymbol{\alpha}$ は MinRank 問題の解である。$\mathbf{M} = (\mathrm{M}_0, \ldots, \mathrm{M}_k)$ に対し，$\mathbf{M}_{\boldsymbol{\alpha}} \in \mathbb{F}_q^{m \times n}$ を

$$\mathbf{M}_{\boldsymbol{\alpha}} = \mathrm{M}_0 + \sum_{i=1}^{k} \alpha_i \mathrm{M}_i$$

とし，$\mathbf{M}_{\boldsymbol{\alpha}}^L \in \mathbb{F}_q^{m \times (n-r)}$, $\mathbf{M}_{\boldsymbol{\alpha}}^R \in \mathbb{F}_q^{m \times r}$ をそれぞれ，$\mathbf{M}_{\boldsymbol{\alpha}}$ の左側 $n-r$ 列，$\mathbf{M}_{\boldsymbol{\alpha}}$ の右側 r 列で定めると，(5.11) は $\mathbf{M}_{\boldsymbol{\alpha}}^L = \mathbf{M}_{\boldsymbol{\alpha}}^R \cdot \mathrm{K}$ と同値である。そこで，（秘匿マルチパーティ計算において設定される関係 $\mathcal{R}$ の）命題を MinRank 問題のインスタンスとし，その証拠を $\boldsymbol{\alpha}, \mathrm{K}$ としておけば，$\boldsymbol{\alpha}$ が MinRank 問題の解であることが（$\mathbf{M}_{\boldsymbol{\alpha}}$ のランクを直接計算をせずとも）効率的に検証できる。以下の検証プロトコルでは，ランダム行列 $\mathrm{R} \in \mathbb{F}_q^{s \times m}$ を用意し，$\mathrm{V} = \mathrm{R}\,(\mathbf{M}_{\boldsymbol{\alpha}}^L - \mathbf{M}_{\boldsymbol{\alpha}}^R \cdot \mathrm{K})$ とおき，$\mathrm{V} = \mathbf{0}_{s \times (n-r)}$ となることで (5.11) を確認する。(5.11) が成り立つときは $\mathrm{V} = \mathbf{0}_{s \times (n-r)}$ が必ず成り立つが，(5.11) が成り立たないときに $\mathrm{V} = \mathbf{0}_{s \times (n-r)}$ となる確率は約 $1/q^s$ である。

安全性パラメータを λ とし，以下の関数を用意する。

- Expand： 任意の λ-ビット列を入力とする疑似乱数生成関数
- $\mathcal{H}_1, \mathcal{H}_2 : \{0,1\}^* \to \{0,1\}^{2\lambda}$ ：暗号学的ハッシュ関数

- Commit：コミットメント関数

鍵生成

1. $\mathrm{seed}_{\mathrm{pk}},\ \mathrm{seed}_{\mathrm{sk}} \in \{0,1\}^{\lambda}$ をランダムに選ぶ。
2. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{M}_1, \ldots, \mathrm{M}_k \in \mathbb{F}_q^{m \times n}$ を得る.
3. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{sk}})$ の計算により，$\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q,\ \mathrm{K} \in \mathbb{F}_q^{r \times (n-r)},\ \mathrm{E}^R \in \mathbb{F}_q^{m \times r}$ を得る。
4. $\mathrm{E}^R\,\mathrm{K}$ を計算し，左側の $n-r$ 列を $\mathrm{E}^R\,\mathrm{K}$，右側の r 列を E^R として定まる行列を $\mathrm{E} \in \mathbb{F}_q^{m \times n}$ とする。
5. $\mathrm{M}_0 = \mathrm{E} - \sum_{\ell=1}^{k} \alpha_\ell\,\mathrm{M}_\ell$ を計算する。

公開鍵は $\mathrm{pk} = (\mathrm{seed}_{\mathrm{pk}}, \mathrm{M}_0)$，秘密鍵は $\mathrm{sk} = \mathrm{seed}_{\mathrm{sk}}$ である。次に，署名生成である。メッセージを $M \in \{0,1\}^*$ とする。

署名生成

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \mathrm{M}_0)$ を取り出す。
2. sk から $\mathrm{seed}_{\mathrm{sk}}$ を取り出す。
3. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{M}_1, \ldots, \mathrm{M}_k \in \mathbb{F}_q^{m \times n}$ を得る.
4. $\mathrm{salt} \in \{0,1\}^{2\lambda}$ をランダムに選ぶ。
5. $e = 1, \ldots, \tau$ に対して以下を計算する：
   - 5-1. $\mathrm{seed}^{[e]} \in \{0,1\}^{\lambda}$ をランダムに選ぶ。
   - 5-2. $\mathrm{Expand}(\mathrm{salt},\ \mathrm{seed}^{[e]})$ の計算により，$\mathrm{seed}_j^{[e]} \in \{0,1\}^{\lambda}\ \ (j = 1, \ldots, N)$ を得る。
   - 5-3. すべての $j = 1, \ldots, N$ に対し，$\mathrm{Expand}(\mathrm{salt}, \mathrm{seed}_j^{[e]})$ の計算により以下を得る：
     - $j < N$ ならば，$[\![\mathrm{A}^{[e]}]\!]_j \in \mathbb{F}_q^{s \times r}$, $[\![\alpha_1^{[e]}]\!]_j, \ldots, [\![\alpha_k^{[e]}]\!]_j\ (\in \mathbb{F}_q)$, $[\![\mathrm{K}^{[e]}]\!]_j \in \mathbb{F}_q^{r \times (n-r)}$, $[\![\mathrm{C}^{[e]}]\!]_j \in \mathrm{F}_q^{s \times (n-r)}$
     - $j = N$ ならば，$[\![\mathrm{A}^{[e]}]\!]_N \in \mathbb{F}_q^{s \times r}$
   - 5-4. $[\![\alpha_\ell^{[e]}]\!]_N = \alpha_\ell - \sum_{j=1}^{N-1} [\![\alpha_\ell^{[e]}]\!]_j\ \ (\ell = 1, \ldots, k)$ を計算する。
   - 5-5. $[\![\mathrm{K}^{[e]}]\!]_N = \mathrm{K} - \sum_{j=1}^{N-1} [\![\mathrm{K}^{[e]}]\!]_j$，$[\![\mathrm{C}^{[e]}]\!]_N = \mathrm{A}^{[e]}\,\mathrm{K} - \sum_{j=1}^{N-1} [\![\mathrm{C}^{[e]}]\!]_j$ を計算する。
   - 5-6. $\mathrm{state}_j^{[e]} = \begin{cases} (\mathrm{seed}_j^{[e]}) & j = 1, \ldots, N-1 \\ (\mathrm{seed}_N^{[e]}, [\![\alpha_1^{[e]}]\!]_N, \ldots, [\![\alpha_k^{[e]}]\!]_N, [\![\mathrm{K}^{[e]}]\!]_N, [\![\mathrm{C}^{[e]}]\!]_N) & j = N \end{cases}$ とおく。
   - 5-7. コミットメント $\mathrm{com}_j^{[e]} = \mathrm{Commit}(\mathrm{salt}, e, j, \mathrm{state}_j^{[e]})\ \ (j = 1, \ldots, N)$ を計算する。
6. $h_1 = \mathcal{H}_1(M, \mathrm{salt}, \mathrm{com}_1^{[1]}, \ldots, \mathrm{com}_N^{[\tau]})$ を計算する。
7. $\mathrm{Expand}(h_1)$ の計算により，$\mathrm{R}^{[e]} \in \mathbb{F}_q^{s \times m}\ \ (e = 1, \ldots, \tau)$ を得る.
8. $e = 1, \ldots, \tau$ に対して以下を計算する：
   - 8-1. $[\![\alpha_1^{[e]}]\!], \ldots, [\![\alpha_k^{[e]}]\!]$ より，$[\![\mathbf{M}_{\boldsymbol{\alpha}}^{L\,[e]}]\!]$, $[\![\mathbf{M}_{\boldsymbol{\alpha}}^{R\,[e]}]\!]$ を計算する。
   - 8-2. $[\![\mathrm{S}^{[e]}]\!] = \mathrm{R}^{[e]}\,[\![\mathbf{M}_{\boldsymbol{\alpha}}^{R\,[e]}]\!] + [\![\mathrm{A}^{[e]}]\!]$ を計算する。
   - 8-3. $\mathrm{S}^{[e]} = \sum_{j=1}^{N} [\![\mathrm{S}^{[e]}]\!]_j$ を計算する。
   - 8-4. $[\![\mathrm{V}^{[e]}]\!] = \mathrm{S}^{[e]}\,[\![\mathrm{K}^{[e]}]\!] - \mathrm{R}^{[e]}\,[\![\mathbf{M}_{\boldsymbol{\alpha}}^{L\,[e]}]\!] - [\![\mathrm{C}^{[e]}]\!]$ を計算する。
   - 8-5. $\mathrm{V}^{[e]} = \sum_{j=1}^{N} [\![\mathrm{V}^{[e]}]\!]_j$ を計算する。
   - 8-6. $[\![\mathrm{broad}^{[e]}]\!] = ([\![\mathrm{S}^{[e]}]\!], [\![\mathrm{V}^{[e]}]\!])$, $\mathrm{broad}^{[e]} = (\mathrm{S}^{[e]}, \mathrm{V}^{[e]})$ とおく。
9. $h_2 = \mathcal{H}_2(M, \mathrm{salt}, h_1, [\![\mathrm{broad}^{[1]}]\!], \cdots, [\![\mathrm{broad}^{[\tau]}]\!])$ を計算する。
10. $\mathrm{Expand}(h_2)$ の計算により，$i^{*[1]}, \ldots, i^{*[\tau]} \in \{1, \ldots, N\}$ を得る。
11. $\mathrm{view}^{[e]}$ をすべての $j \in \{1, \ldots, N\} \backslash \{i^{*[e]}\}$ に対する $\mathrm{state}_j^{[e]}$ のリストとする。
12. $\sigma = (\mathrm{salt}, h_1, h_2, \{\mathrm{view}^{[e]}, \mathrm{broad}^{[e]}, \mathrm{com}_{i^{*[e]}}\}_{e=1,\ldots,\tau})$ とおく。

σ が署名となる。最後に検証である。

検証

1. pk から $(\mathrm{seed}_{\mathrm{pk}}, \mathrm{M}_0)$ を取り出す。
2. σ から $(\mathrm{salt}, h_1, h_2, \{\mathrm{view}^{[e]}, \mathrm{broad}^{[e]}, \mathrm{com}_{i^{*[e]}}\}_{e=1,\ldots,\tau})$ を取り出す。
3. $\mathrm{Expand}(\mathrm{seed}_{\mathrm{pk}})$ の計算により，$\mathrm{M}_1, \ldots, \mathrm{M}_k \in \mathbb{F}_q^{m\times n}$ を得る.
4. $\mathrm{Expand}(h_1)$ の計算により，$\mathrm{R}^{[e]} \in \mathbb{F}_q^{s\times m}$ $(e = 1, \ldots, \tau)$ を得る.
5. $\mathrm{Expand}(h_2)$ の計算により，$i^{*[1]}, \ldots, i^{*[\tau]} \in \{1, \ldots, N\}$ を得る。
6. $e = 1, \ldots, \tau$ に対して以下を計算する：
   6-1. $\mathrm{broad}^{[e]}$ から $(\mathrm{S}^{[e]}, \mathrm{V}^{[e]})$ を取り出す。
   6-2. $\mathrm{view}^{[e]}$ から $(\mathrm{state}_j^{[e]})_{j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}}$ を取り出す。
   6-3. コミットメント $\mathrm{com}_j^{[e]} = \mathrm{Commit}(\mathrm{salt}, e, j, \mathrm{state}_j^{[e]})$ $(j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\})$ を計算する。
   6-4. すべての $j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\}$ に対し，$\mathrm{state}_j^{[e]}$ により以下を得る：
      · $j < N$ ならば，$\mathrm{seed}_j^{[e]}$
      · $j = N$ ならば，$\mathrm{seed}_N^{[e]}, [\![\alpha_1^{[e]}]\!]_N, \ldots, [\![\alpha_k^{[e]}]\!]_N, [\![\mathrm{K}^{[e]}]\!]_N, [\![\mathrm{C}^{[e]}]\!]_N$
   6-5. すべての $j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\}$ に対し，$\mathrm{Expand}(\mathrm{salt}, \mathrm{seed}_j^{[e]})$ の計算により以下を得る：
      · $j < N$ ならば，$[\![\mathrm{A}^{[e]}]\!]_j, [\![\alpha_1^{[e]}]\!]_j, \ldots, [\![\alpha_k^{[e]}]\!]_j, [\![\mathrm{K}^{[e]}]\!]_j, [\![\mathrm{C}^{[e]}]\!]_j$
      · $j = N$ ならば，$[\![\mathrm{A}^{[e]}]\!]_N$
   6-6. すべての $j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\}$ に対し，$[\![\alpha_1^{[e]}]\!]_j, \ldots, [\![\alpha_k^{[e]}]\!]_j$ より，$[\![\mathbf{M}_{\boldsymbol{\alpha}}^{L\,[e]}]\!]_j, [\![\mathbf{M}_{\boldsymbol{\alpha}}^{R\,[e]}]\!]_j$ を計算する。
   6-7. $[\![\mathrm{S}^{[e]}]\!]_j = \mathrm{R}^{[e]}\,[\![\mathbf{M}_{\boldsymbol{\alpha}}^{R\,[e]}]\!]_j + [\![\mathrm{A}^{[e]}]\!]_j$ $(j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\})$ を計算する。
   6-8. $[\![\mathrm{V}^{[e]}]\!]_j = \mathrm{S}^{[e]}\,[\![\mathrm{K}^{[e]}]\!]_j - \mathrm{R}^{[e]}\,[\![\mathbf{M}_{\boldsymbol{\alpha}}^{L\,[e]}]\!]_j - [\![\mathrm{C}^{[e]}]\!]_j$ $(j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\})$ を計算する。
   6-9. $[\![\mathrm{broad}^{[e]}]\!]_j = ([\![\mathrm{S}^{[e]}]\!]_j, [\![\mathrm{V}^{[e]}]\!]_j)$ $(j \in \{1, \ldots, N\}\backslash\{i^{*[e]}\})$ とおく。
   6-10. $[\![\mathrm{broad}^{[e]}]\!]_{i^{*[e]}} = \mathrm{broad}^{[e]} - \sum_{j\in\{1,\ldots,N\}\backslash\{i^{*[e]}\}} [\![\mathrm{broad}^{[e]}]\!]_j$ を計算する。
7. $h_1' = \mathcal{H}_1(M, \mathrm{salt}, \mathrm{com}_1^{[1]}, \ldots, \mathrm{com}_N^{[\tau]})$ を計算する。
8. $h_2' = \mathcal{H}_2(M, \mathrm{salt}, h_1', [\![\mathrm{broad}^{[1]}]\!], \cdots, [\![\mathrm{broad}^{[\tau]}]\!])$ を計算する。
9. $\mathrm{V}^{[e]} \neq \mathbf{0}_{s\times(n-r)}$ なる $e \in \{1, \ldots, \tau\}$ があれば，'棄却' を返す。
10. $(h_1', h_2') \neq (h_1, h_2)$ ならば，'棄却' を返す。それ以外は '受理' を返す。

5.3.5.2 MiRitH のパラメータ選択

MiRitH の設計に必要なパラメータは，$\lambda, q, m, n, k, r, s, N, \tau$ である。NIST PQC 標準化プロジェクト追加署名第 1 ラウンドに提出されたドキュメント [3] では，さらに効率性向上のテクニック（hypercube optimization，seed tree など）が追加されており，それを踏まえて以下のように MiRitH のパラメータの見積もりが公開されている。

$(\lambda, q, m(=n), k, r, s, N, \tau)$	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ（平均）
$(128, 16, 15, 78, 6, 5, 16, 39)$	レベル 1	129 Bytes	16 Bytes	7,661 Bytes
$(128, 16, 15, 78, 6, 9, 256, 19)$	レベル 1	129 Bytes	16 Bytes	5,665 Bytes
$(128, 16, 16, 142, 4, 5, 16, 39)$	レベル 1	144 Bytes	16 Bytes	8,800 Bytes
$(128, 16, 16, 142, 4, 9, 256, 19)$	レベル 1	144 Bytes	16 Bytes	6,298 Bytes
$(192, 16, 19, 109, 8, 7, 16, 55)$	レベル 3	205 Bytes	24 Bytes	16,668 Bytes
$(192, 16, 19, 109, 8, 9, 256, 29)$	レベル 3	205 Bytes	24 Bytes	12,423 Bytes
$(192, 16, 19, 167, 6, 7, 16, 55)$	レベル 3	205 Bytes	24 Bytes	17,882 Bytes
$(192, 16, 19, 167, 6, 9, 256, 29)$	レベル 3	205 Bytes	24 Bytes	13,115 Bytes
$(256, 16, 21, 189, 7, 7, 16, 74)$	レベル 5	253 Bytes	32 Bytes	29,568 Bytes
$(256, 16, 21, 189, 7, 10, 256, 38)$	レベル 5	253 Bytes	32 Bytes	21,763 Bytes
$(256, 16, 22, 254, 6, 7, 16, 74)$	レベル 5	274 Bytes	32 Bytes	31,980 Bytes
$(256, 16, 22, 254, 6, 10, 256, 38)$	レベル 5	274 Bytes	32 Bytes	23,144 Bytes

5.4 多変数多項式に基づく暗号技術に関するまとめ

1984 年に，Ong と Schnorr が多変数 2 次多項式を利用した署名方式 [31] を提案した。したがって，多変数多項式を利用した暗号技術は，既に 40 年以上の歴史を持つことになる。Ong と Schnorr の署名方式は，合成数を法とする剰余環を係数としており，合成数の素因数分解ができないことを安全性の仮定としていたが，1988 年に，松本と今井により，初めて有限体を係数とした多変数多項式を利用した暗号化方式 [30] が提案された。これ以降，MQ 問題の解読困難性を安全性の仮定とする方式が数多く提案されており，現在に至る。

多変数公開鍵暗号の暗号化方式，および，署名方式の多くは双極型システムを用いて構成されている。双極型システムは，暗号化や検証が効率的に実行できるという特徴を持つ。双極型システムを用いて構成されている署名方式 UOV も，この特徴を持ち，さらに，署名長が短いという特徴も持っている。一方で，双極型システムは公開鍵長が大きくなりやすいという課題もある。UOV に対しては，公開鍵長を削減する改良を加えた変種として QR-UOV や MAYO が提案されている。

一方で，MQ 問題や MinRank 問題に付随する秘匿マルチパーティ計算から MPC-in-the-Head の枠組みを利用して署名方式を構成することができる。こちらは方式が提案されてからまだ数年しかたっていないということもあり，今後の研究動向を見守る必要がある。

第 5 章の参照文献

[1] W. W. Adams, P. Loustaunau. An Introduction to Gröbner Bases. Vol. 3. Graduate Studies in Mathematics. American Mathematical Society.

[2] G. Adj, L. Rivera-Zamarripa, J. A. Verbel. MinRank in the Head: Short Signatures from Zero-Knowledge Proofs. (2022). https://eprint.iacr.org/2022/1501.

[3] G. Adj, L. Rivera-Zamarripa, J. A. Verbel, E. Bellini, S. Barbero, A. Esser, C. Sanna, F. Zweydinger. MiRitH. 2022-08. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/mirith-submission.zip. Submission to the NIST's Post-Quantum Cryptography: Additional Digital Signature Schemes, round 1.

[4] M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. A. Perlner, D. Smith-Tone, J.-P. Tillich, J. A. Verbel. Improvements of Algebraic Attacks for Solving the Rank Decoding and MinRank Problems. ASIACRYPT (1). Vol. 12491. Lecture Notes in Computer Science. Springer, 2020, pp. 507–536.

[5] M. Bardet, J.-C. Faugère, B. Salvy, B.-Y. Yang. Asymptotic expansion of the degree of regularity for semi-regular systems of equations. Effective Methods in Algebraic Geometry (MEGA). 2004, pp. 71–74.

[6] E. Bellini, A. Esser, C. Sanna, J. Verbel. MR-DSS – Smaller MinRank-based (Ring-)Signatures. (2022). https://eprint.iacr.org/2022/973.

[7] W. Beullens. Improved cryptanalysis of UOV and Rainbow. Cryptology ePrint Archive, Paper 2020/1343. 2020. https://eprint.iacr.org/2020/1343.

[8] W. Beullens. MAYO: Practical Post-quantum Signatures from Oil-and-Vinegar Maps. SAC. Vol. 13203. Lecture Notes in Computer Science. Springer, 2021, pp. 355–376.

[9] W. Beullens, F. Campos, S. Celi, B. Hess, M. J. Kannwischer. MAYO. 2022-08. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/mayo-submission.zip. Submission to the NIST's Post-Quantum Cryptography: Additional Digital Signature Schemes, round 1.

[10] W. Beullens et al. UOV. 2022-08. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/UOV-submission.zip. Submission to the NIST's Post-Quantum Cryptography: Additional Digital Signature Schemes, round 1.

[11] Ward Beullens. Breaking Rainbow Takes a Weekend on a Laptop. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 464–479.

[12] A. Caminata, E. Gorla. Solving degree, last fall degree, and related invariants. Journal of Symbolic Computation. Vol. 114 (2023), pp. 322–335.

[13] A. Casanova, J.-C. Faugere, G. Macario-Rat, J. Patarin, L. Perret, J. Ryckeghem. GeMSS. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/GeMSS-Round2.zip. Submission to the NIST's Post-Quantum Cryptography, round 2.

[14] N. T. Courtois. Efficient Zero-Knowledge Authentication Based on a Linear Algebra Problem MinRank. ASIACRYPT. Vol. 2248. Lecture Notes in Computer Science. Springer, 2001, pp. 402–421.

[15] J. Ding, J. E. Gower, D. S. Schmidt. Multivariate Public Key Cryptosystems. Vol. 25. Advances in Information Security. Springer, 2006.

[16] J. Ding, D. Schmidt. Rainbow, a New Multivariable Polynomial Signature Scheme. ACNS. Vol. 3531. Lecture Notes in Computer Science. 2005, pp. 164–175.

[17] V. Dubois, N. Gama. The Degree of Regularity of HFE Systems. ASIACRYPT. Vol. 6477. Lecture Notes in Computer Science. Springer, 2010, pp. 557–576.

[18] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). J. Pure Appl. Algebra. Vol. 139 (1999), pp. 61–88.

[19] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). ISSAC. ACM, 2002, pp. 75–83.

[20] J.-C. Faugère, M. S. El Din, P.-J. Spaenlehauer. Computing loci of rank defects of linear matrices using Gröbner bases and applications to cryptology. ISSAC. ACM, 2010, pp. 257–264.

[21] T. Feneuil, M. Rivain. MQOM (MQ on my mind). 2022-08. `https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/mqom-submission.zip`. Submission to the NIST's Post-Quantum Cryptography: Additional Digital Signature Schemes, round 1.

[22] H. Furue, Y. Ikematsu, Y. Kiyomura, T. Takagi. A New Variant of Unbalanced Oil and Vinegar Using Quotient Ring: QR-UOV. ASIACRYPT (4). Vol. 13093. Lecture Notes in Computer Science. Springer, 2021, pp. 187–217.

[23] H. Furue, Y. Ikematsu, Y. Kiyomura, T. Takagi, K. Yasuda, T. Miyazawa, T. Saito, A. Nagai. QR-UOV. 2022-08. `https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/submission-pkg/QR-UOV-submission.zip`. Submission to the NIST's Post-Quantum Cryptography: Additional Digital Signature Schemes, round 1.

[24] M. R. Garay, D. S. Johnson. A Guide to the Theory of NP-Completeness. In Computers and Intractability. W.H. Freeman, 1979.

[25] L. Goubin, N.T. Courtois. Cryptanalysis of the TTM Cryptosystem. ASIACRYPT. Vol. 1976. Lecture Notes in Computer Science. Springer, 2000, pp. 44–57.

[26] Y. Hashimoto. An improvement of algorithms to solve under-defined systems of multivariate quadratic equations. JSIAM Letters. Vol. 15 (2023), pp. 53–56. `https://www.jstage.jst.go.jp/article/jsiaml/15/0/15_53/_article`.

[27] Y. Ishai, E. Kushilevitz, R. Ostrovsky, A. Sahai. Zero-knowledge from secure multiparty computation. STOC. ACM, 2007, pp. 21–30.

[28] A. Kipnis, L. Patarin, L. Goubin. Unbalanced Oil and Vinegar Signature Schemes. EUROCRYPT. Vol. 1592. Lecture Notes in Computer Science. Springer, 1999, pp. 206–222.

[29] A. Kipnis, A. Shamir. Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. CRYPTO. Vol. 1666. Lecture Notes in Computer Science. Springer, 1999, pp. 19–30.

[30] T. Matsumoto, H. Imai. Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. EUROCRYPT. Vol. 330. Lecture Notes in Computer Science. Springer, 1988, pp. 419–453.

[31] H. Ong, C.-P. Schnorr. Signatures through Approximate Representation by Quadratic Forms. CRYPTO. Plenum Press, New York, 1983, pp. 117–131.

[32] J. Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. EUROCRYPT. Vol. 1070. Lecture Notes in Computer Science. Springer, 1996, pp. 33–48.

[33] A. Petzoldt, M.-S. Chen, B.-Y. Yang, C. Tao, J. Ding. Design Principles for HFEv-Based Multivariate Signature Schemes. ASIACRYPT (1). Vol. 9452. Lecture Notes in Computer Science. Springer, 2015, pp. 311–334.

[34] B. Santoso, Y. Ikematsu, S. Nakamura, T. Yasuda. Three-Pass Identification Scheme Based on MinRank Problem with Half Cheating Probability. arXiv: 2205.03255.

[35] C. Tao, A. Petzoldt, J. Ding. Efficient Key Recovery for All HFE Signature Variants. CRYPTO (1). Vol. 12825. Lecture Notes in Computer Science. Springer, 2021, pp. 70–93.

[36] C. Wolf. Taxonomy of public key schemes based on the problem of Multivariate Quadratic equations. Cryptology ePrint Archive, Paper 2005/077. 2005. https://eprint.iacr.org/2005/077.

[37] B.-Y. Yang, J.-M. Chen. All in the XL Family: Theory and Practice. ICISC. Vol. 3506. Lecture Notes in Computer Science. Springer, 2004, pp. 67–86.

第 6 章

同種写像に基づく暗号技術

本章では同種写像に基づく暗号技術についてまとめる。同種写像に基づく暗号技術の安全性は，同種写像問題を解く計算の困難性及び（それと同値な）自己準同型環計算問題の困難性に依存しており，同種写像暗号に関する研究はこれまで継続して進められている。特に，本報告書の 2022 年度版と比べて，高次元同種写像計算を利用した鍵共有・署名構成に進展が見られている（6.2.2.2 節，6.3.1.2 節及び 6.4 節参照）。

6.1 節では, 安全性の根拠となる問題として, 同種写像問題の一般形を述べた後, 最近発見された SIDH（Supersingular Isogeny Diffie–Hellman）同種写像問題 [58] に対する解法について述べる。そして，その攻撃法を回避する計算問題として，レベル構造付き同種写像問題，同種写像に基づく一方向性群作用に関する計算問題，自己準同型環計算問題及び SQIsign（Short Quaternion and Isogeny Signature）署名方式 [60] の安全性に関する計算問題の順に，その概要を記述していく。6.2 節では，代表的な暗号方式として，一方向性群作用に基づく CSIDH （Commutative Supersingular Isogeny Diffie–Hellman）鍵共有 [28] とその変種，SIDH 型の鍵共有方式，CSIDH ベースの SeaSign 署名方式 [57], CSI-FiSh（Commutative Supersingular Isogeny Fiat–Shamir）署名方式 [15]，そして自己準同型環計算問題に基づく GPS（Galbraith–Petit–Silva）署名方式 [71] を取り上げる。6.3 節では，主要な暗号方式として，GPS 署名方式を改良した SQIsign 署名方式を解説する。

本章では，超特異楕円曲線を用いた暗号技術を主に扱う。しかし，通常楕円曲線に基づく CRS （Couveignes–Rostovtsev–Stolbunov）鍵共有法 [41, 108] を改良した De Feo ら [59] の方式は，それ自体は実用的な性能にはまだ遠いが，6.2.1.1 節で説明する CSIDH 鍵共有の原型を与えているという点で重要である。また，群作用暗号を量子マネーへ応用した [119, 89] では，通常楕円曲線が用いられている。

同種写像の数学的詳細については，De Feo の概説記事 [54] や Washington の楕円曲線の教科書 [117] を参照のこと。和書では，相川らによる概説書 [120] において，同種写像暗号に必要な数学も詳しく説明されている。また，Galbraith–Vercauteren による同種写像関連問題のサーベイ [72] も参照する。

■記法　$x \leftarrow_R X$ は，x を有限集合 X から一様ランダムにサンプリングすることを表す。以下では，有限体上に定義された楕円曲線のみを扱い，同種写像暗号では，多くの場合，モンゴメリ型の楕円曲線定義式 $E_{a,b} : by^2 = x^3 + ax^2 + x$ が用いられる。標数 p の有限体 $\mathbb{F}$ 上定義された楕円曲線 E に対し，O_E は E の無限遠点であり，$\mathbb{F}$ の拡大体 $\mathbb{K}$ に対して，$\mathbb{K}$-有理点群は $E(\mathbb{K}) := \{(x, y) \in \mathbb{K}^2 \,|\, (x, y) \text{ は } E \text{ の定義式を満たす}\} \cup \{O_E\}$ で与えられる。また，正整数 r に対して E の r-ねじれ部分群は $E[r] := \{P \in E(\overline{\mathbb{F}}_p) \,|\, rP = O_E\}$ で与えられる。ここで $\overline{\mathbb{F}}_p$ は有限体 $\mathbb{F}_p$ の代数閉包を表す。

6.1 同種写像に基づく暗号技術の安全性の根拠となる問題

6.1.1 節で同種写像問題の一般形を述べた後，6.1.2 節で 2022 年に発見された SIDH 同種写像問題に対する解法について述べ，そして，その攻撃法を回避する計算問題として，レベル構造付き同種写像問題（6.1.3 節），同種写像に基づく一方向性群作用に関する計算問題（6.1.4 節），自己準同型環計算問題と SQIsign 署名方式の安全性に関する計算問題（6.1.5 節）の順に，その概要及びそれら問題に対する解析状況について記述していく。

レベル構造付き同種写像問題に基づく QFESTA 暗号方式・鍵共有 [95] は，現時点で効率面で最も良い同種写像ベース鍵共有法を実現しており，6.2.2.2 節で紹介される。また，同種写像ベースの一方向性群作用に関する計算問題は，様々な暗号応用に対して有用であり，6.2.1 節及び 6.2.3 節でそれぞれ基本的な CSIDH 鍵共有と CSI-FiSh 署名方式が示される。そして，自己準同型環計算問題と関連する計算問題は，NIST PQC 標準化プロジェクト追加署名第 1 ラウンドを通過した SQIsign 署名方式（6.3.1 節）を基礎付けるために重要である。

つまり，6.1.3 節から 6.1.5 節までで示される 3 種の同種写像問題群はそれぞれに重要な応用先を有しており，それらの安全性に関して検討することが同種写像暗号を評価する第一歩である。

6.1.1 同種写像問題の一般形

同種写像とは，2 つの楕円曲線 E, E' の間の写像 φ であり，E の座標 (x, y) の有理式で与えられると共に，楕円曲線の加法構造に関する準同型性，即ち $\varphi(P+Q) = \varphi(P) + \varphi(Q)$，を有する非零写像である。（その正確な定義は，前掲の各文献を参照のこと。）また，E, E' の間に，同種写像 φ が存在する時に，E と E' は同種であるという。

同種写像 φ は，その核 $C = \ker(\varphi)$ によって決まるので，φ の定義域曲線（始点曲線）E に対して φ の値域となる楕円曲線を E/C と書き表す，すなわち，$\varphi : E \to E/C$。核 $C = \ker(\varphi)$ の位数がセキュリティパラメータ λ の多項式サイズであれば，$C = \ker(\varphi)$ の生成元から φ を効率的に計算するアルゴリズムが Vélu によって与えられている [115]。（モンゴメリ型楕円曲線に対する Vélu の公式に関しては，[104] を参照のこと。） 特に核の位数 $\#C$ が小素数になる同種写像を同種写像基本演算として，それらの合成が同種写像暗号での基本的な暗号演算を与えることになる。そして，その合成における基本演算の組み合わせ方法が，秘密鍵情報を与える。

つまり，同種な楕円曲線の間の同種写像を計算することを要求する次の同種写像問題が，具体的な暗号方式の安全性を根拠づける次節以降の諸問題の基本形となる。（超特異同種写像問題と自己準同型環計算問題との計算量的同値性に関しては 6.1.5 節で触れる。）

定義 6.1 (一般形同種写像問題 [72]) *2* つの同種な楕円曲線 E, E' に対して，同種写像 φ を計算せよ。（φ のコンパクトな表現を与えよ。）

ここで，「φ のコンパクトな表現」とは，様々な表現方法が考えられる。例えば，$\deg(\varphi)$ が小素数 ℓ_i によって $\prod_i \ell_i^{e_i}$ となっている場合には，この分解に沿って φ を分解した各 ℓ_i 次同種写像の像に現れる値域楕円曲線（又は j 不変量）の列挙で与えることができる。また，6.1.2 節にて後述する SIDH 同種写像問題（定義 6.2）の設定では，核の生成点が，同種写像のコンパクトな表現を与える。そして，6.2.1.1 節で与えられる CSIDH 鍵共有では虚 2 次整環（オーダー）のイデアル類によって同種写像が表現される。6.1.2 節で述べられる SIDH 鍵共有に対する攻撃法は，楕円曲線同種写像に対する新しい表現法を与えた [106]。最近の高次元同種写像を用いた同種写像暗号の進展は，この新しい同種写像表現法に基づいて行われている。SIDH 同種写像問題，6.1.4 節にて後述する CSIDH-(R)EGA-DL 問題（定義 6.5，6.7）は，定義 6.1 の同種写像問題に基づいて定義される。

定義 6.1 において，φ の次数が多項式サイズであれば，この問題は簡単に解けるので，φ の次数は通常は指数サイズのものを考える。また，CSIDH 鍵共有では $\mathbb{F}_p$-有理な楕円曲線のみを対象とするので，$\overline{\mathbb{F}}_p$-同型であるが $\mathbb{F}_p$-同型でな

いツイスト曲線を判別する必要性があるが，ツイスト曲線は j 不変量では判別できない。これにより，Galbraith ら [72] は j 不変量を使って同種写像問題を定式化しているが，上ではあえて，より素朴な形を採用して，2 つの同種な楕円曲線 E, E' を使って同種写像問題を提示した。

同種写像問題の初期の考察には，自己準同型環計算を扱った Kohel の博士論文 [79] や Galbraith による同種写像問題に関する研究 [68] 及び Couveignes と Rostovtsev–Stolbunov による初期の暗号応用への提案 [41, 108] がある。その後，Charles らによる同種写像に基づいたハッシュ関数の提案 [32] は，同種写像一方向性関数を一方向性の観点からだけでなく，衝突困難性の観点からも見直すことになり，初期の同種写像暗号の研究では重要な役割を果たした。特に，同種写像グラフがエクスパンダーグラフであることに着目して暗号に応用した意義は大きい。

■超特異同種写像問題と通常同種写像問題　標数 p の有限体上の楕円曲線 E の p-ねじれ部分群 $E[p]$ が，$E[p] = \{O_E\}$ の時，E を超特異楕円曲線といい，そうでない時，E を通常楕円曲線という。超特異楕円曲線の j 不変量は，$\mathbb{F}_{p^2}$ の要素である。つまり，超特異 j 不変量の個数は，有限個であり，具体的に $[p/12] + \epsilon$（但し $\epsilon \in \{0, 1, 2\}$）で与えられる。超特異，通常という楕円曲線の性質は，同種写像によって保存されるため，同種写像問題も，この 2 つの性質によって，超特異同種写像問題と通常同種写像問題という 2 つの問題に分類される。

■超特異同種写像問題の計算困難性　超特異同種写像問題は，6.1.4.2 節で述べられる CSIDH 鍵共有や CSI-FiSh 署名方式の安全性に関する計算問題の一般形であり，その計算困難性を評価することは重要である。また，自己準同型環計算問題との関係性については 6.1.5 節を参照のこと。

超特異同種写像問題の古典計算機による解読時間は $\tilde{O}(\sqrt{p})$，量子計算機による解読時間は $\tilde{O}(\sqrt[4]{p})$ と見積もられている。Kohel [79] による超特異同種写像グラフ上のアルゴリズム解析に基づいて，現在最良の古典解読アルゴリズムは Delfs–Galbraith [48] によるもの及びその改良 [110] で，解読時間は $\tilde{O}(\sqrt{p})$ である。Delfs–Galbraith アルゴリズムでは $\mathbb{F}_p$ 上の超特異楕円曲線からなる部分グラフが利用されている。量子解読アルゴリズムは Biasse ら [16] によって時間計算量が $\tilde{O}(\sqrt[4]{p})$ の量子アルゴリズムが知られている。これは，$\mathbb{F}_p$ 上の超特異楕円曲線の同種写像問題に対する準指数時間量子アルゴリズム [36] と Grover アルゴリズムに基づく $\tilde{O}(\sqrt[4]{p})$ の道探索アルゴリズムを結合したものである。

また，Costello ら [40]，Longa ら [85] による報告，Udovenko–Vitto [114] による$IKEp182 Challenge [39] 解読報告，Jaques–Schanck [75] による同種写像問題に対する（量子）安全性評価報告は，いずれも SIDH 鍵共有（及び SIKE 暗号方式 [74]）法への攻撃として提案されているが，多くの部分は一般的な超特異同種写像問題に関する知見としても有効であることに注意する。更に，固定次数の同種写像を計算する問題に対して，CRYPTO 2024 において Benčina ら [10] により改善された古典/量子アルゴリズムが提案されている。

6.1.2　SIDH 同種写像問題とその解法

SIDH 鍵共有（及び SIKE 暗号方式）は，これまで同種写像暗号の中核方式と位置づけられてきたが，SIDH 同種写像問題（定義 6.2）に対して，2022 年に Castryck–Decru [22] に始まる一連の鍵導出解法が発表されて，暗号として完全に破れてしまった。しかし，これらの解法は，SIDH 同種写像問題という補助点の情報を入力に含む問題に対する解法であって，一般の同種写像問題（定義 6.1）には適用できないことに注意する。B-SIDH 鍵共有 [38]，Séta 暗号方式 [62] も本攻撃法により同様に解読可能である。SIDH 鍵共有に対して修正を図ろうとする試みについては，6.2.2 節で述べる。

■SIDH 同種写像問題　SIDH 鍵共有の公開パラメータは $pp_{\mathtt{sidh}} := (\ell_{\mathtt{A}}, \ell_{\mathtt{B}}, e_{\mathtt{A}}, e_{\mathtt{B}}, f; E, P_{\mathtt{A}}, Q_{\mathtt{A}}, P_{\mathtt{B}}, Q_{\mathtt{B}})$ で与えられる。ここで，$p+1 = f \cdot \ell_{\mathtt{A}}^{e_{\mathtt{A}}} \ell_{\mathtt{B}}^{e_{\mathtt{B}}}$ で，p は素数，$\ell_{\mathtt{A}}, \ell_{\mathtt{B}}$ は 2 つの異なる小素数である（f は小さい正整数で，多くの場合 $f = 1$）。例えば，$\ell_{\mathtt{A}} = 2, \ell_{\mathtt{B}} = 3$。$E$ は，$\mathbb{F}_{p^2}$ 上定義された超特異楕円曲線であり，$P_{\mathtt{A}}, Q_{\mathtt{A}}$ は，$E[\ell_{\mathtt{A}}^{e_{\mathtt{A}}}]$ の基底，$P_{\mathtt{B}}, Q_{\mathtt{B}}$ は，$E[\ell_{\mathtt{B}}^{e_{\mathtt{B}}}]$ の基底である。

定義 6.2 (SIDH 同種写像問題 [58, 72]) *SIDH* 鍵共有公開パラメータ $pp_{\mathtt{sidh}}$ と，そこで定義された E と $\ell_{\mathtt{B}}^{e_{\mathtt{B}}}$-同種な $E_{\mathtt{B}}$ と $P'_{\mathtt{A}}, Q'_{\mathtt{A}} \in E_{\mathtt{B}}[\ell_{\mathtt{A}}^{e_{\mathtt{A}}}]$ が与えられた時，$P'_{\mathtt{A}} = \varphi_{\mathtt{B}}(P_{\mathtt{A}}), Q'_{\mathtt{A}} = \varphi_{\mathtt{B}}(Q_{\mathtt{A}})$ となる次数 $\ell_{\mathtt{B}}^{e_{\mathtt{B}}}$ の同種写像 $\varphi_{\mathtt{B}} : E \to E_{\mathtt{B}}$ を計算せよ。特に，$\varphi_{\mathtt{B}}$ の核 $\ker(\varphi_{\mathtt{B}})$ の生成元 $R_{\mathtt{B}} \in E[\ell_{\mathtt{B}}^{e_{\mathtt{B}}}]$ を計算せよ。

以下，SIDH 同種写像問題に対して，Castryck–Decru による鍵導出攻撃 [22]，Maino らによる始点曲線に依らない攻撃法 [86]，Robert による SIDH 問題に対する多項式時間攻撃 [105] の順に概説する。

■Castryck–Decru による鍵導出攻撃 [22]　始点曲線 E が特殊極値整環 $\mathcal{O}_0$ を自己準同型環にもつ場合に主に限られるが[*1]，アーベル曲面間の同種写像が分解するかどうかという事実を使って SIDH 問題にアプローチした最初の論文である。特筆すべきは，SIKE Challenge [39] に挙げられていたパラメータ $IKE217 や SIKE パラメータ SIKEp434, SIKEp503，SIKEp610，SIKEp751 をいずれも現実的な時間内で解読することに成功したことである。後続の実装報告 [98] では，NIST 安全性レベル 5 に相当するとされていたパラメータ SIKEp751 が通常の PC（Intel Core i7-9750H CPU）で 1-2 時間程度で解けることが報告されている。

以下に示す Kani の補題が鍵となっている： 秘密同種写像の次数 $N_{\mathtt{B}} := \ell_{\mathtt{B}}^{e_{\mathtt{B}}}$ と互いに素な次数 a の同種写像 $\alpha : E \to E'$ が与えられれば，α と SIDH 同種写像問題の入力から楕円曲線積の間の $(a + N_{\mathtt{B}}, a + N_{\mathtt{B}})$-同種写像 $F : E \times E'' \to E_{\mathtt{B}} \times E'$ が構成可能になる。ここで Castryck–Decru [22] は，$N_{\mathtt{A}} := \ell_{\mathtt{A}}^{e_{\mathtt{A}}}, \ell_{\mathtt{A}} = 2, \ell_{\mathtt{B}} = 3$ として $N_{\mathtt{A}} > N_{\mathtt{B}}$ の下で，$a := N_{\mathtt{A}} - N_{\mathtt{B}}$ が $a = a_1^2 + 4a_2^2$ と 2 整数 a_1, a_2 による表現をもつ場合に有効な攻撃法を始点曲線 E が特殊極値的な場合に示した（特殊極値的楕円曲線に関しては 6.1.5.2 節を参照）。更に，その攻撃では，$a + N_{\mathtt{B}} = (N_{\mathtt{A}} - N_{\mathtt{B}}) + N_{\mathtt{B}} = N_{\mathtt{A}} = 2^{e_{\mathtt{A}}}$ であるので，F は $(2^{e_{\mathtt{A}}}, 2^{e_{\mathtt{A}}})$-同種写像，つまり Richelot 同種写像の列となっている。Richelot 同種写像列の計算や Richelot 同種写像のなすグラフに関する研究はこれまでも種数 2 同種写像暗号と関連して行われてきており [113, 78, 24, 63, 45, 42]，それらの研究と関連した形で Castryck–Decru 攻撃法が実現されている。

■Maino らによる始点曲線に依らない攻撃法 [86]　Maino らによる論文 [86] では，始点曲線に依らない攻撃法を実現するために，Kani の補題で必要な次数 a が平滑（smooth）になる場合の解析を進めて，De Feo の寄与も取り込んで準指数時間攻撃法を実現した。更に，Castryck–Decru 法では秘密同種写像 $\varphi_{\mathtt{B}}$ を徐々に部分的な同種写像を決定していく方法であったが，Maino ら [86] は，より直接的に 1 度の Kani の補題適用により $\varphi_{\mathtt{B}}$ を見つけ出すアルゴリズムに改良した。

■Robert による SIDH 問題に対する多項式時間攻撃 [105]　Robert [105] では，楕円曲線 8 つの直積の間の同種写像 F を上述の Kani の補題の一般化より得て，その分解性を使って多項式時間が証明可能な SIDH 攻撃アルゴリズムを構成した。さらに，[105] では，Castryck–Decru 法，Maino らによる方法を「$2g$ 次元攻撃法」という形で統合した。それによりそれぞれの方法の利点と課題が分かりやすい形で把握できるようになった。

例えば，[105] の主結果は，上述のように 8 次元アーベル多様体間の同種写像計算に基づいた証明可能多項式時間「8 次元攻撃法」アルゴリズムの提示にあるが，一方，ヒューリスティック多項式時間の「4 次元攻撃法」アルゴリズム [105, 系 4.5, 命題 4.6] についても詳細な解析を与えており，実装可能性という点から貴重な考察が含まれている。そして，2024 年に Dartois [42] により 4 次元攻撃法の効率的な実装結果が報告された。

また，Robert の論文 [105] においては，Petit [101] から始まり de Quehen ら [102] により発展させられた SIDH に対する「ねじれ点攻撃（torsion point attack）」との関連性もまとめられており，さらに広い見地から補助点情報を使った攻撃法の全体像も概観できるようになっている。

[*1] 特殊極値整環 $\mathcal{O}_0$ に関しては 6.1.5.2 節を参照のこと。

6.1.3 レベル構造付き同種写像問題

定義 6.2 の SIDH 同種写像問題では，$P'_{\mathtt{A}} = \varphi_{\mathtt{B}}(P_{\mathtt{A}}), Q'_{\mathtt{A}} = \varphi_{\mathtt{B}}(Q_{\mathtt{A}})$ となる $E[\ell_{\mathtt{A}}^{e_{\mathtt{A}}}]$ の基底 $P_{\mathtt{A}}, Q_{\mathtt{A}}$ および $E_{\mathtt{B}}[\ell_{\mathtt{A}}^{e_{\mathtt{A}}}]$ の基底 $P'_{\mathtt{A}}, Q'_{\mathtt{A}}$ が補助点情報として与えられたことにより多項式時間攻撃を受けた。それを回避するために M-SIDH 鍵共有や FESTA 鍵共有が提案されたが，De Feo ら [56] によって，それらの鍵共有の安全性を保証する問題として，SIDH 同種写像問題を一般化したレベル構造付き同種写像問題（Γ-SIDH 問題）が提案された。

まず，レベル構造を定義する。E が定義されている有限体の標数を p として，N を p と互いに素な正整数とする。$E[N]$ を生成する 2 点 (P, Q) を $E[N]$ の基底と呼び，$E[N]$ の全ての基底からなる集合を $\mathcal{B}_E[N]$ で表す。2×2 の可逆行列全体 $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ の $\mathcal{B}_E[N]$ への作用を

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot (P, Q) = (aP + bQ, cP + dQ)$$

で定める。$\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ の部分群 Γ に対して，$\Gamma \cdot (P, Q) := \{\gamma \cdot (P, Q) \,|\, \gamma \in \Gamma\}$ を Γ-軌道とすると基底集合 $\mathcal{B}_E[N]$ は Γ-軌道によって分割される。Γ-軌道を（レベル N の）Γ-レベル構造と呼び，全ての Γ-レベル構造からなる集合を $\mathcal{B}_E[\Gamma]$ と表す。すなわち，基底集合 $\mathcal{B}_E[N]$ は $\mathcal{B}_E[N] = \bigcup_{S \in \mathcal{B}_E[\Gamma]} S$ と Γ-レベル構造 $S \in \mathcal{B}_E[\Gamma]$ によって交わりなく分割される。

定義 6.3 ((d,Γ)-SIDH 問題，Γ-SIDH 問題 [56]) 楕円曲線 E と d 次の同種写像 ϕ によって d-同種な $E'(= \phi(E))$，Γ-レベル構造 $S \in \mathcal{B}_E[\Gamma]$ に対して，$(E, S, E', \phi(S))$ が与えられて ϕ を計算せよ（(d,Γ)-*SIDH* 問題）。次数 d が文脈から明らかな場合は，単に Γ-*SIDH* 問題と呼ぶ。

多くの暗号応用においては [56, 補題 6] に示されるように Γ を $\mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ の部分群としてよいので，以下でも $\Gamma \subseteq \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ とする。特に，Γ-SIDH 問題は，$\Gamma = \mathrm{I} = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\}$ の時には，定義 6.2 の SIDH 同種写像問題と等しくなり多項式時間解法を有する一方，$\Gamma = \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ の時には，一般の同種写像問題と等しくなり現状では効率的な解法が知られていない。よって，$\mathrm{I} \subsetneq \Gamma \subsetneq \mathrm{SL}_2(\mathbb{Z}/\mathrm{N}\mathbb{Z})$ となる Γ によって暗号構成に有用な Γ-SIDH 問題を見つけることが重要な課題である。

そのような Γ の例として，$\Gamma_{\mathrm{M}} := \left\{ \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right\} (\subsetneq \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z}))$ に対する Γ_{M}-SIDH 問題の困難性が M-SIDH 鍵共有の安全性の根拠となるので，Γ_{M}-SIDH 問題は M-SIDH 問題と呼ばれる [56]。また，FESTA 鍵共有は $\Gamma_{\mathrm{D}} = \left\{ \begin{pmatrix} \lambda & 0 \\ 0 & \lambda^{-1} \end{pmatrix} \right\} (\subsetneq \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z}))$ に対する Γ_{D}-SIDH 問題の困難性を安全性の根拠としており，Γ_{D}-SIDH 問題は，対角-SIDH 問題（Diagonal-SIDH 問題）と呼ばれている [56]。6.2.2 節での参照のため，定義 6.4 で，改めて M-SIDH 問題，対角-SIDH 問題を定義する。

定義 6.4 (M-SIDH 問題，対角-SIDH 問題 [56]) 上で与えた $\mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ の部分群 $\Gamma_{\mathrm{M}}, \Gamma_{\mathrm{D}}$ に対して，Γ_{M}-*SIDH* 問題を *M-SIDH* 問題と定義して，Γ_{D}-*SIDH* 問題を対角-*SIDH* 問題と定義する。

M-SIDH 鍵共有においては, 相異なる小素数 $q_1, \ldots, q_t$ を用いて $N = q_1 \cdots q_t$ とした Γ_{M} を用いることで, $\#\Gamma_{\mathrm{M}} = 2^t$ となることを安全性の根拠にしている。また，FESTA では，N を 2 のべき乗とした Γ_{D} を用いる。それらのパラメータに対して，M-SIDH 問題，対角-SIDH 問題共に，現状，指数関数時間の攻撃法しか知られていない [56, 表 1]。

6.1.4 同種写像に基づく一方向性群作用（暗号学的群作用）に関する計算問題

素体 $\mathbb{F}_p$ 上定義された超特異楕円曲線間の同種写像問題の困難性に基づく鍵共有法として，CSIDH 鍵共有（6.2.1 節参照）が 2018 年になって Castryck らによって提案された [28]。その安全性の根拠となる計算問題を [41, 3] に従ってまとめる。以降では，整数 a, b（$a < b$）に対して $\mathbb{Z}$ の部分集合 $[a, b]$ を $[a, b] := \{a, a+1, \ldots, b-1, b\}$ とする。

6.1.4.1 2 種の一方向性群作用：REGA と EGA

■CSIDH 鍵共有・CSI-FiSh 署名方式の公開パラメータ CSIDH 鍵共有で，公開パラメータは $pp_{\texttt{csidh}} := (\mathfrak{O}, (\mathfrak{l}_1, \mathfrak{l}_2, \ldots, \mathfrak{l}_n), E, B)$ で与えられる。ここで，$\mathfrak{O}$ は虚 2 次代数体の整環，$\mathfrak{l}_1, \mathfrak{l}_2, \ldots, \mathfrak{l}_n$ はノルムが小さい奇素数 ℓ_i になる $\mathfrak{O}$ の素イデアルで，ℓ_i は $(\ell_i) := \mathfrak{l}_i\bar{\mathfrak{l}}_i$（$i = 1, 2, \ldots, n$）と 2 個の異なる素イデアル $\mathfrak{l}_i, \bar{\mathfrak{l}}_i$ の積に分解している。そして $p + 1 = 4 \cdot \ell_1 \cdots \ell_n$ とした時，p は素数である必要がある。小奇素数 ℓ_i は，例えば，$\ell_1 = 3, \ell_2 = 5, \ldots$ である。E は，$\mathbb{F}_p$ 上定義されて，$\mathfrak{O}$ を $\mathbb{F}_p$-自己準同型環にもつ超特異楕円曲線である。B は指数 e_i の絶対値の上限値，すなわち $-B \leq e_i \leq B$ となる指数 e_i を CSIDH 鍵共有では使う。

CSI-FiSh 署名方式の公開パラメータには，イデアル類群 $\mathrm{cl}(\mathfrak{O})$ の構造計算がなされた $pp_{\texttt{csidh}}$ が含まれる。そして，$\mathrm{cl}(\mathfrak{O})$ の生成元の間の関係式の情報 $\mathcal{R}$ も付された公開パラメータ $pp_{\texttt{csi-fish}} := (pp_{\texttt{csidh}}, \mathcal{R})$ を用いることで，有限アーベル群 $\mathrm{cl}(\mathfrak{O})$ 上での効率的な一様サンプリングが可能になる。特にその代表的なパラメータである CSIDH-512 では，$\mathrm{cl}(\mathfrak{O})$ がノルム 3 のイデアル $\mathfrak{l}_1$ により生成される巡回群であることが [15] において示された。現在，イデアル類群 $\mathrm{cl}(\mathfrak{O})$ の構造計算は入力 $\mathfrak{O}$ のサイズに対して（古典）準指数時間必要であるので，CSIDH-512 のように生成元が既知の十分大きいイデアル類群を得ることは一般に困難で，公開パラメータ $pp_{\texttt{csi-fish}}$ 生成に対する大きな制約となっている。

■CSIDH・CSI-FiSh 群作用の抽象形としての REGA・EGA CSIDH 鍵共有，及び CSI-FiSh 署名方式での基本演算は，$\mathbb{F}_p$-自己準同型環に虚 2 次代数体の整環 $\mathfrak{O}$ をもつ楕円曲線集合 X に対する $\mathfrak{O}$ のイデアル類群 $G := \mathrm{cl}(\mathfrak{O})$ の群作用 $(g, x) \mapsto gx \in X$ (但し $g \in G, x \in X$) として理解できる。その群作用は，自由かつ推移的である。群作用の詳細については 6.2.1.1 節を参照のこと。その記法に従えば，CSIDH・CSI-FiSh における同種写像問題は，この群作用の（G に関する）逆関数 $(x, gx) \mapsto g$ を計算する問題と理解できる。このことから，CSIDH・CSI-FiSh における群作用は，一方向性群作用（または暗号学的群作用）と呼ばれる [41, 112]。

但し，CSIDH 鍵共有と CSI-FiSh 署名方式では，$pp_{\texttt{csidh}}$ と $pp_{\texttt{csi-fish}}$ の違いに応じて G からのサンプリング方法が異なる。CSI-FiSh 署名方式ではすでに述べたように G から効率的に一様サンプリングするのに対して，CSIDH では $[-B, B]^n \subset \mathbb{Z}^n$ から適切に選んだ $(e_1, e_2, \ldots, e_n)$ により計算した $\prod_{i=1}^{n} \mathfrak{l}_i^{e_i}$ によって G からのサンプリングを行う。

これらサンプリング方法の違いに基づいて，最近の一方向性群作用の研究 [3, 90] では，CSI-FiSh 署名方式の場合の一方向性群作用を EGA（Effective Group Action：有効群作用）と呼び，CSIDH 鍵共有の場合の群作用を REGA（Restricted Effective Group Action：制限的有効群作用）と呼んでいる。以降では，基本構成としての「CSIDH」を接頭辞に付けて，それぞれの一方向性群作用を CSIDH-EGA，CSIDH-REGA と呼ぶことにする。

6.1.4.2 CSIDH-(R)EGA 上の計算問題

■CSIDH-EGA 上の DL,CDH 計算問題とそれらの量子帰着同値性 CSIDH-EGA に関して離散対数問題（Discrete Logarithm: DL）にあたる基本問題は，以下の CSIDH-EGA-DL 問題であり，更に，それに基づいた CDH（Computational Diffie–Hellmann）問題は，CSIDH-EGA-CDH 問題である。それらの問題は，それぞれ，CSIDH ベクトル化問題及び CSIDH 並列化問題と呼ばれることもある [41, 112, 29]。以下，イデアル類 $[\mathfrak{a}]$ の群作用 $[\mathfrak{a}]E$ に関しては 6.2.1.1 節を参照のこと。

定義 6.5 (CSIDH-EGA-DL 問題 [41, 28, 3]) *CSI-FiSh* 署名公開パラメータ $pp_{\texttt{csi-fish}}$ と，$\mathbb{F}_p$ 上定義されており $\mathbb{F}_p$-自己準同型環 $\mathfrak{O}$ をもつ超特異楕円曲線 $E, E_{\mathtt{A}}$ が与えられた時，$E_{\mathtt{A}} = [\mathfrak{a}]E$ となる $\mathfrak{O}$ のイデアル $\mathfrak{a}$ を計算せよ。但し，$\mathfrak{a}$ の E への作用が効率的に計算可能な場合に限る。例えば，$\mathfrak{a}$ が小さい次数のイデアル積で与えられる場合などである。

定義 6.6 (CSIDH-EGA-CDH 問題 [41, 28, 3]) *CSI-FiSh* 署名公開パラメータ $pp_{\texttt{csi-fish}}$ と，$\mathbb{F}_p$ 上定義されており $\mathbb{F}_p$-自己準同型環 $\mathfrak{O}$ をもつ超特異楕円曲線 $E, E_{\mathtt{A}} := [\mathfrak{a}]E, E_{\mathtt{B}} := [\mathfrak{b}]E$（但し，$\mathfrak{a}, \mathfrak{b}$ は群作用が効率的に計算できる $\mathfrak{O}$ のイデアル）が与えられた時，$[\mathfrak{a}\mathfrak{b}]E = [\mathfrak{b}]E_{\mathtt{A}} = [\mathfrak{a}]E_{\mathtt{B}}$ を計算せよ。

通常の DL 問題と CDH 問題の場合のように，CSIDH-EGA-CDH 問題を CSIDH-EGA-DL 問題に帰着させることができるが，[67] において，その逆，CSIDH-EGA-CDH 問題を解くオラクルを用いて CSIDH-EGA-DL 問題を解く多項式時間量子帰着アルゴリズムが提案されている。[67] の帰着では，CSIDH-EGA-CDH 問題を完全に解くオラクルを仮定していた。Montgomery ら [90, 69] により有意な（non-negligible）確率で CSIDH-EGA-CDH 問題に答えるオラクルを用いても [67] の帰着が成り立つことが示された。

更に，[90] では，CSIDH-REGA 上では [67] の帰着結果が成り立たないことも示されており，判定版の CSIDH-EGA-DDH 問題へ拡張する事に関しても否定的な結果が示されている。また，CSIDH-EGA-DL 問題と CSIDH-EGA-CDH 問題の古典帰着に関して，Castryck ら [29] は，（古典）一方向性の準同型写像が存在するという妥当な仮定の下に，一般には EGA-DL 問題を EGA-CDH 問題に古典帰着させることができないことを簡潔な反例によって示した。

■CSIDH-REGA 上の DL,CDH 計算問題　既に述べたように，現在，イデアル類群 $G := \mathrm{cl}(\mathfrak{O})$ の構造計算を多項式時間で行う（古典）アルゴリズムは知られていないため，G 上の一様分布からの効率的なサンプリング法も知られていない。よって，近似的にその一様サンプリングを行う効率的な（秘密鍵）サンプリング法を用いて CSIDH 鍵共有は与えられる（6.2.1 節参照）。それに従って，CSIDH-EGA-DL 問題と CSIDH-EGA-CDH 問題もそれぞれ修正されて，それらを CSIDH-REGA-DL 問題，CSIDH-REGA-CDH 問題として以下に与える。これらの問題も，前段落と同様に，CSIDH ベクトル化問題及び CSIDH 並列化問題と呼ばれることもある [41, 112, 29]。

定義 6.7 (CSIDH-REGA-DL 問題 [28, 3]) *CSIDH* 鍵共有公開パラメータ $pp_{\texttt{csidh}}$ と，$\mathbb{F}_p$ 上定義されており $\mathbb{F}_p$-自己準同型環 $\mathfrak{O}$ をもつ超特異楕円曲線 E，及び $[-B, B]^n \subset \mathbb{Z}^n$ から一様ランダムに選んだ $(e_1, e_2, \ldots, e_n)$ により $\mathfrak{a} := \prod_{i=1}^{n} \mathfrak{l}_i^{e_i}$ となる $\mathfrak{a}$ によって $E_{\mathtt{A}} = [\mathfrak{a}]E$ となる $E_{\mathtt{A}}$ が与えられた時，$\mathfrak{a}$ と同じイデアル類に属する $\mathfrak{a}'$, *i.e.,* $\mathfrak{a}' \in [\mathfrak{a}]$ を計算せよ。

定義 6.8 (CSIDH-REGA-CDH 問題 [28, 3]) *CSIDH* 鍵共有公開パラメータ $pp_{\texttt{csidh}}$ と，$\mathbb{F}_p$ 上定義されており $\mathbb{F}_p$-自己準同型環 $\mathfrak{O}$ をもつ超特異楕円曲線 $E, E_{\mathtt{A}} := [\mathfrak{a}]E, E_{\mathtt{B}} := [\mathfrak{b}]E$（但し，$\mathfrak{a}, \mathfrak{b}$ は共に，$[-B, B]^n \subset \mathbb{Z}^n$ から一様ランダムに選んだ $(e_1, e_2, \ldots, e_n)$ により $\prod_{i=1}^{n} \mathfrak{l}_i^{e_i}$ と表されるイデアル）が与えられた時，$[\mathfrak{a}\mathfrak{b}]E = [\mathfrak{b}]E_{\mathtt{A}} = [\mathfrak{a}]E_{\mathtt{B}}$ を計算せよ。

■CSIDH-(R)EGA-DL 問題の古典計算機による計算困難性　CSIDH-(R)EGA-DL 問題に関しては，$\mathbb{F}_p$ 上の超特異楕円曲線の同種写像問題に対する Delfs–Galbraith [48] の（古典）アルゴリズムを適用するのが，漸近的解読時間が最速の古典アルゴリズムとされており，その解読時間は $\tilde{O}(\sqrt[4]{p})$ である。

■CSIDH-(R)EGA-DL 問題に対する準指数時間での量子攻撃　G の X への作用が自由かつ推移的であるなら，群作用 DL 問題は，隠れシフト問題に帰着されて，それは更に二面体群に関する隠れ部分群問題（Dihedral Hidden Subgroup Problem: DHSP）に帰着する。DHSP には，準指数時間で動く量子アルゴリズムが知られているので，一般に一方向

性群作用に基づいた暗号方式は，量子計算機に対して準指数時間安全性しかもたない。つまり，Childs ら [36] による通常同種写像問題に対する量子準指数時間アルゴリズムは，CSIDH-(R)EGA-DL 問題に対しても有効であり，CSIDH 群作用に関する DL 問題に対する量子計算機による漸近的な解読時間は準指数関数 $L_p[1/2, \sqrt{3}/2]$ で与えられる。[*2]

実用的には，漸近的ではないその正確な見積もりが，与えられた安全性レベルを達成する p の bit 長を決めるのに重要である。まず，EUROCRYPT 2019 において，Bernstein ら [12] は，CSIDH 群作用を行う量子回路のサイズを具体的に見積もることで，上述の準指数時間アルゴリズムが，従来考えられていたより計算オーバーヘッドが大きいのでないか，つまり，攻撃するのはより困難であろうと主張している。

更に，EUROCRYPT 2020 において，Bonnetain–Schrottenloher [18] と Peikert [100] により独立に 2 つの研究成果が報告された。[18] では，詳細に CSIDH 攻撃量子アルゴリズムを検討して，これまで考えられていたより効率的に攻撃可能であると主張している。それにより，彼らは，Castryck ら [28] が 56 bits 量子安全性レベルと主張していたパラメータが，実際には 38 bits レベルの量子安全性しか確保できないのでないか，という試算を述べている。また，[18] では，Kuperberg の 2005 年の論文 [81] に基づいた攻撃法に関して特に詳細な解析がなされたが，Peikert [100] では，その後の DHSP 解法の進展 [103, 82] も取り入れた安全性評価の改善がなされた。

そして，Chávez-Saab ら [34] の最近の評価結果では，NIST 安全性レベル 1 を満たすために素数 p を 4096 bits または 5120 bits 程度に大きくする必要性が示されており，安全性レベル 2 には 6144 bits，安全性レベル 3 には 8192 bits または 9216 bits 程度の大きさが必要であるという評価結果も報告されている。

■CSIDH・CSI-FiSh パラメータ以外のイデアル類群作用 DDH 問題の古典解法　上で見た EGA 上の DL 問題と CDH 問題の同値性と関係した興味深い DDH（Decisional Diffie–Hellman）問題に関する研究成果が Castryck ら [30, 27, 26] により発表された。ある種のイデアル類群作用においては，虚 2 次整環における「種の理論（genus theory）」を用いて，その作用に関する DDH 問題を効率的に解くことができることが示された。

但し，CSIDH 鍵共有・CSI-FiSh 署名方式に対して重要なこととして，その攻撃が有効になるためにはイデアル類群 $G := \mathrm{cl}(\mathfrak{O})$ が 2-ねじれ点を持つ必要があり，$p \equiv 3 \bmod 4$ である CSIDH パラメータに対しては無効であることも示されている。

6.1.4.3　イデアル類群作用に基づく量子マネーの安全性に関する計算問題

また，現在，イデアル類群の群作用に基づいて公開検証可能な量子マネーを構成する研究も活発に行なわれている [84, 119, 89, 94]。特に，Montgomery–Sharif [94] では，量子マネー（quantum money/quantum lightning）の安全性を示すために，楕円曲線重ね合わせ複製問題（Elliptic Curve Superposition Duplication Problem: ECSDP）や楕円曲線重ね合わせ衝突問題（Elliptic Curve Superposition Collision Problem: ECSCP）といった計算問題が定義されて，その計算困難性が論じられている。

6.1.5　自己準同型環計算問題と SQIsign 署名方式の安全性に関する計算問題

6.1.5.1　自己準同型環計算問題

同種写像暗号は，Kohel [79]，Galbraith [68]，Couveignes [41] らの先駆的研究にその起源をもつが，特に，Kohel は有限体上の楕円曲線の自己準同型環を計算するアルゴリズムを探求しており，そのために楕円曲線の同種写像からなる「同種写像グラフ」の性質を見極めることから始めて，目的とする自己準同型環計算を同種写像グラフ上のアルゴリズム構成に帰着していく。その後，Kohel–Lauter–Petit–Tignol [80] は，この「同種写像計算」と「自己準同型環計算」を並置しながら考察する視点を，「構成的 Deuring 対応」として計算論的観点から捉え直した（表 6.1 参照）。そこでは，四元数環側での ℓ-同種写像道探索問題を解く KLPT アルゴリズムが鍵となるアルゴリズムである（[77, 53] 参照）。

[*2] ここで，$L_p[\alpha, c] := \exp\left((c + o(1))\,(\log p)^{\alpha}(\log\log p)^{1-\alpha}\right)$ とする。

そして，この構成的 Deuring 対応に基づき「同種写像計算」と「自己準同型環計算」の等価性が示されており [50, 51, 118]，現在，自己準同型環計算問題の困難性に基づいた暗号構成の研究が進められている [71, 60, 61]。

■自己準同型環計算問題とその超特異同種写像計算問題との同値性　以下の記述に関しては，例えば [83] を参照する。また，四元数環については Voight の教科書 [116] に詳しい説明がある。有理数体 $\mathbb{Q}$ 上 $\{1, i, j, k\}$ を基底とするベクトル空間でありかつ $a, b \in \mathbb{Q}$ により $i^2 = a, j^2 = b, k = ij = -ji$ という積構造が入った $\mathbb{Q}$ 上の代数（環）を四元数環 $\mathcal{B}$ と呼ぶ。各素点 ν（素数または ∞）における $\mathbb{Q}$ の完備化 $\mathbb{Q}_\nu$ による $\mathcal{B} \otimes \mathbb{Q}_\nu$ が $\nu = p, \infty$ の時にのみ斜体（可除環）になる四元数環 $\mathcal{B} = \mathcal{B}_{p,\infty}$ を扱う。これを，$\mathcal{B}_{p,\infty}$ は p, ∞ の 2 点のみで分岐する四元数環であるといい，$\mathcal{B}_{p,\infty}$ は同型を除いて一意的に決まる。この同じ素数 p を標数とする有限体上の超特異楕円曲線 E の自己準同型写像がなす環 $\mathrm{End}(E)$ は E の自己準同型環と呼ばれて，$\mathrm{End}(E)$ は $\mathcal{B}_{p,\infty}$ の極大整環 $\mathcal{O}$ になっている[*3]。ここで，（四元数環の）整環とは $\mathbb{Z}$ 上階数 4 の加群でありかつ環であるものであり，極大整環とは，そのような整環の中で包含関係に関して極大になっているものを指す。この自己準同型環 $\mathrm{End}(E)$ を計算する以下の問題が基本である。

定義 6.9 (自己準同型環計算問題 [79]) 超特異楕円曲線 E が与えられて，E の自己準同型環 $\mathrm{End}(E)$ を計算せよ。

Eisenträger らの研究 [50, 51] により，超特異同種写像計算問題と（超特異）自己準同型環計算問題の間に多項式時間帰着による計算問題としての同値性が示された。そこではヒューリスティックな仮定が使われていたが，Wesolowski [118] は，一般化されたリーマン予想に基づいて，その同値性に対して厳密な証明を与えた。また，[99, 87] においては，非スカラー自己準同型写像を計算する問題（One Endomorphism Problem）と自己準同型環計算問題の等価性も示されている。

6.1.1 節で，超特異同種写像問題の古典計算機による現在最速の解読時間は $\tilde{O}(\sqrt{p})$ と見積もられていたので，この同値性により，自己準同型環計算問題も同等の計算時間であるが，直接に，自己準同型環計算問題を解く研究も進められており，[51] において，$\tilde{O}(\sqrt{p})$ 時間の自己準同型環計算（古典）アルゴリズムが報告されており，その後 [66] により改良されている。また，神戸ら [76] によって，10 から 30 bits の素数 p に対する自己準同型環計算の実装報告がなされている。

■Deuring 対応　自己準同型環計算問題で与えられる楕円曲線 E から極大整環 $\mathcal{O}$ への対応は，表 6.1 に掲げたように，楕円曲線に関する様々な概念から四元数環に関する概念への対応に拡張される。その詳細に関しては，例えば [83, 第 2 章] を参照していただきたいが，特に基本的な対応としては，同種写像 $\varphi : E \to E_1$ が，極大整環の間の同型 $\mathcal{O} \cong \mathrm{End}(E), \mathcal{O}_1 \cong \mathrm{End}(E_1)$ を通して，左 $\mathcal{O}$-整イデアルかつ右 $\mathcal{O}_1$-整イデアルである I_φ に対応していることである。これにより始点曲線 E を固定すると，同種写像 $\varphi : E \to E_1$ の終点曲線 E_1 が $\mathcal{O}$ のイデアル類と対応することがわかり，超特異 j 不変量（$\in \mathbb{F}_{p^2}$）の集合がイデアル類集合 $\mathrm{cl}(\mathcal{O})$ と一対一に対応していることもわかる。

一般に表 6.1 に示されるように，幾何的な情報から成る楕円曲線側のデータと代数的な情報から成る四元数環側のデータの間に対応関係が存在しており，Deuring 対応と呼ばれる。自己準同型環計算問題（定義 6.9）は Deuring 対応に基づいた問題であり，楕円曲線側の超特異 j 不変量 $j(E)$ から対応する四元数環側の極大整環 $\mathcal{O} = \mathrm{End}(E)$ を計算する問題となっている。そして，この Deuring 対応は，6.2.4 節及び 6.3.1 節での暗号構成を理解する際にも重要な鍵となっている。

6.1.5.2　SQIsign 署名方式の安全性に関する計算問題

次に，SQIsign 署名方式 [60, 33] の安全性を示すために必要な計算問題を述べる。近年進展が著しい SQIsign2D 署名方式の安全性に関しては [7, 96] などを参照のこと。

[*3] 自己準同型写像は英語で endomorphism であるので，その全体を $\mathrm{End}(E)$ で表す。

表 6.1: Deuring 対応

楕円曲線側	四元数環側
超特異 j 不変量 $j(E) \in \mathbb{F}_{p^2}$（の $\mathbb{F}_{p^2}/\mathbb{F}_p$-Galois 共役類）	$\mathcal{B}_{p,\infty}$ 内の極大整環 $\mathcal{O} = \mathrm{End}(E)$ の自己同型類（タイプ）
同種写像 $\varphi : E \to E_1$ で定まる (E_1, φ)	左 $\mathcal{O}$-整イデアルかつ右 $\mathcal{O}_1$-整イデアルである I_φ
自己準同型写像 $\theta \in \mathrm{End}(E)$	主イデアル $\mathcal{O}\theta$
同種写像の次数 $\deg(\varphi)$	イデアルのノルム $n(I_\varphi)$
双対同種写像 $\hat{\varphi}$	共役イデアル $\overline{I_\varphi}$
同じ定義域・値域の同種写像 $\varphi : E \to E_1, \psi : E \to E_1$	同値なイデアル $I_\varphi \sim I_\psi$
超特異 j 不変量 $j(E) \in \mathbb{F}_{p^2}$ の集合	イデアル類の集合 $\mathrm{cl}(\mathcal{O})$
同種写像の合成 $\tau \circ \rho : E \to E_1 \to E_2$	イデアル積 $I_{\tau\circ\rho} = I_\rho \cdot I_\tau$
N-同種写像の同型類	レベル N の Eichler 整環の類集合

■SQIsign 署名方式の健全性に関する計算問題　まずは，SQIsign 署名方式の健全性（偽造不可能性）を示すための計算問題である超特異平滑自己準同型写像計算問題（Smooth Endomorphism Problem: SEP）を定義する。以下では，核が巡回群となる自己準同型写像を巡回自己準同型写像と呼ぶ。

定義 6.10 (超特異平滑自己準同型写像計算問題 [60, 33]) 超特異楕円曲線 E が与えられて，平滑な整数を次数にもつ E 上の（非自明な）巡回自己準同型写像を見つけよ。

この問題で問うているような非自明な自己準同型写像が計算できれば，[51] で見るように，自己準同型環 $\mathrm{End}(E)$ 全体も計算できることが知られているので，この問題は，本質的に自己準同型環計算問題と同値である [60]。よって，$\tilde{O}(\sqrt{p})$ 時間での古典アルゴリズム [51] が現状最速と見積もられる。

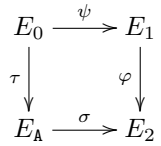
■特殊極値的楕円曲線　次に，SQIsign 署名方式の零知識性を示すための計算問題を述べるが，公開パラメータで重要となる楕円曲線 E_0 を示す。$p = 3 \bmod 4$ の時，j 不変量 $j = 1728$ となる $E_0 : y^2 = x^3 + x$ の $\mathcal{O}_0 = \mathrm{End}(E_0)$ は $i^2 = -1, j^2 = -p$ となる $\mathcal{O}_0 = \mathbb{Z} + \mathbb{Z}i + \mathbb{Z}\frac{i+j}{2} + \mathbb{Z}\frac{1+ij}{2}$ となることが知られている。更に具体的に自己準同型写像 $\iota : (x, y) \mapsto (-x, \sqrt{-1}y), \pi : (x, y) \mapsto (x^p, y^p)$ により $\mathrm{End}(E_0) = \mathbb{Z} + \mathbb{Z}\iota + \mathbb{Z}\frac{\iota+\pi}{2} + \mathbb{Z}\frac{1+\iota\pi}{2}$ で与えられる。

標数 p と ∞ のみで分岐する四元数環 $\mathcal{B}_{p,\infty} := \mathbb{Q}[i, j]$ における極大整環 $\mathcal{O}_0 \subset \mathcal{B}_{p,\infty}$ は，最小判別式の 2 次整環である $\mathfrak{O} \subset \mathcal{O}_0 \cap \mathbb{Q}[i]$ による $\mathfrak{O} + j\mathfrak{O} \subset \mathcal{O}_0$ が部分整環であり $\mathfrak{O} \subset (j\mathfrak{O})^\perp$ と直交分解しているとき[*4]，特殊極値的（special extremal）であるという。詳細は [60, 83] を参照。$p = 7 \bmod 12$ の時，上述の E_0 に対して $\mathrm{End}(E_0)$ は特殊極値的であり，この時，E_0 は特殊極値的曲線と呼ばれる。特殊極値的曲線 E_0 は，その自己準同型環の構造が簡単で計算上扱いやすいため GPS 署名方式 及び SQIsign 署名方式の公開パラメータの一部として必要である。

[*4] $\mathcal{B}_{p,\infty}$ における内積は $\alpha, \beta \in \mathcal{B}_{p,\infty}$ に対して $\frac{1}{2}\mathrm{tr}(\alpha\bar{\beta})$ で与えられて，ここは，その内積に関する直交分解である（$\mathcal{B}_{p,\infty}$ 内のトレース，共役の定義は，例えば [83] を参照のこと）。

■SQIsign 署名方式の零知識性に関する計算問題　SQIsign 署名方式では，右図の同種写像 τ が秘密鍵で，超特異楕円曲線 $E_{\mathtt{A}}$ が公開鍵（の主要な一部）である。署名生成では，同種写像 ψ, φ を適切に生成して得られた合成写像 $\varphi \circ \psi \circ \hat{\tau}$ を「ランダム化」した同種写像 σ を署名とする[*5]。その詳細は 6.3.1.1 節を参照。[60, 61] において定義された E_0 を始点とする同種写像から成るある集合 $\mathcal{P}_{N_\tau}$ を τ によって $E_{\mathtt{A}}$ を始点とした同種写像に移した集合 $[\tau]_*\mathcal{P}_{N_\tau}$（$\mathcal{P}_{N_\tau}$ の τ による pushforward）を考える。正しく生成された署名同種写像 σ は $[\tau]_*\mathcal{P}_{N_\tau}$ に属するのであるが，それが $E_{\mathtt{A}}$ を始点とした 2 べき次数 $D\,(=2^e)$ の巡回同種写像全体 $\mathtt{Iso}_{D,j(E_{\mathtt{A}})}$ から一様ランダムにサンプリングしたのと区別が付くかという問題が以下であり，SQIsign 署名方式の零知識性を示すために必要である。

SQIsign 同種写像図式



CSI-FiSh，GPS 図式と同様に可換図式ではない。

定義 6.11 (SQIsign 署名方式のランダム識別問題 [60, 33]) $\tau : E_0 \to E_{\mathtt{A}}$ を秘密同種写像として，楕円曲線 E_0 を含む *SQIsign* 署名方式の公開パラメータ $pp_{\mathtt{sqisign}}$（詳しくは *6.3.1* 節参照）と公開鍵 $E_{\mathtt{A}}$ が入力として与えられると共に，$[\tau]_*\mathcal{P}_{N_\tau}$ から一様サンプリングして返すオラクル O_τ への多項式回のアクセスが許される時に，$E_{\mathtt{A}}$ を始点とする同種写像 σ が与えられて σ が $\mathtt{Iso}_{D,j(E_{\mathtt{A}})}$ から一様ランダムに選ばれたか，$[\tau]_*\mathcal{P}_{N_\tau}$ から一様ランダムに選ばれたかを判定せよ。

SQIsign 署名方式の提案者によると，現在のところ，SQIsign 署名方式のランダム識別問題を解くのに，E_0 と $E_{\mathtt{A}}$ の情報から τ を暴く攻撃法より効率の良い攻撃法はまだ知られていないとのことである [60, 83]。つまり，$\tilde{O}(\sqrt{p})$ 時間を必要とすると見積もられている。

また，上述の SQIsign 署名方式に関する計算問題は，どちらも補助点を問題に含まないことにより，6.1.2 節で見た最近の SIDH 同種写像問題に対する攻撃法が適用できないことに注意する。

6.2　同種写像に基づく代表的な暗号方式

以下では，6.2.1 節で CSIDH 鍵共有と CSIDH 鍵共有以外の群作用暗号を，6.2.2 節でレベル構造付き同種写像問題に基づく鍵共有方式を，6.2.3 節で暗号学的群作用に基づく署名方式を，6.2.4 節で GPS 署名方式を述べる。

また，ここで述べた方式以外にも POKE [6]，IS-CUBE [91]，LIT-SiGamal [92] など新たな鍵共有方式が提案されていることにも注意する。

6.2.1　暗号学的群作用に基づく鍵共有方式

6.2.1.1　CSIDH 鍵共有

Castryck ら [28] により提案された CSIDH 鍵共有を記述する。CSIDH 鍵共有は，有限アーベル群 G による一方向性群作用をもつ空間 X 上で構成される。ここで，$X = \mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ は，$\mathbb{F}_p$ 上定義されて $\mathbb{F}_p$-有理自己準同型環が固定された虚 2 次整環 $\mathfrak{O}$ と同型であり，かつその同型により p 乗フロベニウス写像が $\pi \in \mathfrak{O}$ に移されるような超特異楕円曲線の $\mathbb{F}_p$-同型類の集合であり，$G = \mathrm{cl}(\mathfrak{O})$ は $\mathfrak{O}$ のイデアル類群である。以下では，$\mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ が空でないと仮定する。Castryck ら [28] は，6.1.4.2 節で定義した CSIDH-REGA-CDH の判定版問題の困難性に基づいて，CSIDH 鍵共有方式を提案した。

K を虚 2 次代数体，$\mathfrak{O} \subset K$ をその整環とする，すなわちランク 2 の自由 $\mathbb{Z}$-加群である K の部分環である。$\mathfrak{O}$-分数イデアルは，$\alpha \in K^*$ と $\mathfrak{O}$-イデアル $\mathfrak{a}$ によって $\alpha\mathfrak{a}$ と表される K 内の $\mathfrak{O}$-部分加群である。$\mathfrak{a}\mathfrak{b} = \mathfrak{O}$ となる $\mathfrak{O}$-分数イデ

[*5] $\hat{\tau}$ は τ の双対同種写像である。表 6.1 も参照のこと。

アル $\mathfrak{b}$ が存在する時に（$\mathfrak{O}$-分数イデアル）$\mathfrak{a}$ は可逆であるという。そして，そのような $\mathfrak{b}$ が存在するならば，$\mathfrak{a}^{-1} = \mathfrak{b}$ と定義する。可逆分数イデアルの集合 $I(\mathfrak{O})$ はイデアル積に関してアーベル群をなす。この群には主イデアルからなる部分群 $P(\mathfrak{O})$ が含まれており，$\mathfrak{O}$ のイデアル類群は商群 $\mathrm{cl}(\mathfrak{O}) = I(\mathfrak{O})/P(\mathfrak{O})$ によって定義される。どのイデアル類 $[\mathfrak{a}] \in \mathrm{cl}(\mathfrak{O})$ にも整イデアルが存在してその代表として使うことができる。$\mathfrak{O}$ のどの整イデアル $\mathfrak{a}$ も $\mathfrak{O}$-イデアルの積として $\mathfrak{a}_s \not\subseteq \pi\mathfrak{O}$ となる整イデアル $\mathfrak{a}_s$ によって $(\pi\mathfrak{O})^r\mathfrak{a}_s$ と表せる。ここで，π は，p 乗フロベニウス写像。この表示により，整イデアル $\mathfrak{a}$ に対して楕円曲線 $E/E[\mathfrak{a}]$ とそこへの $N(\mathfrak{a})$ 次同種写像 $\varphi_{\mathfrak{a}} : E \to E/E[\mathfrak{a}]$ が以下のように定義される。ここで，$N(\mathfrak{a}) := \#(\mathfrak{O}/\mathfrak{a})$ は $\mathfrak{a}$ のノルムである。$\varphi_{\mathfrak{a}}$ の分離的な部分は $E[\mathfrak{a}] = \bigcap_{\alpha \in \mathfrak{a}_s} \ker \alpha$ を核にもつ同種写像であり，純非分離的な部分はフロベニウス写像 π の r 回の繰り返しで与えられる。同種写像 $\varphi_{\mathfrak{a}}$ 及び値域曲線 $E/E[\mathfrak{a}]$ は共に $\mathbb{F}_p$ 上定義されており $\mathbb{F}_p$-同型を除いて一意的に決まる。ここで主イデアルにより定義される同種写像は E 上の自己準同型写像になるので，2 つのイデアルが同じイデアル類に属することと，対応する同種写像が $\mathbb{F}_p$-同型な値域曲線を与えることとは同値である。つまり，$E/E[\mathfrak{a}]$ の $\mathbb{F}_p$-同型類はイデアル類 $[\mathfrak{a}]$ のみにより決まり，特に，この対応はイデアル類群 $\mathrm{cl}(\mathfrak{O})$ の $\mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ への作用を与える。更に，$\mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ に属する 2 つの楕円曲線間の $\mathbb{F}_p$-同種写像 ψ はすべてこの対応により可逆な $\mathfrak{O}$-イデアルから得られる。そして分離部分 $\mathfrak{a}_s$ は ψ から $\mathfrak{a}_s = \{\alpha \in \mathfrak{O} \mid \ker \alpha \supseteq \ker \psi\}$ によって復元できる。その対応は以下の定理にまとめられる。

定理 6.12 ([28]) 虚 2 次代数体内の整環 $\mathfrak{O}$ と $\pi \in \mathfrak{O}$ を $\mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ が空集合でないものとする。その時，以下で与えられるイデアル類群 $\mathrm{cl}(\mathfrak{O})$ の $\mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ への作用は自由かつ推移的である。

$$\begin{array}{ccc} \mathrm{cl}(\mathfrak{O}) \times \mathcal{E}\ell\ell_p(\mathfrak{O}, \pi) & \to & \mathcal{E}\ell\ell_p(\mathfrak{O}, \pi) \\ ([\mathfrak{a}], E) & \mapsto & E/E[\mathfrak{a}], \end{array}$$

ここで，$\mathfrak{a}$ は類 $[\mathfrak{a}]$ を代表する整イデアルである。

以下では $E/E[\mathfrak{a}]$ を $[\mathfrak{a}]E$ と書くことにする。定理 6.12 で述べた群作用に基づいて，以下のように CSIDH 鍵共有プロトコル（図 6.1）を定義する。下の図で，$\mathfrak{a} \leftarrow \mathrm{cl}(\mathfrak{O})$ と書いたのは，実際にはイデアル類群 $\mathrm{cl}(\mathfrak{O})$ からのサンプリングとして，定義 6.7 の CSIDH-REGA-DL 問題及び定義 6.8 の CSIDH-REGA-CDH 問題に記載された REGA としての $\mathfrak{a}$ のサンプリング法を用いる。モンゴメリ型楕円曲線 $E : y^2 = x^3 + ax^2 + x$ に対して，係数 a は，E のモンゴメリ係数と呼ばれる。CSIDH 鍵共有では，始点曲線 $E : y^2 = x^3 + x$ に対して，アリスとボブによって計算される楕円曲線はすべてモンゴメリ型楕円曲線である。

アリス		**ボブ**
$\mathfrak{a} \leftarrow \mathrm{cl}(\mathfrak{O})$: アリスの秘密鍵		$\mathfrak{b} \leftarrow \mathrm{cl}(\mathfrak{O})$: ボブの秘密鍵
$[\mathfrak{a}]E$（のモンゴメリ係数）を計算	$\xrightarrow{[\mathfrak{a}]E}$	$[\mathfrak{b}]E$（のモンゴメリ係数）を計算
$SK_{\mathtt{Alice}} := [\mathfrak{a}]([\mathfrak{b}]E)$ のモンゴメリ係数	$\xleftarrow{[\mathfrak{b}]E}$	$SK_{\mathtt{Bob}} := [\mathfrak{b}]([\mathfrak{a}]E)$ のモンゴメリ係数

図 6.1: CSIDH 鍵共有の概要

イデアル類群 $\mathrm{cl}(\mathfrak{O})$ は可換なので，$[\mathfrak{a}]([\mathfrak{b}]E) = [\mathfrak{a}\mathfrak{b}]E = [\mathfrak{b}\mathfrak{a}]E = [\mathfrak{b}]([\mathfrak{a}]E)$ であり，そのモンゴメリ係数を考えれば $SK_{\mathtt{Alice}} = SK_{\mathtt{Bob}}$ となるので，アリスとボブは同じ鍵を共有できる。その計算アルゴリズムについては，Castryck ら [28]，Meyer ら [88]，Onuki ら [97] などを参照のこと。CSIDH 鍵共有の安全性は，CSIDH-REGA-CDH 問題の困難性に基づく。

最近, 主に CSIDH 系の方式で使う演算を効率化するべき根同種写像（radical isogeny）計算法 [25, 23] や square-root Vélu 計算法 [11]，さらにそれらを組み合わせた計算法 [46] が提案されている。また，Chávez-Saab らによって，効率化された SQALE 鍵共有方式 [34] も提案されており，4096 bits 以上の CSIDH 素数パラメータに関する SQALE 鍵共有の実装報告もなされている。2024 年に出版された CSIDH 実装報告 [20] も参照のこと。

6.2.1.2 群作用に基づく CSIDH 以外の鍵共有方式

CSIDH 鍵共有の変種には，OSIDH 鍵共有 [37, 43] や SiGamal 暗号方式 [93]・Sims 暗号方式 [65] がある。OSIDH 鍵共有では，楕円曲線以外に「向き」（orientation）と呼ばれる付加情報への群作用も考慮しており，SiGamal 暗号方式では，楕円曲線とその上のねじれ点への群作用を考慮した暗号方式の設計になっている。特に，SiGamal 暗号方式では，ねじれ点への群作用を取り入れることで，CSIDH 共有鍵であるモンゴメリ係数（図 6.1 参照）ではなく，ねじれ点の離散対数からなる一様ランダムな乱数を送受信者間で共有できるようになっている。

2023 年に入ってから，CSIDH 鍵共有以外で向き付けられた楕円曲線に対して新たに SCALLOP [55] と呼ばれる群作用暗号が De Feo らによって提案された。SCALLOP により，利用できる「向き」の取り方を増やすことができて，EGA パラメータの選択肢を増やすことが可能となる。その後，SCALLOP を高次元同種写像を用いて効率化した SCALLOP-HD [35] や，実用的なパラメータ選択のための改良を施した PEARL-SCALLOP [4] が提案された。PEARL-SCALLOP により，CSIDH-512，CSIDH-1024，CSIDH-1536 と同等の安全性レベルの EGA が得られて，それらのパラメータに関して実際に格段の計算効率化が実現できたことが [4] で報告されている。

また，レベル構造付きの群作用を考察した論文 [70, 5] や，従来の群作用をランク 1 加群作用と位置付けてランク 2 加群作用に拡張した論文 [107] など，最近になっても引き続いて，暗号学的群作用の新しい構成法が提案されていることにも注意する。

6.2.2 レベル構造付き同種写像問題に基づく鍵共有

定義 6.3，6.4 で定めたレベル構造付き同種写像問題の困難性に安全性の根拠を置いた鍵共有を述べる。

6.2.2.1 M-SIDH 鍵共有と MD-SIDH 鍵共有

6.1.2 節で見たように，SIDH 鍵共有に対して多項式時間の攻撃法が発見されたが，その直後に，それら攻撃法を回避する SIDH 鍵共有の変種方式が 2 つ提案された。1 つ目は，Moriya による「同種写像次数」を隠すことによる次数隠蔽型（masked-degree）SIDH であり MD-SIDH 鍵共有と呼ばれ，2 つ目は，Fouotsa によるねじれ点隠蔽型（masked torsion point images）SIDH であり M-SIDH 鍵共有と呼ばれる [64]。EUROCRYPT 2023 で発表された [64] において，M-SIDH 方式の方が，MD-SIDH 方式より小さい素数 p によって同等の安全性が得られることが述べられている。M-SIDH 鍵共有の安全性は定義 6.4 の M-SIDH 問題の困難性に基づく。一方，これらの方式は，SIDH 方式より効率面では格段に劣ることも指摘されている。例えば，NIST レベル 1,3,5 に対応する M-SIDH 素数 p の bit 長は，それぞれ 5911，9382，13000 bits となることが示されており，SIKE 暗号方式に提案された素数 bit 長（434，610，751 bits）と比べて格段に大きく，それにより計算効率も劣ることになる。また，Castryck–Vercauteren[31] は特殊なパラメータを用いた M-SIDH に対する攻撃法を発表したが，[64] に述べられているパラメータ設定においては有効な攻撃にはなっていない。

6.2.2.2 (Q)FESTA 鍵共有と binSIDH 鍵共有（terSIDH 鍵共有）

M-SIDH 鍵共有以外にも，ASIACRYPT 2023 において，レベル構造付き同種写像問題困難性に基づいた 2 方式が発表された。Basso–Maino–Pope による FESTA 鍵共有 [9] と Basso–Fouotsa による binSIDH 鍵共有（及び terSIDH 鍵共有）[8] である。それらの FESTA 鍵共有，binSIDH 鍵共有共に，対角-SIDH 問題（6.4）の困難性に安全性の根拠を置いている [56, 図 1]。

CRYPTO 2024 において，Nakagawa–Onuki により，2 次元同種写像の使い方を改良して，FESTA の効率を高めた QFESTA[95] が提案された。[95] で NIST 安全性レベル 1 パラメータである QFESTA-128 では，公開鍵サイズは 247 Bytes で SIKEp434 と同程度，暗号文サイズは 494 Bytes で SIKEp434 の約 2 倍となっている。そして，NIST

安全性レベル 3,5 の QFESTA-192，QFESTA-256 に関しても対応する SIKE パラメータ SIKEp610，SIKEp751 と同様のデータサイズ比率（公開鍵サイズは同程度で，暗号文サイズは約 2 倍）をほぼ達成しており，現状，効率面で最も良い同種写像ベース鍵共有法を実現している。

また，[31] では，M-SIDH の時と同様に特殊なパラメータの FESTA に対する攻撃法が発表されたが [9] のパラメータ設定においては有効な攻撃にはなっていない。

6.2.3 暗号学的群作用に基づく署名方式

本節では，暗号学的群作用に基づく署名方式として，6.2.3.1 節で SeaSign 署名方式を，6.2.3.2 節で CSI-Fish 署名方式を述べる。また，6.1.4.3 節でも述べたように，群作用暗号に基づく署名・認証系の研究として，最近，量子マネー構成への応用が活発に行なわれている [84, 119, 89, 94]。

6.2.3.1 SeaSign 署名方式

De Feo と Galbraith [57] により，CSIDH ベースの SeaSign 署名方式が提案された。これは，CSIDH 鍵共有の数学的な構造を利用したものであり，まだ実用的とは言い難いが，現実的な計算時間に収まる署名方式となっている。以下では，[57] で，「基本形」と呼ばれる SeaSign 署名方式を記載する。[57] では，更に，基本形でも使われたパラメータ t と共に，パラメータ s を導入して，署名が短い方式や公開鍵が短い方式といった変形方式を定義している。また，後続研究 [47] において，実用化を目指して SeaSign 署名方式の高速化が図られた。

以下では，ベクトル $\mathbf{e}$ の各成分は e_i（$i = 1, 2, \ldots, n$）とする，即ち，$\mathbf{e} = (e_1, e_2, \ldots, e_n)$ である。ベクトル $\mathbf{f}_k, \mathbf{z}_k$ についても同様の記法を用いる。H は t bit 出力のハッシュ関数とする。また，整数 a, b（$a < b$）に対して $[a, b] := \{a, a+1, \ldots, b-1, b\} \subset \mathbb{Z}$ とする。

鍵生成： 公開パラメータ $pp_{\mathtt{csidh}} := (\mathfrak{O}, (\mathfrak{l}_1, \mathfrak{l}_2, \ldots, \mathfrak{l}_n), E, B)$，すなわち，虚 2 次整環 $\mathfrak{O}$，イデアル $\mathfrak{l}_1, \mathfrak{l}_2, \ldots, \mathfrak{l}_n$，$\mathbb{F}_p$-有理な超特異楕円曲線 E，上限値 B を入力とする。係数ベクトル $\mathbf{e} \leftarrow_R [-B, B]^n$ を生成する。$E_{\mathtt{A}} := \left[\prod_{i=1}^{n} \mathfrak{l}_i^{e_i}\right] E$ を計算して，秘密鍵 $sk := \mathbf{e}$，公開鍵 $pk := E_{\mathtt{A}}$ とする。

署名生成： 公開パラメータ $pp_{\mathtt{csidh}}$, メッセージ $\mathtt{msg}$, 公開鍵 $pk := E_{\mathtt{A}}$, 秘密鍵 $sk := \mathbf{e}$ を入力とする。各 $k = 1, 2, \ldots, t$ に対して，係数ベクトル $\mathbf{f}_k \leftarrow_R [-(nt+1)B, (nt+1)B]^n$ を生成して，$\mathcal{E}_k := \left[\prod_{i=1}^{n} \mathfrak{l}_i^{f_{k,i}}\right] E$ を計算する。ハッシュ値を $b_1 \| \cdots \| b_t := H(j(\mathcal{E}_1), \ldots, j(\mathcal{E}_t), \mathtt{msg})$ と bit 分解する。各 $k = 1, 2, \ldots, t$ に対して，もし $b_k = 0$ であれば，$\mathbf{z}_k := \mathbf{f}_k$ として，もし $b_k = 1$ であれば，$\mathbf{z}_k := \mathbf{f}_k - \mathbf{e}$ として，もし $\mathbf{z}_k \notin [-ntB, ntB]^n$ であれば $\perp$ を出力する。そうでなければ，署名 $\sigma := (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t, b_1, b_2, \ldots, b_t)$ を出力する。

署名検証 公開パラメータ $pp_{\mathtt{csidh}}$，メッセージ $\mathtt{msg}$，公開鍵 $pk := E_{\mathtt{A}}$，署名 σ を入力とする。まず，署名 σ が，$\sigma := (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t, b_1, b_2, \ldots, b_t)$ というデータになっていることを確認する。各 $k = 1, 2, \ldots, t$ に対して，もし $b_k = 0$ であれば，$\mathcal{E}_k := \left[\prod_{i=1}^{n} \mathfrak{l}_i^{z_{k,i}}\right] E$ を計算して，もし $b_k = 1$ であれば，$\mathcal{E}_k := \left[\prod_{i=1}^{n} \mathfrak{l}_i^{z_{k,i}}\right] E_{\mathtt{A}}$ を計算する。ハッシュ値を $b'_1 \| \cdots \| b'_t := H(j(\mathcal{E}_1), \ldots, j(\mathcal{E}_t), \mathtt{msg})$ と bit 分解する。もし，$(b'_1, b'_2, \ldots, b'_t) = (b_1, b_2, \ldots, b_t)$ であれば，受理を出力して，そうでなければ，棄却とする。

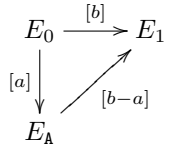
SeaSign 署名方式は，ランダムオラクルモデルにおいて，CSIDH-REGA-CDH 問題困難性の仮定の下で，選択文書攻撃に対して存在的偽造不可（EUF-CMA）安全であることが示されている [57]。

6.2.3.2 CSI-FiSh 署名方式

Beullens–Kleinjung–Vercauteren [15] は，CSIDH-512 パラメータに関するイデアル類群 $\mathrm{cl}(\mathfrak{O})$ の構造計算を遂行することで，効率を高めた CSIDH ベースの CSI-FiSh 署名方式を提案した。CSIDH-512 パラメータでは，512 bits 素数 $p = 4 \cdot \ell_1 \cdot \cdots \cdot \ell_{74} - 1$，$\ell_{74} = 587$ を用いており，Beullens ら [15] は，$\mathbb{Q}(\sqrt{-p})$ のイデアル類群 $\mathrm{cl}(\mathfrak{O})$ が，ノルム 3 のイデアル $\mathfrak{l}_1$ により生成される位数 $N := \#\mathrm{cl}(\mathfrak{O}) = 37 \cdot 1407181 \cdot 51593604295295867744293584889 \cdot 315994145046819958530082787455878322049 09$ の有限巡回群になることを示した。つまり，$[a] \in \mathbb{Z}/N\mathbb{Z}$ に対してイデアル類 $[\mathfrak{l}_1]^a = [\mathfrak{l}_1^a]$ を対応させることで同型 $\mathbb{Z}/N\mathbb{Z} \cong \mathrm{cl}(\mathfrak{O})$ が得られる。更に，イデアル類群の作用 $[\mathfrak{l}_1^a]E$ を同型 $\mathbb{Z}/N\mathbb{Z} \cong \mathrm{cl}(\mathfrak{O})$ を使って $[a]E$ と表し，$\mathbb{Z}/N\mathbb{Z}$ の作用と見なすことにする。これにより 6.1.4.1 節に述べた，より望ましい CSIDH ベースの一方向性群作用 EGA が得られた。

CSI-FiSh 署名方式の構成法は SeaSign 署名方式と変わらないので，Σ-プロトコルにおける記述の差異のみを右図に従って以下に述べる： 位数 N の巡回群 $\mathbb{Z}/N\mathbb{Z}\,(\cong \mathrm{cl}(\mathfrak{O}))$ の作用に基づき，秘密鍵（証拠）を乱数 $a \in \mathbb{Z}/N\mathbb{Z}$，公開鍵 $E_0, E_{\mathtt{A}} := [a]E_0$ とする。証明者は乱数 $b \in \mathbb{Z}/N\mathbb{Z}$ によりコミットメント $E_1 := [b]E_0$ を計算して検証者に送る。検証者はチャレンジ $c \in \{0,1\}$ をランダムに選び証明者に送付，証明者はレスポンス $r := b - ca \bmod N$ を検証者に送る。最後に，検証者は $c = 0$ であれば $E_1 = [r]E_0$ であること，$c = 1$ であれば $E_1 = [r]E_{\mathtt{A}}$ であることが成り立つかどうか検証して検証結果を出力する。

CSI-FiSh 同種写像図式



実際に，CSI-FiSh 署名方式にするには，Fiat–Shamir 変換に則り，チャレンジを $(c_i \in \{0,1\})_{i \in [t]}$ と t 個にするとともに，それらをハッシュ関数を用いて計算して非対話化することで得られる。

上に示したフレームワークをより広範囲の CSIDH パラメータに拡張できれば望ましいが，6.1.4.2 節に述べたように，CSIDH 問題の最近の安全性検討状況によると，NIST 安全性レベル 1 を満たすためには，素数 p を 4096 bits または 5120 bits 程度に大きくする必要性が示されている。そして，それに対応するイデアル類群計算を遂行するのは現状では困難と思われており，このことが，実適用における CSI-FiSh 署名アプローチの限界を示していた。しかし，2023 年に入って，6.2.1.2 節で述べたように，新たな群作用として，SCALLOP [55] が提案されて，EGA パラメータの選択肢を増やすことができるようになり，CSI-FiSh 署名方式の柔軟なパラメータ選択への新しい道が開かれた。その後，実用性を高めるために SCALLOP-HD [35] や PEARL-SCALLOP [4] が提案されたことは 6.2.1.2 節で述べた通りであり，今後の研究に注目する必要がある。

また，Boneh ら [17] により，（ジェネリック）群作用に基づく署名サイズがセキュリティパラメータ λ に対して $\Omega(\lambda^2/\log\lambda)$ となることが示されていることにも注意する。

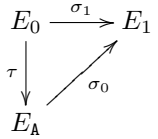
CSI-FiSh 署名提案後に，El Kaafarani ら [52] により，タイト安全な Lossy CSI-FiSh 署名方式の提案もなされた。そして，リング署名・グループ署名など高機能暗号系への拡張研究 [14, 13] があるのが CSI-FiSh 署名方式の利点の一つとなっている。

6.2.4 GPS 署名方式

Galbraith–Petit–Silva（GPS）[71] によって始めて自己準同型環の知識証明に基づく署名方式が提案された。GPS 署名方式は 1 bit チャレンジ空間の零知識証明プロトコルに基づいているため実際に利用するのは困難であろうと思われているが，現在，GPS 署名方式は，SQIsign 署名方式の原型を与えているという点で重要である。6.1.5 節で述べた Deuring 対応と KLPT アルゴリズム [80] が GPS 署名方式の理論的基礎を与える。

右図において E_0 は 6.1.5.2 節で与えた $j(E_0) = 1728$ なる楕円曲線（特殊極値的楕円曲線）であり，そこで見たようにその E_0 に関しては $\mathrm{End}(E_0)$ の構造が簡明な形で与えられている。その楕円曲線 E_0 からの秘密鍵同種写像 $\tau : E_0 \to E_{\mathtt{A}}$ を知っている証明者（署名生成者）は，$E_{\mathtt{A}}$ から別の楕円曲線 E_1 への同種写像 $\sigma_0 : E_{\mathtt{A}} \to E_1$ と τ との合成 $\sigma_0 \circ \tau : E_0 \to E_1$ を KLPT アルゴリズムに基づいて「ランダム化」して同じ始点 E_0 と終点 E_1 をもつ $\sigma_0 \circ \tau$ とは異なる同種写像 $\sigma_1 : E_0 \to E_1$ を得ることができる。

GPS 同種写像図式



さらに，自己準同型環 $\mathrm{End}(E_{\mathtt{A}})$ を計算する問題の困難性に基づけば，このようなランダム化ができるのは，τ を知っている証明者に限られるので，チャレンジ bit $c \in \{0, 1\}$ を送って証明者に同種写像 σ_c を答えさせることにより，τ に関する知識の有無を検査することができて，認証・署名方式が構成できる。それが GPS 認証方式，そしてその Fiat–Shamir 変換署名が GPS 署名方式である。ここでは [71, 第 4 章] と [83, 5.1.2 節] に基づいて GPS 署名方式を記述する。また，[71, 第 4 章] では，通常の Fiat–Shamir 変換を施した署名方式と Unruh 変換を施した署名方式の 2 方式が記述されているが，ここでは記述の簡便さを考慮して前者の記述を基にして以下に署名方式を与える。

鍵生成： 既知の特殊極値的自己準同型環 $\mathcal{O}_0$ をもつ超特異楕円曲線 E_0 とする。互いに素な B-べき平滑数 S_1, S_2[*6]を，S_1, S_2 次の同種写像グラフ上ランダムウォークがグラフのエクスパンダー性により一様分布を導く程度に十分大きくとる。セキュリティパラメータ λ に対して $t := \lambda$（または $t := 2\lambda$）として，t bits 出力のハッシュ関数 H を選ぶ。$pp_{\mathtt{gps}} := (E_0, S_1, S_2, H)$ を公開パラメータとする。さらに，E_0 を始点とする S_1 次のランダムな同種写像 $\tau : E_0 \to E_{\mathtt{A}}$ を計算して，$pp_{\mathtt{gps}}$ と $E_{\mathtt{A}}$ を公開鍵として，τ を秘密鍵とする。

署名生成： 各 $i = 1, \ldots, t$ に関して $E_{\mathtt{A}}$ を始点とする S_2 次のランダムな同種写像 $\sigma_{0,i} : E_{\mathtt{A}} \to E_{1,i}$ を計算する。署名対象メッセージ $\mathtt{msg}$ に対してチャレンジ bit 列 $h := b_1 \| \cdots \| b_t := H(j(E_{1,1}), \ldots, j(E_{1,t}), \mathtt{msg}) \in \{0, 1\}^t$ をハッシュ関数 H で計算する。各 $i = 1, \ldots, t$ に対して，もし $b_i = 1$ なら KLPT アルゴリズムに基づいて「ランダム化」したランダム同種写像 $\sigma_{1,i} : E_0 \to E_{1,i}$ を計算する。署名を $\sigma := (h, \sigma_{b_1,1}, \ldots, \sigma_{b_t,t})$ とする。

署名検証： 公開鍵 $(pp_{\mathtt{gps}}, E_{\mathtt{A}})$，メッセージ $\mathtt{msg}$ と署名 $\sigma = (h, \sigma_1, \ldots, \sigma_t)$ を入力として，各 $i = 1, \ldots, t$ に対して，同種写像 σ_i を計算して，その終点曲線 $E_{1,i}$ を得る。次に $H(j(E_{1,1}), \ldots, j(E_{1,t}), \mathtt{msg})$ を計算して署名内の h と一致するかどうか検証して，全ての $i = 1, \ldots, t$ に対して検証が成功すれば受理を出力して，そうでなければ，棄却とする。

GPS 署名方式は，超特異楕円曲線同種写像計算問題またはそれと同値な自己準同型環計算問題（定義 6.9）の困難性を仮定すればランダムオラクルモデルの下で EUF-CMA 安全であることが示されている [71, 定理 10]。GPS 署名方式では，1 bit のチャレンジを用いた Σ-プロトコルに基づいているため，署名サイズが大きくなるのが欠点である。また，署名生成で使われた KLPT アルゴリズムの計算時間改善も課題であった [83, 5.1.2 節]。以上，GPS 署名方式には (1) 署名サイズ 及び (2) KLPT アルゴリズム計算時間 に関する 2 つの課題が存在する。

6.3 同種写像に基づく主要な暗号方式

本節では，公開鍵と署名サイズが小さいことを特長にもつ SQIsign 署名方式について述べる（表 6.2 参照）。

[*6] S_k（$k = 1, 2$）が B-べき平滑数（powersmooth number）とは，S_k が $\ell_{k,i}^{e_{k,i}} < B$ なる $\ell_{k,i}^{e_{k,i}}$ の積で表される（i.e., $S_k = \prod_i \ell_{k,i}^{e_{k,i}}$）ことである。ただし $\ell_{k,i}$ は互いに異なるものとする。

表 6.2: 同種写像に基づく暗号の分類

文献	暗号化	鍵交換	署名
SQIsign [60, 33, 7]			○

6.3.1 SQIsign 署名方式

以下，自己準同型環計算問題（定義 6.9）の困難性に安全性の根拠を置く SQIsign 署名方式を概説する。SQIsign 署名方式は公開鍵と署名を合わせたサイズが小さい方式として注目されている。また，2024 年 10 月に，SQIsign 署名方式が NIST PQC 標準化プロジェクト追加署名第 2 ラウンドに進むことが発表された [2]。以下では，KLPT アルゴリズムに基づいた SQIsign 署名方式 [60, 33] のアルゴリズムとパラメータを述べた後，最新の改良版である SQIsign2D 署名方式 [7] について報告する。

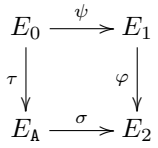
6.3.1.1 KLPT アルゴリズムに基づく SQIsign 署名方式

6.2.4 節で述べた GPS 署名方式を基にして改良を加えた署名方式が SQIsign 署名方式であり，ASIACRYPT 2020 で De Feo–Kohel–Leroux–Petit–Wesolowski [60] により提案された。6.2.4 節末尾に付した GPS 署名方式の 2 つの課題を克服している。チャレンジ空間に同種写像の空間を用いることで，そのサイズをセキュリティパラメータ λ まで大きくして，Σ-プロトコルを 1 度適用するだけで十分な Fiat–Shamir 署名構成とした。これで署名サイズが格段に小さくなった。また，GPS 署名生成においては，表 6.1 の Deuring 対応に基づいて，同種写像のイデアル表現（表 6.1 の四元数環側）をねじれ点を使った表現（表 6.1 の楕円曲線側）に変換する部分で時間が費やされていたが，SQIsign 署名方式ではその処理を速度改善したサブルーチン（IdealToIsogeny）に置き換えるのに成功して現実的な演算効率を達成した（詳細は [60, 61] を参照）。

また，安全性に関しては，健全性は超特異平滑自己準同型写像計算問題（定義 6.10）の困難性に基づき，零知識性は定義 6.11 で述べた SQIsign 署名方式のランダム識別問題の困難性に基づいている。初期提案 [60] では，ノルム方程式を解くサブルーチンに不備があり，生成される署名同種写像 σ に偏りが生じていたことが [61] において指摘された。そして，更に [61] でその不備を除去したアルゴリズム提案が行われた。

■SQIsign 署名アルゴリズム　SQIsign 署名方式では，右図の同種写像 τ が秘密鍵で，超特異楕円曲線 $E_{\mathtt{A}}$ が公開鍵（の主要な一部）である。署名生成では，コミットメント同種写像 ψ とチャレンジ同種写像 φ を適切に生成して得られた合成写像 $\varphi \circ \psi \circ \hat{\tau}$ を一般化された KLPT アルゴリズムに基づいてランダム化した同種写像 σ を署名（Σ-プロトコルのレスポンス）とする。一般化 KLPT アルゴリズムに関しては [33, 2.5.2.2 節] を参照。チャレンジ φ によりセキュリティパラメータ分のランダムネスを与えることができるので，1 度の Σ-プロトコル適用で十分な安全性が達成できる。よって，GPS 署名方式と比べて格段に短い署名サイズが実現できる。

SQIsign 同種写像図式



鍵生成： 既知の特殊極値的自己準同型環 $\mathcal{O}_0$ をもつ超特異楕円曲線 E_0，λ bits の平滑奇数 D_c（λ はセキュリティパラメータ），超特異 2-同種写像グラフの直径より大きな e による $D := 2^e$ を生成して，$pp_{\mathtt{sqisign}} := (E_0, D_c, D)$ を公開パラメータとする。さらに，E_0 を始点とするランダムな同種写像 $\tau : E_0 \to E_{\mathtt{A}}$ を計算して，$pp_{\mathtt{sqisign}}$ と $E_{\mathtt{A}}$ を公開鍵として，τ を秘密鍵とする。

署名生成： E_0 を始点とするランダムな同種写像 $\psi : E_0 \to E_1$ を計算。署名対象メッセージ $\mathtt{msg}$ に対してハッシュ関数 H で計算した $H(j(E_1), \mathtt{msg})$ から決まる D_c 次の巡回同種写像 $\varphi : E_1 \to E_2$ を計算。同種写像の合成 $\varphi \circ \psi \circ \hat{\tau} : E_{\mathtt{A}} \to E_2$ から（一般化された KLPT アルゴリズムを用いて）同じ始点・終点を有して $\hat{\varphi} \circ \sigma$ が巡回同種写像になる D 次のランダム同種写像 $\sigma : E_{\mathtt{A}} \to E_2$ を計算。(E_1, E_2, σ) を $\mathtt{msg}$ の署名として出力。

署名検証： 公開鍵 $(pp_{\mathtt{sqisign}}, E_{\mathtt{A}})$，メッセージ $\mathtt{msg}$ と署名 (E_1, E_2, σ) を入力として，E_1 から E_2 への同種写像 $\varphi := H(j(E_1), \mathtt{msg})$ を計算する。σ が $E_{\mathtt{A}}$ から E_2 への D 次同種写像であることと $\hat{\varphi} \circ \sigma$ が $E_{\mathtt{A}}$ から E_1 への巡回同種写像であることを検証して，共に成立すれば受理を出力して，そうでなければ，棄却とする。

既に述べたように，SQIsign 署名方式の安全性は，超特異平滑自己準同型写像計算問題（定義 6.10）の困難性と，定義 6.11 で述べた SQIsign 署名 σ のランダム識別問題の困難性に基づいている。また，Santos–Eriksen–Meyer–Reijnders [111] は有限拡大体を活用して署名検証を高速に行う方法を提案している。

■SQIsign 署名パラメータ　署名同種写像 σ の次数は $D = 2^e$，チャレンジ同種写像 φ の次数は平滑奇数 D_c である。$\mathbb{F}_p$ 上の超特異楕円曲線 E の位数 $p+1$ のねじれ点及びそのツイスト曲線上の位数 $p-1$ のねじれ点を利用して次数 D, D_c の同種写像を小さい拡大次数の有限体で効率的に計算するために，できるだけ大きい正整数 f，正奇数 T に関して $2^f \cdot T \,|\, p^2 - 1$ が満たされる素数 p（SQIsign 素数）を生成することが必要である。具体的には，ある B に対して B-平滑な T，$T \approx p^{5/4+\epsilon}$（[19] では例えば $0.02 < \epsilon < 0.1$ とする）に対して $2^f \cdot T \,|\, p^2 - 1$ となる素数 p を探索する必要がある。SQIsign 素数の選択基準として，署名検証の効率化には f をできるだけ大きくして，署名生成の効率性にとっては $\sqrt{B}/f$ をできるだけ小さくするのが望ましい [61]。

6.3.1.2　SQIsign2D 署名方式

Dartois ら [44] により高次元同種写像を用いて改善を図った SQIsignHD 署名方式が提案された。さらに 2 次元同種写像によってデータサイズ，演算時間，安全性に関して改善された複数の方式が相次いで発表されている [7, 96, 49, 21]。以下では，それらの中で，特に，SQIsign2D-West 署名方式 [7] に関して，[7] で述べられたパラメータ，データサイズ及び性能報告に関して述べる。SQIsign 署名方式を 2 次元同種写像を用いて改善することができたのは Nakagawa–Onuki [95, 96] の貢献が大きい。

特筆すべきは，素数 p の選択である。上に述べたように，従来の SQIsign 署名方式では B-平滑な T を適切に設定する必要があるなど素数 p の選択には限界が伴っていた。しかし，SQIsign2D-West では，できるだけ小さな c により $p + 1 = c \times 2^e$ となる素数 p を用いるため，セキュリティレベルに応じて柔軟なパラメータ選択がしやすい。また，一般化メルセンヌ素数 $p = c \times 2^e - 1$ を用いることで高速実装も可能になり，表 6.3 に示すように，鍵生成・署名生成・署名検証において実用的な実行時間が達成できることが報告されている [7]。

そして，表 6.3 に示されているように，セキュリティパラメータ λ ($\sim \frac{1}{2}\log_2 p$) に対して公開鍵サイズを $4\lambda + 16$ bits，署名サイズを $9\lambda + 16 + 2\log_2(2\lambda)$ bits と小さく抑えることができるのも特長である。

6.4　同種写像に基づく暗号技術に関するまとめ

本章では，同種写像に基づいた暗号技術をまとめた。NIST PQC 標準化プロジェクト追加署名第 2 ラウンドに進んだ SQIsign 署名方式，CSIDH 鍵共有に代表される群作用暗号，ここ数年進展著しいレベル構造付き同種写像問題に基づく鍵共有方式などに関して方式記述と安全性研究についてまとめてきた。また，SIDH 攻撃に端を発した高次元同種写像の暗号応用に関しても調査結果を報告した。

[41] によると，Couveignes は，1997 年の École Normale Supérieure でのセミナーで既に同種写像に基づく暗号技術を提案しており，ほぼ同時期に Kohel [79] や Galbraith [68] も，同種写像問題に関する研究を始めていた。つまり，同種写像暗号技術の研究は既に 27 年の歴史をもつ。そして，最近になり，耐量子計算機暗号の必要性が高まること

表 6.3: SQIsign2D-West 素数パラメータ p 及び公開鍵・署名サイズ（Bytes），Intel Xeon Gold 6338（Ice Lake, 2GHz）上での鍵生成・署名生成・署名検証の実行時間（ms）[7]

NIST 安全性レベル	1	3	5
素数 p	$5 \cdot 2^{248} - 1$	$65 \cdot 2^{376} - 1$	$27 \cdot 2^{500} - 1$
公開鍵サイズ	66	98	130
署名サイズ	148	222	294
鍵生成	30	85	180
署名生成	80	230	470
署名検証	4.5	14.5	31

で，同種写像暗号技術は注目されて研究が進み，NIST PQC 標準化プロジェクト第 4 ラウンドにも選ばれた SIKE 暗号方式及びその基本形である SIDH 鍵共有は，最近まで堅調に安全性評価を積み重ねてきた。しかし，2022 年の Castryck–Decru の攻撃法 [22] を始めとする一連の攻撃法 [86, 105] は SIDH 鍵共有に対して決定的な結果をもたらした。

一方，本章においても随所に見られるように，Kani の補題に基づいて楕円曲線同種写像を高次元同種写像に埋め込むことで，平滑次数でない同種写像も暗号演算に取り込むことが可能になるなど，SIDH 攻撃法に端を発した全く新しい同種写像暗号研究が現在展開されつつある。例えば，SIDH 攻撃の発案者である Castryck は，“An Attack Became a Tool: Isogeny-based Cryptography 2.0” と題する EUROCRYPT 2024 の招待講演において，同種写像暗号研究が今新しい転換点に差し掛かっており，その技術的な核となるのが高次元同種写像の利用であると述べている。更に，6.1.3 節で示したレベル構造付き同種写像問題などの新たな安全性解析の枠組みに関しても研究が進んでおり，そのような理論的基盤に基づいて，新しい方式提案も含む活発な研究活動が引き続いて行われている。

現在，特に，公開鍵と署名を合わせたサイズが小さい SQIsign 署名方式が注目されていると共に，6.1.4 節で見たような CSIDH ベースの一方向性群作用に関する研究も注目されており，種々の暗号プロトコルへの応用も視野に入れた研究も進んでいる。それらも含めて，今後，特に注意すべきこと数点について以下にまとめておく。

- SQIsign 署名方式は，公開鍵と署名のサイズの小ささ，補助点なしの署名構成，そして短署名に対する強い社会的ニーズなどを踏まえると，現在有望な同種写像暗号技術と思われる。その一方，零知識性に関する計算問題（定義 6.11）の安全性検討などに関して，まだ安全性評価が不十分であり，その安全性評価は今後の重要な課題の一つである。さらに，今後は，実装研究を進める必要もあり，特にさまざまなプラットフォームでの実装結果を蓄えていく必要がある。また，SQIsign2D-West 論文 [7]（ASIACRYPT 2024）の副題は “The Fast, the Small, and the Safer” となっており，2 次元同種写像の利用により演算速度，データサイズ，安全性と多方面での改善が図られており，この方向性での今後の研究進展に注目していく必要がある。
- SQIsign 署名方式は NIST PQC 標準化プロジェクト追加署名第 2 ラウンドに進むことが決定しており [2]，SQIsign（及び SQIsign2D）署名パラメータに対して，（一般的な）超特異同種写像問題及びそれと同値な自己準同型環計算問題に対する古典・量子アルゴリズムの詳細な解析・見積もりを行うことが今後の重要な課題である。
- 鍵共有方式として，レベル構造付き同種写像問題に基づく M-SIDH 鍵共有，(Q)FESTA 鍵共有，binSIDH 鍵共有を 6.2.2 節で取り上げた。群作用ベースの鍵共有には，例えば，CSIDH，SCALLOP，SiGamal などがあるが，他にも POKE，IS-CUBE，LIT-SiGamal など新たな鍵共有・暗号方式が提案されてきており，これからも同種写像に基づく鍵共有・暗号方式の安全性解析と方式改良（及び新規提案）は大変重要な課題である。
- 6.2.3 節で述べたリング署名・グループ署名の他にもパスワード認証鍵共有（PAKE）[1, 73] や紛失疑似ランダム関数（OPRF）[109] などといった一方向性群作用の暗号応用に関する研究が進められており，耐量子計算機

性をもつ方式として注目する必要がある。更に，近年では 6.1.4.3 節で見たように量子マネーなどの新しい応用研究も進んでおり，一方向性群作用の新たな暗号応用を探ることも今後の重要な課題の一つである。

- 上で述べたように高次元同種写像を利用した暗号・署名構成，及び安全性解析は，現在も研究が進展し続けている。全体に，同種写像暗号技術は，まだまだ研究の余地があり，鍵・暗号文・署名サイズの小ささの点で他の耐量子計算機暗号にない特長があるので，さまざまな利用用途を見据えて今後も継続的な研究が望まれる。

第 6 章の参照文献

[1] M. Abdalla, T. Eisenhofer, E. Kiltz, S. Kunzweiler, D. Riepel. Password-Authenticated Key Exchange from Group Actions. CRYPTO (2). Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 699–728.

[2] G. Alagic et al. Status Report on the First Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8528, `https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8528.pdf`. 2024-10.

[3] N. Alamati, L. De Feo, H. Montgomery, S. Patranabis. Cryptographic Group Actions and Applications. ASIACRYPT (2). Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 411–439.

[4] B. Allombert, J.-F. Biasse, J. Komada Eriksen, P. Kutas, C. Leonardi, A. Page, R. Scheidler, M. Tot Bagi. PEARL-SCALLOP: Parameter Extension Applicable in Real-Life SCALLOP. Cryptology ePrint Archive, Paper 2024/1744. 2024. `https://eprint.iacr.org/2024/1744`.

[5] S. Arpin, W. Castryck, J. Komada Eriksen, G. Lorenzon, F. Vercauteren. Generalized class group actions on oriented elliptic curves with level structure. Cryptology ePrint Archive, Paper 2024/1172. 2024. `https://eprint.iacr.org/2024/1172`. to appear in the proceedings of WAIFI 2024.

[6] A. Basso. POKE: A Framework for Efficient PKEs, Split KEMs, and OPRFs from Higher-dimensional Isogenies. Cryptology ePrint Archive, Paper 2024/624. 2024. `https://eprint.iacr.org/2024/624`.

[7] A. Basso, L. De Feo, P. Dartois, A. Leroux, L. Maino, G. Pope, D. Robert, B. Wesolowski. SQIsign2D-West - The Fast, the Small, and the Safer. 2024.

[8] A. Basso, T. B. Fouotsa. New SIDH Countermeasures for a More Efficient Key Exchange. ASIACRYPT (8). Vol. 14445. Lecture Notes in Computer Science. Springer, 2023, pp. 208–233.

[9] A. Basso, L. Maino, G. Pope. FESTA: Fast Encryption from Supersingular Torsion Attacks. ASIACRYPT (7). Vol. 14444. Lecture Notes in Computer Science. Springer, 2023, pp. 98–126.

[10] B. Bencina, P. Kutas, S.-P. Merz, C. Petit, M. Stopar, C. Weitkämper. Improved Algorithms for Finding Fixed-Degree Isogenies Between Supersingular Elliptic Curves. CRYPTO (5). Vol. 14924. Lecture Notes in Computer Science. Springer, 2024, pp. 183–217.

[11] D. J. Bernstein, L. De Feo, A. Leroux, B. Smith. Faster computation of isogenies of large prime degree. ANTS 2020. Vol. 4. The Open Book Series 1. Mathematical Sciences Publishers, 2020, pp. 39–55.

[12] D. J. Bernstein, T. Lange, C. Martindale, L. Panny. Quantum Circuits for the CSIDH: Optimizing Quantum Evaluation of Isogenies. EUROCRYPT (2). Vol. 11477. Lecture Notes in Computer Science. Springer, 2019, pp. 409–441.

[13] W. Beullens, S. Dobson, S. Katsumata, Y.-F. Lai, F. Pintore. Group Signatures and More from Isogenies and Lattices: Generic, Simple, and Efficient. EUROCRYPT (2). Vol. 13276. Lecture Notes in Computer Science. Springer, 2022, pp. 95–126.

[14] W. Beullens, S. Katsumata, F. Pintore. Calamari and Falafl: Logarithmic (Linkable) Ring Signatures from Isogenies and Lattices. ASIACRYPT (2). Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 464–492.

[15] W. Beullens, T. Kleinjung, F. Vercauteren. CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations. ASIACRYPT (1). Vol. 11921. Lecture Notes in Computer Science. Springer, 2019, pp. 227–247.

[16] J.-F. Biasse, D. Jao, A. Sankar. A Quantum Algorithm for Computing Isogenies between Supersingular Elliptic Curves. INDOCRYPT. Vol. 8885. Lecture Notes in Computer Science. Springer, 2014, pp. 428–442.

[17] D. Boneh, J. Guan, M. Zhandry. A Lower Bound on the Length of Signatures Based on Group Actions and Generic Isogenies. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 507–531.

[18] X. Bonnetain, A. Schrottenloher. Quantum Security Analysis of CSIDH. EUROCRYPT (2). Vol. 12106. Lecture Notes in Computer Science. Springer, 2020, pp. 493–522.

[19] G. Bruno, M. Corte-Real Santos, C. Costello, J. Komada Eriksen, M. Meyer, M. Naehrig, B. Sterner. Cryptographic Smooth Neighbors. Cryptology ePrint Archive, Paper 2022/1439. 2022. `https://eprint.iacr.org/2022/1439`.

[20] F. Campos, J. Chávez-Saab, J.-J. Chi-Domínguez, M. Meyer, K. Reijnders, F. Rodríguez-Henríquez, P. Schwabe, T. Wiggers. Optimizations and Practicality of High-Security CSIDH. IACR Commun. Cryptol. Vol. 1, Num. 1 (2024), p. 5.

[21] W. Castryck, M. Chen, R. Invernizzi, G. Lorenzon, F. Vercauteren. Breaking and Repairing SQIsign2D-East. Cryptology ePrint Archive, Paper 2024/1453. 2024. `https://eprint.iacr.org/2024/1453`. to appear in the proceedings of ASIACRYPT 2024 merging with [96].

[22] W. Castryck, T. Decru. An Efficient Key Recovery Attack on SIDH. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 423–447.

[23] W. Castryck, T. Decru, M. Houben, F. Vercauteren. Horizontal Racewalking Using Radical Isogenies. ASIACRYPT (2). Vol. 13792. Lecture Notes in Computer Science. Springer, 2022, pp. 67–96.

[24] W. Castryck, T. Decru, B. Smith. Hash functions from superspecial genus-2 curves using Richelot isogenies. J. Math. Cryptol. Vol. 14, Num. 1 (2020), pp. 268–292.

[25] W. Castryck, T. Decru, F. Vercauteren. Radical Isogenies. ASIACRYPT (2). Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 493–519.

[26] W. Castryck, M. Houben, S.-P. Merz, M. Mula, S. van Buuren, F. Vercauteren. Weak Instances of Class Group Action Based Cryptography via Self-pairings. CRYPTO (3). Vol. 14083. Lecture Notes in Computer Science. Springer, 2023, pp. 762–792.

[27] W. Castryck, M. Houben, F. Vercauteren, B. Wesolowski. On the decisional Diffie–Hellman problem for class group actions on oriented elliptic curves. ANTS 2022. Vol. 8. Research in Number Theory 99. Springer, 2022, pp. 39–55.

[28] W. Castryck, T. Lange, C. Martindale, L. Panny, J. Renes. CSIDH: An Efficient Post-Quantum Commutative Group Action. ASIACRYPT (3). Vol. 11274. Lecture Notes in Computer Science. Springer, 2018, pp. 395–427.

[29] W. Castryck, N. Vander Meeren. Two Remarks on the Vectorization Problem. INDOCRYPT. Vol. 13774. Lecture Notes in Computer Science. Springer, 2022, pp. 658–678.

[30] W. Castryck, J. Sotáková, F. Vercauteren. Breaking the Decisional Diffie-Hellman Problem for Class Group Actions Using Genus Theory: Extended Version. J. Cryptol. Vol. 35, Num. 4 (2022), p. 24.

[31] W. Castryck, F. Vercauteren. A Polynomial Time Attack on Instances of M-SIDH and FESTA. ASIACRYPT (7). Vol. 14444. Lecture Notes in Computer Science. Springer, 2023, pp. 127–156.

[32] D. X. Charles, K. E. Lauter, E. Z. Goren. Cryptographic Hash Functions from Expander Graphs. J. Cryptol. Vol. 22, Num. 1 (2009), pp. 93–113.

[33] J. Chavez-Saab et al. SQIsign: Algorithm specifications and supporting documentation. submission to the NIST's PQC standardization. (2023).

[34] J. Chávez-Saab, J.-J. Chi-Domínguez, S. Jaques, F. Rodríguez-Henríquez. The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. J. Cryptogr. Eng. Vol. 12, Num. 3 (2022), pp. 349–368.

[35] M. Chen, A. Leroux, L. Panny. SCALLOP-HD: Group Action from 2-Dimensional Isogenies. Public Key Cryptography (3). Vol. 14603. Lecture Notes in Computer Science. Springer, 2024, pp. 190–216.

[36] A. M. Childs, D. Jao, V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. J. Math. Cryptol. Vol. 8, Num. 1 (2014), pp. 1–29.

[37] L. Colò, D. Kohel. Orienting supersingular isogeny graphs. J. Math. Cryptol. Vol. 14, Num. 1 (2020), pp. 414–437.

[38] C. Costello. B-SIDH: Supersingular Isogeny Diffie-Hellman Using Twisted Torsion. ASIACRYPT (2). Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 440–463.

[39] C. Costello. The Case for SIKE: A Decade of the Supersingular Isogeny Problem. Cryptology ePrint Archive, Paper 2021/543. 2021. https://eprint.iacr.org/2021/543.

[40] C. Costello, P. Longa, M. Naehrig, J. Renes, F. Virdia. Improved Classical Cryptanalysis of SIKE in Practice. Public Key Cryptography (2). Vol. 12111. Lecture Notes in Computer Science. Springer, 2020, pp. 505–534.

[41] J.-M. Couveignes. Hard Homogeneous Spaces. Cryptology ePrint Archive, Paper 2006/291. 2006. https://eprint.iacr.org/2006/291.

[42] P. Dartois. Fast computation of 2-isogenies in dimension 4 and cryptographic applications. Cryptology ePrint Archive, Paper 2024/1180. 2024. https://eprint.iacr.org/2024/1180.

[43] P. Dartois, L. De Feo. On the Security of OSIDH. Public Key Cryptography (1). Vol. 13177. Lecture Notes in Computer Science. Springer, 2022, pp. 52–81.

[44] P. Dartois, A. Leroux, D. Robert, B. Wesolowski. SQIsignHD: New Dimensions in Cryptography. EUROCRYPT (1). Vol. 14651. Lecture Notes in Computer Science. Springer, 2024, pp. 3–32.

[45] P. Dartois, L. Maino, G. Pope, D. Robert. An Algorithmic Approach to $(2, 2)$-isogenies in the Theta Model and Applications to Isogeny-based Cryptography. Cryptology ePrint Archive, Paper 2023/1747. 2023. https://eprint.iacr.org/2023/1747.

[46] T. Decru. Radical $\sqrt[N]{\text{élu}}$ Isogeny Formulae. CRYPTO (5). Vol. 14924. Lecture Notes in Computer Science. Springer, 2024, pp. 107–128.

[47] T. Decru, L. Panny, F. Vercauteren. Faster SeaSign Signatures Through Improved Rejection Sampling. PQCrypto. Vol. 11505. Lecture Notes in Computer Science. Springer, 2019, pp. 271–285.

[48] C. Delfs, S. D. Galbraith. Computing isogenies between supersingular elliptic curves over F_p. Des. Codes Cryptogr. Vol. 78, Num. 2 (2016), pp. 425–440.

[49] M. Duparc, T. B. Fouotsa. SQIPrime: A Dimension 2 Variant of SQISignHD with Non-smooth Challenge Isogenies. 2024.

[50] K. Eisenträger, S. Hallgren, K. E. Lauter, T. Morrison, C. Petit. Supersingular Isogeny Graphs and Endomorphism Rings: Reductions and Solutions. EUROCRYPT (3). Vol. 10822. Lecture Notes in Computer Science. Springer, 2018, pp. 329–368.

[51] K. Eisenträger, S. Hallgren, C. Leonardi, T. Morrison, J. Park. Computing endomorphism rings of supersingular elliptic curves and connections to pathfinding in isogeny graphs. ANTS 2020. Vol. 4. The Open Book Series 1. Mathematical Sciences Publishers, 2020, pp. 215–232.

[52] A. El Kaafarani, S. Katsumata, F. Pintore. Lossy CSI-FiSh: Efficient Signature Scheme with Tight Reduction to Decisional CSIDH-512. Public Key Cryptography (2). Vol. 12111. Lecture Notes in Computer Science. Springer, 2020, pp. 157–186.

[53] J. Komada Eriksen, L. Panny, J. Sotáková, M. Veroni. Deuring for the People: Supersingular Elliptic Curves with Prescribed Endomorphism Ring in General Characteristic. LuCaNT: LMFDB, Computation, and Number Theory. Vol. 796. Contemporary Mathematics. AMS, 2024. `https://www.ams.org/books/conm/796/16008/conm796-16008.pdf`.

[54] L. De Feo. Mathematics of Isogeny Based Cryptography. 2017. arXiv: `1711.04062`.

[55] L. De Feo, T. B. Fouotsa, P. Kutas, A. Leroux, S.-P. Merz, L. Panny, B. Wesolowski. SCALLOP: Scaling the CSI-FiSh. Public Key Cryptography (1). Vol. 13940. Lecture Notes in Computer Science. Springer, 2023, pp. 345–375.

[56] L. De Feo, T. B. Fouotsa, L. Panny. Isogeny Problems with Level Structure. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 181–204.

[57] L. De Feo, S. D. Galbraith. SeaSign: Compact Isogeny Signatures from Class Group Actions. EUROCRYPT (3). Vol. 11478. Lecture Notes in Computer Science. Springer, 2019, pp. 759–789.

[58] L. De Feo, D. Jao, J. Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. J. Math. Cryptol. Vol. 8, Num. 3 (2014), pp. 209–247.

[59] L. De Feo, J. Kieffer, B. Smith. Towards Practical Key Exchange from Ordinary Isogeny Graphs. ASIACRYPT (3). Vol. 11274. Lecture Notes in Computer Science. Springer, 2018, pp. 365–394.

[60] L. De Feo, D. Kohel, A. Leroux, C. Petit, B. Wesolowski. SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies. ASIACRYPT (1). Vol. 12491. Lecture Notes in Computer Science. Springer, 2020, pp. 64–93.

[61] L. De Feo, A. Leroux, P. Longa, B. Wesolowski. New Algorithms for the Deuring Correspondence – Towards Practical and Secure SQISign Signatures. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 659–690.

[62] L. De Feo, C. D. de Saint Guilhem, T. B. Fouotsa, P. Kutas, A. Leroux, C. Petit, J. Silva, B. Wesolowski. Séta: Supersingular Encryption from Torsion Attacks. ASIACRYPT (4). Vol. 13093. Lecture Notes in Computer Science. Springer, 2021, pp. 249–278.

[63] E. Florit, B. Smith. An atlas of the Richelot isogeny graph. RIMS Kôkyûroku Bessatsu. Vol. B90 (2022), pp. 195–219. `https://repository.kulib.kyoto-u.ac.jp/dspace/handle/2433/276282`.

[64] T. B. Fouotsa, T. Moriya, C. Petit. M-SIDH and MD-SIDH: Countering SIDH Attacks by Masking Information. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 282–309.

[65] T. B. Fouotsa, C. Petit. SimS: A Simplification of SiGamal. PQCrypto. Vol. 12841. Lecture Notes in Computer Science. Springer, 2021, pp. 277–295.

[66] J. Fuselier, A. Iezzi, M. Kozek, T. Morrison, C. Namoijam. Computing supersingular endomorphism rings using inseparable endomorphisms. 2023. arXiv: 2306.03051.

[67] S. Galbraith, L. Panny, B. Smith, F. Vercauteren. Quantum equivalence of the DLP and CDHP for group actions. Mathematical Cryptology. Vol. 1, Num. 1 (2021), pp. 40–44. https://journals.flvc.org/mathcryptology/article/view/122741.

[68] S. D. Galbraith. Constructing Isogenies between Elliptic Curves Over Finite Fields. LMS Journal of Computation and Mathematics. Vol. 2 (1999), pp. 118–138.

[69] S. D. Galbraith, Y.-F. Lai, H. Montgomery. A Simpler and More Efficient Reduction of DLog to CDH for Abelian Group Actions. Public Key Cryptography (3). Vol. 14603. Lecture Notes in Computer Science. Springer, 2024, pp. 36–60.

[70] S. D. Galbraith, D. Perrin, J. F. Voloch. CSIDH with Level Structure. Cryptology ePrint Archive, Paper 2023/1726. 2023. https://eprint.iacr.org/2023/1726.

[71] S. D. Galbraith, C. Petit, J. Silva. Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems. J. Cryptol. Vol. 33, Num. 1 (2020), pp. 130–175.

[72] S. D. Galbraith, F. Vercauteren. Computational problems in supersingular elliptic curve isogenies. Quantum Inf. Process. Vol. 17, Num. 10 (2018), p. 265.

[73] R. Ishibashi, K. Yoneyama. Compact Password Authenticated Key Exchange from Group Actions. ACISP. Vol. 13915. Lecture Notes in Computer Science. Springer, 2023, pp. 220–247.

[74] D. Jao et al. Supersingular Isogeny Key Encapsulation. https://sike.org/files/SIDH-spec.pdf. 2022-09. (2024-11-12 閲覧).

[75] S. Jaques, J. M. Schanck. Quantum Cryptanalysis in the RAM Model: Claw-Finding Attacks on SIKE. CRYPTO (1). Vol. 11692. Lecture Notes in Computer Science. Springer, 2019, pp. 32–61.

[76] Y. Kambe, A. Katayama, Y. Aikawa, Y. Ishihara, M. Yasuda, K. Yokoyama. Computing Endomorphism Rings of Supersingular Elliptic Curves by Finding Cycles in Concatenated Supersingular Isogeny Graphs. Commentarii Mathematici Universitatis Sancti Pauli. Vol. 72, Num. 1 (2024), pp. 19–42.

[77] Y. Kambe, Y. Takahashi, M. Yasuda, K. Yokoyama. On the feasibility of computing constructive Deuring correspondence. NuTMiC 2021. Vol. 126. Banach Center Publications. Institute of Mathematics, Polish Academy od Sciences, 2023. https://www.impan.pl/en/publishing-house/banach-center-publications/all/126/0/115356/on-the-feasibility-of-computing-constructive-deuring-correspondence.

[78] T. Katsura, K. Takashima. Counting Richelot isogenies between superspecial abelian surfaces. ANTS 2020. Vol. 4. The Open Book Series 1. Mathematical Sciences Publishers, 2020, pp. 283–300.

[79] D. Kohel. Endomorphism rings of elliptic curves over finite fields. PhD thesis. University of California at Berkeley, 1996.

[80] D. Kohel, K. Lauter, C. Petit, J.-P. Tignol. On the quaternion ℓ-isogeny path problem. LMS Journal of Computation and Mathematics. Vol. 17 (2014), pp. 418–432. Special Issue A: Algorithmic Number Theory Symposium XI.

[81] G. Kuperberg. A Subexponential-Time Quantum Algorithm for the Dihedral Hidden Subgroup Problem. SIAM J. Comput. Vol. 35, Num. 1 (2005), pp. 170–188.

[82] G. Kuperberg. Another Subexponential-time Quantum Algorithm for the Dihedral Hidden Subgroup Problem. 8th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2013, May 21-23, 2013, Guelph, Canada. Ed. by S. Severini, F. G. S. L. Brandão. Vol. 22. LIPIcs. 2013, pp. 20–34.

[83] A. Leroux. Quaternion algebras and isogeny-based cryptography. PhD thesis. Ecole Polytechnique, 2022.

[84] J. Liu, H. Montgomery, M. Zhandry. Another Round of Breaking and Making Quantum Money: How to Not Build It from Lattices, and More. EUROCRYPT (1). Vol. 14004. Lecture Notes in Computer Science. Springer, 2023, pp. 611–638.

[85] P. Longa, W. Wang, J. Szefer. The Cost to Break SIKE: A Comparative Hardware-Based Analysis with AES and SHA-3. CRYPTO (3). Vol. 12827. Lecture Notes in Computer Science. Springer, 2021, pp. 402–431.

[86] L. Maino, C. Martindale, L. Panny, G. Pope, B. Wesolowski. A Direct Key Recovery Attack on SIDH. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 448–471.

[87] A. Herlédan Le Merdy, B. Wesolowski. The supersingular endomorphism ring problem given one endomorphism. Cryptology ePrint Archive, Paper 2023/1448. 2023. `https://eprint.iacr.org/2023/1448`.

[88] M. Meyer, S. Reith. A Faster Way to the CSIDH. INDOCRYPT. Vol. 11356. Lecture Notes in Computer Science. Springer, 2018, pp. 137–152.

[89] H. Montgomery, S. Sharif. Quantum Money from Class Group Actions on Elliptic Curves. 2024.

[90] H. Montgomery, M. Zhandry. Full Quantum Equivalence of Group Action DLog and CDH, and More. ASIACRYPT (1). Vol. 13791. Lecture Notes in Computer Science. Springer, 2022, pp. 3–32.

[91] T. Moriya. IS-CUBE: An isogeny-based compact KEM using a boxed SIDH diagram. Cryptology ePrint Archive, Paper 2023/1506. 2023. `https://eprint.iacr.org/2023/1506`.

[92] T. Moriya. LIT-SiGamal: An efficient isogeny-based PKE based on a LIT diagram. Cryptology ePrint Archive, Paper 2024/521. 2024. `https://eprint.iacr.org/2024/521`.

[93] T. Moriya, H. Onuki, T. Takagi. SiGamal: A Supersingular Isogeny-Based PKE and Its Application to a PRF. ASIACRYPT (2). Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 551–580.

[94] S. Mutreja, M. Zhandry. Quantum State Group Actions. Cryptology ePrint Archive, Paper 2024/1636. 2024. `https://eprint.iacr.org/2024/1636`.

[95] K. Nakagawa, H. Onuki. QFESTA: Efficient Algorithms and Parameters for FESTA Using Quaternion Algebras. CRYPTO (5). Vol. 14924. Lecture Notes in Computer Science. Springer, 2024, pp. 75–106.

[96] K. Nakagawa, H. Onuki. SQIsign2D-East: A New Signature Scheme Using 2-dimensional Isogenies. Cryptology ePrint Archive, Paper 2024/771. 2024. `https://eprint.iacr.org/2024/771`. to appear in the proceedings of ASIACRYPT 2024 merging with [21].

[97] H. Onuki, Y. Aikawa, T. Yamazaki, T. Takagi. A Constant-Time Algorithm of CSIDH Keeping Two Points. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. Vol. 103-A, Num. 10 (2020), pp. 1174–1182.

[98] R. Oudompheng, G. Pope. A Note on Reimplementing the Castryck–Decru Attack and Lessons Learned for SageMath. Cryptology ePrint Archive, Paper 2022/1283. 2022. https://eprint.iacr.org/2022/1283.

[99] A. Page, B. Wesolowski. The Supersingular Endomorphism Ring and One Endomorphism Problems are Equivalent. EUROCRYPT (6). Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 388–417.

[100] C. Peikert. He Gives C-Sieves on the CSIDH. EUROCRYPT (2). Vol. 12106. Lecture Notes in Computer Science. Springer, 2020, pp. 463–492.

[101] C. Petit. Faster Algorithms for Isogeny Problems Using Torsion Point Images. ASIACRYPT (2). Vol. 10625. Lecture Notes in Computer Science. Springer, 2017, pp. 330–353.

[102] V. de Quehen, P. Kutas, C. Leonardi, C. Martindale, L. Panny, C. Petit, K. E. Stange. Improved Torsion-Point Attacks on SIDH Variants. CRYPTO (3). Vol. 12827. Lecture Notes in Computer Science. Springer, 2021, pp. 432–470.

[103] O. Regev. A Subexponential Time Algorithm for the Dihedral Hidden Subgroup Problem with Polynomial Space. 2004. arXiv: quant-ph/0406151.

[104] J. Renes. Computing Isogenies Between Montgomery Curves Using the Action of $(0, 0)$. PQCrypto. Vol. 10786. Lecture Notes in Computer Science. Springer, 2018, pp. 229–247.

[105] D. Robert. Breaking SIDH in Polynomial Time. EUROCRYPT (5). Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 472–503.

[106] D. Robert. On the efficient representation of isogenies (a survey). Cryptology ePrint Archive, Paper 2024/1071. 2024. https://eprint.iacr.org/2024/1071.

[107] D. Robert. The module action for isogeny based cryptography. Cryptology ePrint Archive, Paper 2024/1556. 2024. https://eprint.iacr.org/2024/1556.

[108] A. Rostovtsev, A. Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Paper 2006/145. 2006. https://eprint.iacr.org/2006/145.

[109] C. D. de Saint Guilhem, R. Pedersen. New Proof Systems and an OPRF from CSIDH. Public Key Cryptography (3). Vol. 14603. Lecture Notes in Computer Science. Springer, 2024, pp. 217–251.

[110] M. Corte-Real Santos, C. Costello, J. Shi. Accelerating the Delfs-Galbraith Algorithm with Fast Subfield Root Detection. CRYPTO (3). Vol. 13509. Lecture Notes in Computer Science. Springer, 2022, pp. 285–314.

[111] M. Corte-Real Santos, J. Komada Eriksen, M. Meyer, K. Reijnders. AprèsSQI: Extra Fast Verification for SQIsign Using Extension-Field Signing. EUROCRYPT (1). Vol. 14651. Lecture Notes in Computer Science. Springer, 2024, pp. 63–93.

[112] B. Smith. Pre- and Post-quantum Diffie-Hellman from Groups, Actions, and Isogenies. WAIFI. Vol. 11321. Lecture Notes in Computer Science. Springer, 2018, pp. 3–40.

[113] K. Takashima. Efficient Algorithms for Isogeny Sequences and Their Cryptographic Applications. *CREST Crypto-Math Project*. Mathematics for Industry. Springer Singapore, 2017, pp. 97–114.

[114] A. Udovenko, G. Vitto. Revisiting Meet-in-the-Middle Cryptanalysis of SIDH/SIKE with Application to the $IKEp182 Challenge. Cryptology ePrint Archive, Paper 2021/1421. 2021. https://eprint.iacr.org/2021/1421.

[115] J. Vélu. Isogénies entre courbes elliptiques. C. R. Acad. Sci. Paris, Sér. A. Vol. 273 (1971), pp. 305–347.

[116] J. Voight. Quaternion algebras. Springer International Publishing, 2021-06.

[117] L. C. Washington. Elliptic curves : number theory and cryptography. 2nd ed. Discrete mathematics and its applications. Chapman & Hall/CRC, 2008.

[118] B. Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. FOCS. IEEE, 2021, pp. 1100–1111.

[119] M. Zhandry. Quantum Money from Abelian Group Actions. ITCS. Vol. 287. LIPIcs. 2024, 101:1–101:23.

[120] 相川 勇輔, 神戸 祐太, 工藤 桃成, 高島 克幸, 安田 雅哉. 代数曲線の計算理論と暗号への応用. 数学メモアール 10. 日本数学会, 2024.

第 7 章

ハッシュ関数に基づく署名技術

本章ではハッシュ関数に基づく署名技術についてまとめる。ハッシュ関数に基づく署名技術の安全性はハッシュ関数の第二原像攻撃に対する安全性に依存している。

ハッシュ関数に基づく署名技術は，最初に Lamport により one-time signature として提案された [15, 31]。また，この方式を改良した Winternitz one-time signature が Merkle [35] により述べられている。これらの方式は一組の公開鍵と秘密鍵を用いて一つのメッセージに署名を行う 1 回署名方式である。1 回署名方式とマークル木とを用いて複数回署名を行うことを可能とする方式が Merkle [34, 35] により述べられている。

7.1 ハッシュ関数に基づく署名技術の安全性の根拠となる問題

ハッシュ関数は任意長あるいは実用上十分な長さ以下の入力 $\{0,1\}$ 系列に対して固定長の $\{0,1\}$ 系列を出力する関数である。ハッシュ関数を $H : \mathcal{D} \to \mathcal{R}$ とする。ここで，$\mathcal{D}$ は任意長の $\{0,1\}$ 系列の集合 $\{0,1\}^*$ の部分集合であり，$\mathcal{R}$ は固定長の $\{0,1\}$ 系列の集合である。ハッシュ関数の第二原像攻撃は，第一原像 $X \in \mathcal{D}$ が与えられたとき，$X \neq X'$ かつ $H(X) = H(X')$ を満たす第二原像 $X' \in \mathcal{D}$ を求めるという問題を解くことを目的とする攻撃である。なお，第二原像攻撃に対する安全性は，しばしば，ハッシュ関数が各入力に対する出力を無作為に選択するランダム関数であると仮定して評価される。このようなランダム関数はランダムオラクルとも呼ばれる。H がランダムオラクルであるとき，第二原像の計算時間は $\Theta(|\mathcal{R}|)$ である。また，量子コンピュータでは，Grover の探索アルゴリズム [20] を用いることにより，第二原像の計算時間は $\Theta(\sqrt{|\mathcal{R}|})$ となる。

本章で取り上げるハッシュ関数に基づく署名技術では，米国 NIST の指定する標準ハッシュ関数族である SHA-2 [38], SHA-3 [39] のうちのいくつかのハッシュ関数を用いることが想定されている。

SHA-2 は固定長入出力の圧縮関数からなる Merkle-Damgård 構造 [14, 36] を有するハッシュ関数の族であり，Secure Hash Standard [38] のうち，SHA-1 を除く SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 からなる。SHA-2 の各ハッシュ関数の名称の末尾の数値は出力の bit 長を表す。SHA-3 は固定長入出力の置換を用いたスポンジ構造 [9] を有するハッシュ関数の族であり，SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256 からなる。SHA3-224, SHA3-256, SHA3-384, SHA3-512 については，末尾の数値は出力の bit 長を表す。なお，出力長は内部状態のうちスポンジ構造を有するハッシュ関数の安全性に大きく関わるキャパシティと呼ばれる部分の bit 長の半分である。SHAKE128, SHAKE256 については，出力長は任意に設定できる。なお，末尾の数値はキャパシティと呼ばれる部分の bit 長の半分である。SHA-2, SHA-3 の出力長および安全性の一覧を表 7.1 に示す。この表では，第二原像攻撃に対する安全性のみでなく，それとともにハッシュ関数の主な安全性要件である衝突攻撃，原像攻撃に対する安全性も示されている。この表で，攻撃に対する安全性を表す数値 μ は bit 安全性と呼ばれ，攻撃に成功するために必要な計算時間（ハッシュ関数を構成する圧縮関数あるいは置換の計算回数）がおよそ 2^{μ} であることを示している。なお，SHA-224, SHA-256, SHA-512 の第二原像攻撃に対する安全性は Kelsey と Schneier により提案

された攻撃 [29] に基づいて評価されており，$L(m) := \lceil \log_2(m/B) \rceil$ である。ここで，m は第一原像の bit 長であり，SHA-224, SHA-256 については $B = 512$，SHA-512 については $B = 1024$ である。

表 7.1: SHA-2, SHA-3 の安全性 [39]

ハッシュ関数	出力長	攻撃に対する安全性		
		衝突攻撃	原像攻撃	第二原像攻撃
SHA-224	224	112	224	$\min\{224, 256 - L(m)\}$
SHA-512/224	224	112	224	224
SHA-256	256	128	256	$256 - L(m)$
SHA-512/256	256	128	256	256
SHA-384	384	192	384	384
SHA-512	512	256	512	$512 - L(m)$
SHA3-224	224	112	224	224
SHA3-256	256	128	256	256
SHA3-384	384	192	384	384
SHA3-512	512	256	512	512
SHAKE128	d	$\min\{d/2, 128\}$	$\min\{d, 128\}$ 以上	$\min\{d, 128\}$
SHAKE256	d	$\min\{d/2, 256\}$	$\min\{d, 256\}$ 以上	$\min\{d, 256\}$

本章で使用する記号・用語を以下にまとめる。

- $\{0,1\}$ 系列 α, β の連接を $\alpha \| \beta$ と表記する。
- $\ll$ は左論理シフトを表す。
- 整数 ν について $[\nu]_l$ は ν の長さ l Bytes の 2 進数表記を表す。
- $\mathbb{B} := \{0,1\}^8$ とする。
- `8080` のように typewriter font で書かれている数字は 16 進数として解釈する。

7.2 ハッシュ関数に基づく代表的な署名方式

7.2.1 Winternitz One-Time Signature

Winternitz one-time signature [35] は，一組の公開鍵と秘密鍵を用いて一つのメッセージに署名を行う 1 回署名方式である。この方式では，署名対象のメッセージのハッシュ値 N を b 進数表記の整数とみなす。N が ℓ_{m} 桁の b 進数 $N_{\ell_{\mathrm{m}}-1} N_{\ell_{\mathrm{m}}-2} \cdots N_1 N_0$ で表記されるとする。このとき，$0 \leq k \leq \ell_{\mathrm{m}} - 1$ について $N_k \in \{0, 1, \ldots, b-1\}$ であり，$N = \sum_{k=0}^{\ell_{\mathrm{m}}-1} N_k 2^k$ である。さらに，N のチェックサムを $C := \sum_{k=0}^{\ell_{\mathrm{m}}-1} (b - 1 - N_k)$ と定義する。C が ℓ_{c} 桁の b 進数 $N_{\ell_{\mathrm{m}}+\ell_{\mathrm{c}}-1} N_{\ell_{\mathrm{m}}+\ell_{\mathrm{c}}-2} \cdots N_{\ell_{\mathrm{m}}+1} N_{\ell_{\mathrm{m}}}$ で表記されるとする。$\ell := \ell_{\mathrm{m}} + \ell_{\mathrm{c}}$ とする。

■鍵生成アルゴリズム　秘密鍵 $(x_0, x_1, \ldots, x_{\ell-1})$，公開鍵 $(pub_0, pub_1, \ldots, pub_{\ell-1})$ は以下のように生成される。

1. $x_0, x_1, \ldots, x_{\ell-1} \in \mathcal{D}$ を無作為に選択する。
2. $0 \leq k \leq \ell - 1$ について $pub_k := H^{b-1}(x_k) := \underbrace{H(H(\cdots(H}_{b-1 \text{ times}}(x_k))\cdots))$ とする。

■署名アルゴリズム　メッセージのハッシュ値 N の署名 $(s_0, s_1, \ldots, s_{\ell-1})$ は以下のように生成される。

1. $0 \leq k \leq \ell - 1$ について $s_k := H^{N_k}(x_k)$ とする。

■検証アルゴリズム　メッセージのハッシュ値 N とその署名 $(s_0, s_1, \ldots, s_{\ell-1})$ の検証は以下のように行われる。

1. $0 \leq k \leq \ell - 1$ について $pub_k = H^{b-1-N_k}(s_k)$ かつそのときに限り，$(s_0, s_1, \ldots, s_{\ell-1})$ は N の正しい署名である。

仮にチェックサムが導入されていないとすると，N の署名 $(s_0, s_1, \ldots, s_{\ell_{\mathrm{m}}-1})$ が得られたとき，$0 \leq k \leq \ell_{\mathrm{m}} - 1$ について $N'_k \geq N_k$ を満たす N' について，$s'_k := H^{N'_k - N_k}(s_k)$ によって，署名 $(s'_0, s'_1, \ldots, s'_{\ell_{\mathrm{m}}-1})$ が容易に偽造できる。

Winternitz one-time signature の偽造不能性は，Dods ら [16] により論じられている。Winternitz one-time signature に基づく方式については，Lafrance と Menezes [30] によりまとめられている。

7.2.2　マークル木を用いた署名方式

1 回署名方式を用いて複数のメッセージに署名を行う場合，メッセージの個数と同じ個数の公開鍵と秘密鍵の組が必要となる。マークル木を用いることにより，このような複数回署名方式の公開鍵の大きさを削減できる [34]。

2^h 個のメッセージに署名を行うための 1 回署名の公開鍵を $pk_0, pk_1, \ldots, pk_{2^h-1}$ とする。このとき，高さが h，すなわち，葉の個数が 2^h のマークル木は以下のように構成される。高さ $j(\geq 0)$ の左から $i(\geq 0)$ 番目の節点を $v_{i,j}$ と表記する。$v_{i,j}$ は以下のように計算される。

1. $0 \leq i \leq 2^h - 1$ について，$v_{i,0} := H(pk_i)$ とする。
2. $1 \leq j \leq h$ に対し，$0 \leq i \leq 2^{h-j} - 1$ について，$v_{i,j} := H(v_{2i,j-1} \| v_{2i+1,j-1})$ とする。

この署名方式の公開鍵は $v_{0,h}$ である。秘密鍵は 1 回署名の公開鍵 $pk_0, pk_1, \ldots, pk_{2^h-1}$ に対応するすべての秘密鍵である。i 個目のメッセージの署名を検証するためには，$v_{0,h}$ を用いて pk_i が正しいことを検証する必要がある。このために，i 個目のメッセージの署名には，マークル木の $v_{i,0}$ から $v_{0,h}$ に至る経路上の各節点の，経路上にない子節点が含まれる。これらの節点の列は認証パスと呼ばれる。

7.2.3　マークル木の階層構造による署名方式

前節で述べた一つのマークル木を用いた署名方式では，鍵生成時にすべての 1 回署名の公開鍵と秘密鍵を生成する必要があり，例えば，2^{50} 個の署名を行うために高さ 50 のマークル木を構成することは，所要計算時間の観点から非実用的である。このような多数のメッセージに署名を行う際には，マークル木を用いた署名方式の階層構造による署名方式が提案されている [24]。

この署名方式で構成されるマークル木を用いた署名方式の階層構造の階層数を L とし，根に相当する最上層を第 $(L-1)$ 層，葉に相当する最下層を第 0 層とする。さらに，$0 \leq i \leq L-1$ について，第 i 層のマークル木の高さはすべて等しく h_i であると仮定する。このとき，第 i 層のマークル木は $2^{\sum_{j=i+1}^{L-1} h_j}$ 個存在する。この署名方式では $2^{\sum_{j=0}^{L-1} h_j}$ 個のメッセージに署名できる。

この署名方式では，第 $(L-1)$ 層のマークル木の根が公開鍵となる。この公開鍵を生成する際には，1 回署名の公開鍵と秘密鍵の組を $2^{h_{L-1}}$ 個だけ生成すれば良い。$0 < i \leq L-1$ について，第 i 層の各マークル木は第 $(i-1)$ 層の 2^{h_i} 個のマークル木の根を署名するために使用される。第 0 層のマークル木は，それぞれ 2^{h_0} 個のメッセージの署名に使用される。

この署名方式では，一つのメッセージの署名の際に，各層についてそれぞれ一つのマークル木を生成しておけば十分である。各メッセージの署名は，そのメッセージに対する第 0 層のマークル木による署名と，$0 < i \leq L-1$ について，そのメッセージの署名の際に使用された第 i 層のマークル木による第 $(i-1)$ 層のマークル木の根の署名からなる。この署名方式について，階層数 $L = 3$，各階層のマークル木の高さ $h_0 = h_1 = h_2 = 3$ の模式図を図 7.1 に示す。灰色の節点は認証パスをなす節点である。

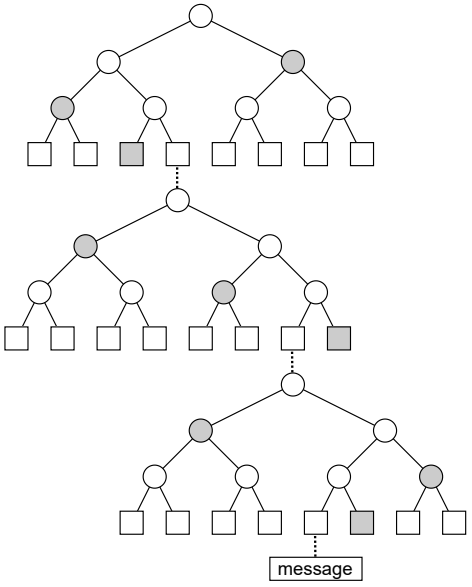


図 7.1: マークル木の階層構造による署名方式

7.2.4 プレフィクスとビットマスク

プレフィクスは，ハッシュ関数に基づく署名方式の処理において，すべてのハッシュ関数の計算がそれぞれ異なる入力に対して行われるよう入力に付加される系列である。プレフィクスは，Lighton と Micali [32] により，security string という名称で，ハッシュ関数に基づく署名方式の安全性をハッシュ関数の第二原像攻撃に対する安全性にタイトに帰着するために導入された。なお，プレフィクスは，ハッシュ関数の用途とそれが用いられる位置（例えば，どの 1 回署名方式か，どのマークル木のどの節点か）により自然に定義できることから，現在は通常，アドレスと呼ばれる。

ビットマスクは，Dahmen ら [13] により，ハッシュ関数に基づく署名方式の安全性をハッシュ関数の第二原像攻撃に対する安全性に帰着するために導入された。ビットマスクは乱数系列であり，ハッシュ関数への入力をランダム化するために，bit ごとの排他的論理和により入力に加えられる。

7.3 ハッシュ関数に基づく主要な署名方式

本章で取り上げるハッシュ関数に基づく署名方式を表 7.2 に示す。

NIST SP 800-208 [12] は，以下のハッシュ関数に基づく stateful な署名方式を規定している。

- Lighton-Micali Signatures (LMS), Hierarchical Signature System (HSS) [33]
- eXtended Merkle Signature Scheme (XMSS), multi-tree XMSS (XMSS^{MT}) [21]

LMS は Lighton と Micali による署名方式 [32] に基づく。HSS, XMSS^{MT} はそれぞれ，7.2.3 節で述べられたような，LMS, XMSS の階層構造による署名方式である。ハッシュ関数に基づく stateful な署名方式では，同一の秘密鍵が複

表 7.2: ハッシュ関数に基づく署名方式

文献	暗号化	鍵交換	署　名
Lighton-Micali Hash-Based Signatures [33, 12]			○
eXtended Merkle Signature Scheme (XMSS) [21, 12]			○
Stateless Hash-Based Digital Signature Algorithm (SLH-DSA) [40]			○

数のメッセージの署名に使用されることがないように秘密鍵を管理することが必須である。

NIST FIPS 205 [40] はハッシュ関数に基づく stateless な署名方式 SLH-DSA (Stateless Hash-Based Digital Signature Algorithm) を規定している。stateless な署名方式では，stateful な方式に求められるような秘密鍵の管理は不要である。SLH-DSA は，2022 年 7 月に NIST PQC 標準化プロジェクトで標準化候補アルゴリズムの一つに選出された SPHINCS$^+$ v.3.1 [2] に基づく。SPHINCS$^+$ は SPHINCS [8] の改良版として提案され [6, 7]，その後も NIST PQC 標準化プロジェクトで改良が行われ，v.3.1 となった。

以下では，Lighton-Micali Hash-Based Signatures，XMSS，SLH-DSA についてそれぞれ 7.3.1 節，7.3.2 節，7.3.3 節で述べられるが，どの順番で読んでも差し支えない。

7.3.1 Lighton-Micali Hash-Based Signatures

IRTF RFC 8554 [33] では，LMS, HSS が述べられている。LMS, HSS では Winternitz one-time signature に基づく LM-OTS が用いられる。LMS は LM-OTS とマークル木とを用いて構成され，この構造は LMS 木と呼ばれる。HSS は LMS 木の階層構造による署名方式である。LM-OTS, LMS, HSS にはそれぞれ，それらのアルゴリズムで用いられるハッシュ関数，パラメータセットに対応する長さ 4 Bytes の符号なし整数が割り当てられる。これは typecode と呼ばれる。

本節では，ハッシュ関数 $H : \mathcal{D} \to \mathcal{R}$ について，$\mathcal{D} = \bigcup_{i \geq 0} \{0,1\}^{8i}, \mathcal{R} = \{0,1\}^{8n}$ とする。すなわち，H は任意長の byte 系列を入力とし，長さ n Bytes の系列を出力する。なお，本節の表記は概ね [18] の表記法に基づいている。

7.3.1.1 LM-OTS

$w \in \{1, 2, 4, 8\}$ を Winternitz 係数の幅（bit 長）とする。p を LM-OTS を構成する長さ n Bytes の系列の個数とする。*type* を typecode とする。ハッシュ関数 H を用いて以下の関数が定義される。

$$H^i_{I,q,d}(x; j) := \begin{cases} x & i = 0 \text{ のとき} \\ H(I \| [q]_4 \| [d]_2 \| [i + j - 1]_1 \| H^{i-1}_{I,q,d}(x; j)) & i \geq 1 \text{ のとき} \end{cases}$$

ここで，I は長さ 16 Bytes の系列であり，LM-OTS が，LMS や HSS においてどのマークル木で使用されるかを表す。q は長さ 4 Bytes の整数であり，LM-OTS の公開鍵が対応するマークル木の葉を表す。

■鍵生成アルゴリズム　与えられた I, q に対応する秘密鍵と公開鍵の組を生成するアルゴリズムを以下に示す。

1. $x_0, x_1, \ldots, x_{p-1} \in \{0,1\}^{8n}$ を無作為に選択する。
2. $0 \leq i \leq p-1$ について，$y_i := H^{2^w-1}_{I,q,i}(x_i; 0)$ とする（図 7.2）。
3. $K := H(I \| [q]_4 \| [8080]_2 \| y_0 \| y_1 \| \cdots \| y_{p-1})$ とする。

秘密鍵は $(type, I, q, x_0, x_1, \ldots, x_{p-1})$ である。公開鍵は $[type]_4 \| I \| [q]_4 \| K$ である。

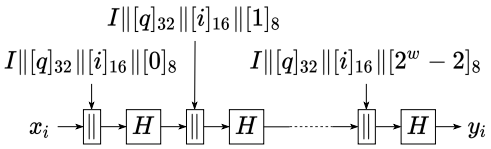


図 7.2: y_i の計算

■署名アルゴリズム　$\mathsf{Checksum} : \{0,1\}^{8n} \to \{0,1\}^{16}$ は以下のように定義される関数である。

$$\mathsf{Checksum}(S) := \Big(\sum_{i=0}^{8n/w-1} (2^w - 1 - d_i) \Big) \ll ls$$

ここで，$S = d_0 \| d_1 \| \cdots \| d_{8n/w-1}$ であり，$0 \le i \le 8n/w - 1$ について $d_i \in \{0,1\}^w$ である。上式で d_i は整数とみなされている。また，ls は n, w に応じて決まる整数であり，$16 - ls$ が w の倍数であり，かつ，$\mathsf{Checksum}(S)$ の 2 進数表記の長さが常に $16 - ls$ bits 以下となるよう定められる。

メッセージ M に対する署名アルゴリズムを以下に示す。

1. アルゴリズムのパラメータセットに応じて $type, n, p, w$ の値を定める。
2. $C \in \{0,1\}^{8n}$ を無作為に選択し，$Q := H(I\|[q]_4\|[\mathsf{8181}]_2\|C\|M)$，$c := \mathsf{Checksum}(Q)$ とする。
3. $Q\|c \in \{0,1\}^{wp}$ をそれぞれ長さ w bits のブロック $V_0, V_1, \ldots, V_{p-1}$ に分割する。
4. $0 \le i \le p-1$ について，$\sigma_i := H^{V_i}_{I,q,i}(x_i; 0)$ とする。ここで，V_i は整数とみなされる。

メッセージ M に対する署名は $\sigma := [type]_4 \| C \| \sigma_0 \| \sigma_1 \| \cdots \| \sigma_{p-1}$ である。

■検証アルゴリズム　鍵生成と署名のアルゴリズムより容易に導出される。詳細は [18, 33] を参照のこと。

7.3.1.2　LMS

LMS は LM-OTS と同じハッシュ関数を用いることが推奨されている。LMS のマークル木の各節点には番号が付される。根の番号は 1 であり，番号 ν の節点の左の子と右の子の番号はそれぞれ $2\nu, 2\nu + 1$ である。

h はマークル木の高さを表す。m はマークル木の各節点に対応する系列の byte 長を表し，これはハッシュ関数の出力長である。h, m の値は LMS の typecode によって定められる。

■鍵生成アルゴリズム

1. $I \in \{0,1\}^{128}$ を無作為に選択する。
2. $0 \le q \le 2^h - 1$ について，LM-OTS の公開鍵と秘密鍵の組 (pk^q, sk^q) を生成する。
3. マークル木の番号 r の節点に対応する系列 $T[r]$ は以下のように定義される。

$$T[r] := \begin{cases} H(I\|[r]_4\|[\mathsf{8282}]_2\|pk^{r-2^h}) & 2^h \le r \le 2^{h+1} - 1 \text{ のとき} \\ H(I\|[r]_4\|[\mathsf{8383}]_2\|T[2r]\|T[2r+1]) & 1 \le r \le 2^h - 1 \text{ のとき} \end{cases}$$

公開鍵は $[type]_4\|[otstype]_4\|I\|T[1]$ である。秘密鍵は $sk^0, sk^1, \ldots, sk^{2^h-1}$ である。ここで $type, otstype$ はそれぞれ LMS, LMS-OTS の typecode を表す。

■署名アルゴリズム　以下では $0 \le q \le 2^h - 1$ である。最初の署名では $q = 0$ とし，1 回の署名ごとに q の値を 1 だけ増やすことにより，LM-OTS の各秘密鍵を複数回使用しないようにしなければならない。

メッセージ M に対する署名アルゴリズムを以下に示す。

1. 秘密鍵 sk^q を用いて LM-OTS による M の署名 σ を計算する。
2. $0 \leq i \leq h-1$ について，$p_i := T[\lfloor (q+2^h)/2^i \rfloor \oplus 1]$ とする。

M の署名は $[q]_4 \| \sigma \| [type]_4 \| p_0 \| p_1 \| \cdots \| p_{h-1}$ である。$p_0, p_1, \ldots, p_{h-1}$ は LM-OTS の公開鍵 pk^q の認証パスである。

■検証アルゴリズム　鍵生成と署名のアルゴリズムより容易に導出される。詳細は [18, 33] を参照のこと。

7.3.1.3　HSS

HSS は 7.2.3 節で述べられたような LMS 木の階層構造による署名方式である。階層数 L は $1 \leq L \leq 8$ を満たす。

HSS は stateful な署名方式なので，メッセージの署名の際に最下層の LMS 木の秘密鍵が使い尽くされたとき，そのメッセージの署名で使用された L 個の LMS 木のうち，第 l 層から下のすべての LMS 木の秘密鍵が使い尽くされた最大の l を求める。$l = L-1$ のときは，新たな署名を作成しない。$l \leq L-2$ のとき，$0 \leq i \leq l$ について，秘密鍵が使い尽くされた第 i 層の LMS 木を破棄し，それぞれに替わる新たな LMS 木を使用して新しいメッセージへの署名を行う。

HSS の鍵生成，署名，検証の各アルゴリズムについての詳細は [18, 33] を参照のこと。

7.3.1.4　パラメータの設定と安全性

LMS の選択文書攻撃に対する存在偽造不能性（EUF-CMA）については，Katz [28] や Fluhrer [18] によりランダムオラクルモデルを仮定して示されており，また，Eaton [17] により量子ランダムオラクルモデルを仮定して示されている。なお，IRTF RFC 8554 [33] には，ハッシュ関数は第二原像攻撃に対する安全性を満たさなければならないと記されている。

NIST SP 800-208 [12] では，ハッシュ関数として SHA-256, SHA-256/192, SHAKE256/256, SHAKE256/192 を使用することが認可されている。ここで，SHA-256/192 は SHA-256 の出力の上位 192 bits を出力とするハッシュ関数である。SHAKE256/256, SHAKE256/192 はそれぞれ，出力長を 256 bits，192 bits とする SHAKE256 である。NIST SP 800-208 [12] と IRTF RFC 8554 [33] の両方に掲載されている SHA-256 を用いる場合の LM-OTS, LMS のパラメータセットの値の一覧をそれぞれ表 7.3, 7.4 に示す。

表 7.3: LM-OTS のパラメータセットと署名長（単位は Byte）

名称	n	w	p	ls	署名長
LMOTS_SHA256_N32_W1	32	1	265	7	8,516
LMOTS_SHA256_N32_W2	32	2	133	6	4,292
LMOTS_SHA256_N32_W4	32	4	67	4	2,180
LMOTS_SHA256_N32_W8	32	8	34	0	1,124

表 7.4: LMS のパラメータセット

名称	m	h
LMS_SHA256_M32_H5	32	5
LMS_SHA256_M32_H10	32	10
LMS_SHA256_M32_H15	32	15
LMS_SHA256_M32_H20	32	20
LMS_SHA256_M32_H25	32	25

7.3.2　XMSS: eXtended Merkle Signature Scheme

XMSS は [10, 24] で提案された方式の改良版 [25] に基づく署名方式であり，WOTS$^+$ と呼ばれる Winternitz one-time signature に基づく 1 回署名方式 [22] を用いる[*1]。

XMSS では三つの鍵付きハッシュ関数 $\mathsf{F}, \mathsf{H}, \mathsf{H}_{\mathrm{msg}}$ と擬似ランダム関数 R が用いられる。いずれも出力の byte 長は等しく，これを n とする。F の入力は byte 長 n の鍵と byte 長 n の系列である。H の入力は byte 長 n の鍵と byte

[*1] 7.3.3 節の SLH-DSA で用いられる 1 回署名方式とマークル木を用いた署名方式もそれぞれ WOTS$^+$，XMSS と呼ばれるが，アルゴリズムには相違点が存在する。

長 $2n$ の系列である。$\mathsf{H}_{\mathrm{msg}}$ の入力は byte 長 $3n$ の鍵と任意 byte 長の系列である。R の入力は byte 長 n の鍵と byte 長 32 の系列である。これらの関数は SHA-2 [38] または SHA-3 [39] を用いて定義される。例えば，$n = 32$ のとき，SHA-256 を用いて以下のように定義される。

$$\begin{aligned}\mathsf{F}(k, x) &:= \text{SHA-256}([0]_{32} \| k \| x)\\ \mathsf{H}(k, x) &:= \text{SHA-256}([1]_{32} \| k \| x)\\ \mathsf{H}_{\mathrm{msg}}(k, x) &:= \text{SHA-256}([2]_{32} \| k \| x)\\ \mathsf{R}(k, x) &:= \text{SHA-256}([3]_{32} \| k \| x)\end{aligned}$$

XMSS では，ハッシュ関数の呼び出しをランダム化するために，それぞれのハッシュ関数の呼び出しで，鍵とビットマスクが用いられる。これらは擬似ランダム関数を用いて生成され，入力として byte 系列の seed と長さ 32 Bytes のアドレス ADRS が与えられる。アドレスは 3 種あり，それぞれ OTS ハッシュアドレス，L 木アドレス，ハッシュ木アドレスと呼ばれる。それらの構造を図 7.3 に示す。

layer address	(4 Bytes)
tree address	(8 Bytes)
type = 0	(4 Bytes)
OTS address	(4 Bytes)
chain address	(4 Bytes)
hash address	(4 Bytes)
keyAndMask	(4 Bytes)

(a) OTS ハッシュアドレス

layer address	(4 Bytes)
tree address	(8 Bytes)
type = 1	(4 Bytes)
L-tree address	(4 Bytes)
tree height	(4 Bytes)
tree index	(4 Bytes)
keyAndMask	(4 Bytes)

(b) L 木アドレス

layer address	(4 Bytes)
tree address	(8 Bytes)
type = 2	(4 Bytes)
Padding = 0	(4 Bytes)
tree height	(4 Bytes)
tree index	(4 Bytes)
keyAndMask	(4 Bytes)

(c) ハッシュ木アドレス

図 7.3: アドレスの構造

7.3.2.1 WOTS$^+$

$w \in \{4, 16\}$ は Winternitz パラメータと呼ばれる。$\ell := \ell_1 + \ell_2$ は公開鍵，秘密鍵，署名を構成する byte 長 n の要素の個数を表す。ここで，

$$\ell_1 := \lceil 8n / \log_2 w \rceil, \quad \ell_2 := \lfloor \log_2(\ell_1(w-1)) / \log_2 w \rfloor + 1$$

である。

■チェイニング関数 チェイニング関数 chain の入力は，長さ n Bytes の系列 X，スタートインデクス i，ステップ数 s，長さ 32 Bytes のアドレス ADRS，長さ n Bytes のシード seed であり，以下のように定義される。

$$\mathsf{chain}(X, i, s, \mathsf{seed}, \mathrm{ADRS}) := \begin{cases} X & s = 0 \text{ のとき} \\ \texttt{NULL} & i + s \geq w \text{ のとき} \\ \mathsf{F}(Key, \mathsf{chain}(X, i, s-1, \mathsf{seed}, \mathrm{ADRS}) \oplus BM) & \text{それ以外のとき} \end{cases}$$

ここで，

$$Key := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [i+s-1]_4 \| [0]_4), \quad BM := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [i+s-1]_4 \| [1]_4)$$

である。なお，ADRS$'$ は ADRS の上位 24 Bytes であり，例えば，ADRS$' \| [i+s-1]_4 \| [0]_4$ は図 7.3a の ADRS の hash address, keyAndMask の値をそれぞれ，$[i+s-1]_4, [0]_4$ とすることを表している。

■鍵生成アルゴリズム　入力は ADRS, seed である。

1. $0 \le i \le \ell - 1$ について，$sk_i \in \{0,1\}^{8n}$ を無作為に選択する。
2. $0 \le i \le \ell - 1$ について，ADRS の chain address の値を $[i]_4$ とし，

$$pk_i := \mathsf{chain}(sk_i, 0, w-1, \mathsf{seed}, \mathrm{ADRS})$$

とする。この計算を図 7.4 に示す。この図で

$$Key_j := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [j]_4 \| [0]_4), \quad BM_j := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [j]_4 \| [1]_4)$$

である。

公開鍵は $pk := (pk_0, pk_1, \ldots, pk_{\ell-1})$ である。秘密鍵は $sk := (sk_0, sk_1, \ldots, sk_{\ell-1})$ である。

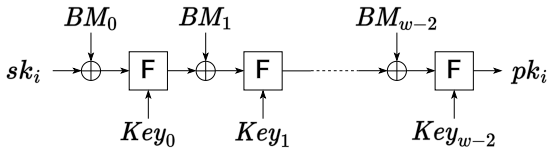


図 7.4: pk_i の計算

■署名アルゴリズム　入力は byte 長 n のメッセージ M，秘密鍵 sk，アドレス ADRS，シード seed である。

1. M をそれぞれ長さ $\log_2 w$ bits の ℓ_1 個のブロックに分割し，先頭から順に $M_0, M_1, \ldots, M_{\ell_1-1}$ とする。これらを整数とみなすと，$0 \le i \le \ell_1 - 1$ について，$M_i \in \{0, 1, \ldots, w-1\}$ である。
2. $C := \sum_{i=0}^{\ell_1-1} (w - 1 - M_i)$ とする。
3. $C \cdot 2^{8-(\ell_2 \log_2 w \bmod 8)}$ を長さ $\lceil (\ell_2 \log_2 w)/8 \rceil$ Bytes の系列とみなし，それぞれ長さ $\log_2 w$ bits の ℓ_2 個のブロックに分割し，先頭から順に $M_{\ell_1}, M_{\ell_1+1}, \ldots, M_{\ell-1}$ とする。
4. $0 \le i \le \ell - 1$ について，ADRS の chain address の値を i とし，

$$sig_i := \mathsf{chain}(sk_i, 0, M_i, \mathsf{seed}, \mathrm{ADRS})$$

とする。

メッセージ M に対する署名は $sig_0, sig_1, \ldots, sig_{\ell-1}$ である。

■検証アルゴリズム　鍵生成と署名のアルゴリズムより容易に導出される。詳細は [21] を参照のこと。

7.3.2.2　XMSS

XMSS はマークル木を用いた署名方式であり，公開鍵と秘密鍵の各組は完全二分木に対応付けられる。

XMSS のハッシュ木の構成のために，ランダム化ハッシュ関数 RH が導入されている。RH の入力は長さ n Bytes の $LEFT, RIGHT$，長さ n Bytes のシード seed，長さ 32 Bytes のアドレス ADRS であり，以下のように定義される。

$$\mathsf{RH}(LEFT, RIGHT, \mathsf{seed}, \mathrm{ADRS}) := \mathsf{H}(Key, (LEFT \oplus BM_0) \| (RIGHT \oplus BM_1))$$

ここで，

$$Key := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [0]_4), \quad BM_0 := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [1]_4), \quad BM_1 := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [2]_4)$$

である。なお，ADRS′ は ADRS の上位 28 Bytes であり，例えば，ADRS′$\|[0]_4$ は ADRS の図 7.3 の keyAndMask の値を $[0]_4$ とすることを表している。

秘密鍵の生成には [10] に示されているような擬似ランダム鍵生成法を用いることが許容されているが，その安全性は少なくとも XMSS の安全性と同等でなければならない。

■鍵生成アルゴリズム　鍵生成アルゴリズムではマークル木が構成され，その各葉には WOTS$^+$ の公開鍵が対応する。WOTS$^+$ の公開鍵に対して L 木と呼ばれるハッシュ木が構成され，その木の根のハッシュ値が XMSS のマークル木の葉に割り当てられる。L 木の高さ $j(\geq 0)$ の左から $i(\geq 0)$ 番目の節点を $\mathrm{Node}_{i,j}$ と表記する。L 木は以下にしたがって構成される。入力は WOTS$^+$ の公開鍵 $pk := (pk_0, pk_1, \ldots, pk_{\ell-1})$，L 木アドレス ADRS，シード seed である。

1. $0 \leq i \leq \ell - 1$ について，$\mathrm{Node}_{i,0} := pk_i$ とする。
2. $j \geq 0$ について，根が得られるまで以下にしたがって $\mathrm{Node}_{i,j+1}$ を計算する。なお，値の定義された $\mathrm{Node}_{i,j}$ の個数を ℓ' とする。
   (a) $0 \leq i < \lfloor \ell'/2 \rfloor$ について，$\mathrm{Node}_{i,j+1} := \mathsf{RH}(\mathrm{Node}_{2i,j}, \mathrm{Node}_{2i+1,j}, \mathsf{seed}, \mathrm{ADRS})$ とする。ここで，ADRS の tree height を $[j]_4$，tree index を $[i]_4$ とする。さらに，ℓ' が奇数のとき，$\mathrm{Node}_{\lfloor \ell'/2 \rfloor, j+1} := \mathrm{Node}_{\ell'-1,j}$ とする。
   (b) $j \leftarrow j + 1$ とする。

鍵生成アルゴリズムで構成されるマークル木の高さを h とすると，このマークル木には 2^h 個の葉が存在する。このマークル木に対応する 2^h 個の WOTS$^+$ の公開鍵，それらの L 木，さらに，このマークル木の計算に用いられる OTS ハッシュアドレス，L 木アドレス，ハッシュ木アドレスの layer address，tree address はすべて，それぞれ $[0]_4$，$[0]_8$ である。左から $k\,(\geq 0)$ 番目の葉に対応する OTS ハッシュアドレスの OTS address，L 木アドレスの L-tree address は $[k]_4$ である。

鍵生成アルゴリズムで構成されるマークル木の葉は対応する L 木の根である。葉以外の節点は L 木の節点と同じ方法で計算される。なお，このマークル木は完全二分木なので，上述の L 木の計算手続きで，ℓ' は常に偶数となる。

秘密鍵は，2^h 個の WOTS$^+$ の秘密鍵，次の署名に使用される WOTS$^+$ の秘密鍵に対応するマークル木の葉の番号 idx，署名されるメッセージのハッシュの計算に使用される SK_{PRF}，マークル木の根 $root$，seed である。公開鍵は，マークル木の根，seed である。ここで，SK_{PRF} と seed はこの鍵生成アルゴリズムで無作為に選択される長さ n Bytes の系列である。また，公開鍵には識別子 OID が付される。

■署名アルゴリズム　メッセージ M の署名は，署名に使用される WOTS$^+$ の秘密鍵の番号 idx，M のダイジェストの計算に使用される乱数 r，WOTS$^+$ による署名，マークル木の idx 番目の葉の認証パスからなる。

1. M のダイジェストを $M' := \mathsf{H}_{\mathrm{msg}}(r \| root \| [idx]_n, M)$ とする。ここで，$r := \mathsf{R}(SK_{\mathrm{PRF}}, [idx]_4)$ である。
2. WOTS$^+$ の idx 番目の秘密鍵を用いて M' に署名し，マークル木の idx 番目の葉の認証パスを計算する。

WOTS$^+$ の同じ秘密鍵が 2 回以上使用されないよう，idx は $idx \leftarrow idx + 1$ により更新される。

■検証アルゴリズム　鍵生成と署名のアルゴリズムより容易に導出される。詳細は [21] を参照のこと。

7.3.2.3　XMSS^{MT}

XMSS^{MT} は，7.2.3 節のマークル木の階層構造による署名方式に相当する。XMSS^{MT} 木はハイパー木と呼ばれ，d 層の XMSS 木からなる。ここで，XMSS 木は 7.3.2.2 節の鍵生成アルゴリズムで生成される L 木とマークル木からなる木を表す。第 $(d-1)$ 層と第 0 層はそれぞれ，XMSS^{MT} 木の根と葉に相当する。すべての XMSS 木の高さは等しく，Winternitz パラメータもすべて同じ値が用いられる。第 x 層の左から y 番目の XMSS 木の構成で使用される

OTS ハッシュアドレス，L 木アドレス，ハッシュ木アドレスの layer address と tree address は，それぞれ $[x]_4$，$[y]_4$ である。

XMSS^{MT} の鍵生成，署名，検証の各アルゴリズムについての詳細は [21] を参照のこと。

7.3.2.4 パラメータの設定と安全性

Kampanakis と Fluhrer [26] により，LMS と XMSS の比較が論じられている。

Hülsing [25] らは，XMSS について安全性証明を与え，選択文書攻撃に対する存在偽造不能性（EUF-CMA）を満たすことを鍵付きハッシュ関数 $\mathsf{F}, \mathsf{H}, \mathsf{H}_{\mathrm{msg}}$ と擬似ランダム関数 R の以下の安全性に帰着している。

- F が以下の性質を満たすこと
  - multi-function, multi-target second preimage resistance (MM-SPR)
  - すべての出力が 2 個以上の原像を持つこと
- H が MM-SPR を満たすこと
- $\mathsf{H}_{\mathrm{msg}}$ が multi-target extended target collision resistance (M-ETCR) を満たすこと
- R が擬似ランダム関数 (PRF) であること

ここで，MM-SPR, M-ETCR は，$\mathsf{F}, \mathsf{H}, \mathsf{H}_{\mathrm{msg}}$ の構成に用いられるハッシュ関数の第二原像攻撃に対する安全性に基づく性質である。一方，PRF は，秘密鍵入力を有するハッシュ関数が擬似ランダム関数であることを要求する。さらに，R による鍵とビットマスクの生成については，ハッシュ関数がランダムオラクルであることが仮定される。

IRTF RFC 8391 [21] では，上述の XMSS の安全性に関する結果に基づいて，$n = 32, 64$ のとき，それぞれ，256 bit 安全性，512 bit 安全性が提供されると記されている。また，量子計算機を用いた攻撃に対してはそれぞれ，128 bit 安全性，256 bit 安全性が提供されると記されている。

IRTF RFC 8391 [21] では，ハッシュ関数として SHA-256 を用いることが要求されているが，オプションとして SHAKE128/256, SHA-512, SHAKE256/512 を用いることが記されている。一方, NIST SP 800-208 では, SHA-256, SHA-256/192, SHAKE256/256, SHAKE256/192 を用いることが認可されている。NIST SP 800-208 [12] と IRTF RFC 8391 [21] の両方に掲載されている SHA-256 を用いる場合の WOTS$^+$, XMSS, XMSS^{MT} のパラメータセットの値の一覧をそれぞれ表 7.5, 7.6, 7.7 に示す。

表 7.5: WOTS$^+$ のパラメータセット

名称	n	w	ℓ
WOTSP-SHA2_256	32	16	67

表 7.6: XMSS のパラメータセットと署名長（単位は Byte）

名称	n	w	ℓ	h	署名長
XMSS-SHA2_10_256	32	16	67	10	2,500
XMSS-SHA2_16_256	32	16	67	16	2,692
XMSS-SHA2_20_256	32	16	67	20	2,820

7.3.3 SLH-DSA

SLH-DSA [40] は 7.2.3 節のマークル木の階層構造による署名方式に基づく stateless な署名方式である。SLH-DSA で用いられる 1 回署名方式とマークル木を用いた署名方式はそれぞれ WOTS$^+$ (Winternitz One-Time Signature Plus scheme)，XMSS (eXtended Merkle Signature Scheme) と呼ばれる[*2]。また，XMSS で構成されるマークル木は XMSS 木と呼ばれる。SLH-DSA が 7.2.3 節で述べられた方式と異なる点は，FORS (Forest of Random Subsets)

[*2] これらの名称は 7.3.2 節の XMSS の対応する署名方式の名称と同一であるが，アルゴリズムには相違点が存在する。

表 7.7: XMSS^{MT} のパラメータセットと署名長（単位は Byte）

名称	n	w	ℓ	h	d	署名長
XMSSMT-SHA2_20/2_256	32	16	67	20	2	4,963
XMSSMT-SHA2_20/4_256	32	16	67	20	4	9,251
XMSSMT-SHA2_40/2_256	32	16	67	40	2	5,605
XMSSMT-SHA2_40/4_256	32	16	67	40	4	9,893
XMSSMT-SHA2_40/8_256	32	16	67	40	8	18,469
XMSSMT-SHA2_60/3_256	32	16	67	60	3	8,392
XMSSMT-SHA2_60/6_256	32	16	67	60	6	14,824
XMSSMT-SHA2_60/12_256	32	16	67	60	12	27,688

と呼ばれるハッシュ関数に基づく数回 (few-time) 署名方式が導入されている点である。数回署名方式は，一組の公開鍵と秘密鍵の組を用いて，複数個のメッセージに署名できる。SLH-DSA では，メッセージは FORS を用いて署名され，FORS の公開鍵が hypertree と呼ばれる XMSS 木の階層構造による署名方式を用いて署名される。SLH-DSA は，数回署名方式を導入して署名可能な回数を増加させることにより，stateless であることを達成している。なお，WOTS$^+$，XMSS，hypertree，FORS は SLH-DSA の構成要素として使用されるのみであり，それぞれの単独での使用は許容されていない。

SLH-DSA の公開鍵は長さ n Bytes の 2 つの系列 **PK**.root と **PK**.seed である。**PK**.root は hypertree の最上層の XMSS 木の根である。**PK**.seed は無作為に選択される。SLH-DSA の秘密鍵は n Bytes の 2 つの系列 **SK**.seed と **SK**.prf であり，いずれも無作為に選択される。なお，NIST FIPS 205 [40] では，**PK**.seed, **SK**.seed, **SK**.prf の生成に SP 800-90A [4]，SP 800-90B [44]，SP 800-90C [5] で規定されているランダム bit 生成器を使用することが求められている。WOTS$^+$ と FORS のすべての秘密鍵は，**SK**.seed を用いて擬似ランダム関数により生成される。**SK**.prf は，メッセージダイジェストの計算に使用される乱数系列の生成に使用される。

SLH-DSA の署名では，メッセージダイジェストは上記の乱数系列を用いたランダム化されたハッシュ関数により生成され，そのメッセージダイジェストの一部を用いてメッセージの署名に用いる FORS の公開鍵と秘密鍵の組が選択される。

SLH-DSA では以下の関数が用いられる。

- $\mathbf{PRF}_{msg} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^* \to \mathbb{B}^n$ はメッセージダイジェストの計算に使用される乱数系列を生成する擬似ランダム関数である。
- $\mathbf{H}_{msg} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^* \to \mathbb{B}^m$ はメッセージダイジェストを計算するハッシュ関数である。
- $\mathbf{PRF} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^{32} \to \mathbb{B}^n$ は WOTS$^+$，FORS の秘密鍵を生成する擬似ランダム関数である。
- $\mathbf{T}_\ell : \mathbb{B}^n \times \mathbb{B}^{32} \times \mathbb{B}^{\ell n} \to \mathbb{B}^n$ は WOTS$^+$，XMSS 木，FORS で用いられるハッシュ関数である。

さらに，$\mathbf{T}_1, \mathbf{T}_2$ について，$\mathbf{F} := \mathbf{T}_1$, $\mathbf{H} := \mathbf{T}_2$ の表記が用いられる。

SLH-DSA では，図 7.5 に示す 7 種のアドレスが用いられる。どのアドレスも長さは 32 Bytes である。各アドレスの layer address と tree address は XMSS 木の階層構造で，ハッシュ関数がどの XMSS 木で用いられるかを表す。これに基づき，FORS 木アドレス，FORS 木根圧縮アドレス，FORS 鍵生成アドレスの layer address の値はすべて 0 と定められている。

layer address	(4 Bytes)
tree address	(12 Bytes)
type = 0	(4 Bytes)
key pair address	(4 Bytes)
chain address	(4 Bytes)
hash address	(4 Bytes)

(a) WOTS$^+$ ハッシュアドレス

layer address	(4 Bytes)
tree address	(12 Bytes)
type = 1	(4 Bytes)
key pair address	(4 Bytes)
0	(4 Bytes)
0	(4 Bytes)

(b) WOTS$^+$ 公開鍵圧縮アドレス

layer address	(4 Bytes)
tree address	(12 Bytes)
type = 2	(4 Bytes)
0	(4 Bytes)
tree height	(4 Bytes)
tree index	(4 Bytes)

(c) ハッシュ木アドレス

layer address = 0	(4 Bytes)
tree address	(12 Bytes)
type = 3	(4 Bytes)
key pair address	(4 Bytes)
tree height	(4 Bytes)
tree index	(4 Bytes)

(d) FORS 木アドレス

layer address = 0	(4 Bytes)
tree address	(12 Bytes)
type = 4	(4 Bytes)
key pair address	(4 Bytes)
0	(4 Bytes)
0	(4 Bytes)

(e) FORS 木根圧縮アドレス

layer address	(4 Bytes)
tree address	(12 Bytes)
type = 5	(4 Bytes)
key pair address	(4 Bytes)
chain address	(4 Bytes)
0	(4 Bytes)

(f) WOTS$^+$ 鍵生成アドレス

layer address = 0	(4 Bytes)
tree address	(12 Bytes)
type = 6	(4 Bytes)
key pair address	(4 Bytes)
0	(4 Bytes)
tree index	(4 Bytes)

(g) FORS 鍵生成アドレス

図 7.5: アドレスの構造

7.3.3.1 WOTS$^+$

WOTS$^+$ は Winternitz one-time signature に基づく 1 回署名方式である。WOTS$^+$ は 2 つのパラメータ n と lg_w を用いる。n はセキュリティパラメータであり，署名されるメッセージ，公開鍵，秘密鍵，署名を構成する系列の byte 長である。lg_w は Winternitz パラメータと呼ばれる正整数 w について $lg_w := \log_2 w$ と定義される。WOTS$^+$ では $lg_w = 4$ と定められており，$w = 16$ である。

WOTS$^+$ の公開鍵，秘密鍵，署名を構成する系列の個数は $len := len_1 + len_2$ で表される。ここで，

$$len_1 := \lceil 8n/lg_w \rceil, \quad len_2 := \lfloor \log_2(len_1(w-1))/lg_w \rfloor + 1$$

である。$lg_w = 4$ なので，$len_1 = 2n$，$len_2 = 3$，$len = 2n + 3$ である。

■チェイニング関数 チェイニング関数 chain の入力は，長さ n Bytes の系列 X，スタートインデクス i，ステップ数 s，**PK**.seed，WOTS$^+$ ハッシュアドレス **ADRS** であり，以下のように定義される。

1. $tmp \leftarrow X$ とする。
2. $i \leq j \leq i + s - 1$ について，**ADRS** の hash address を j とし，$tmp \leftarrow \mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, tmp)$ とする。
3. tmp を返す。

■鍵生成アルゴリズム　入力は **SK**.seed, **PK**.seed, WOTS$^+$ ハッシュアドレス **ADRS** である。なお，**ADRS** の chain address, hash address の値はいずれも 0 である。

1. $0 \leq i \leq len - 1$ について，**ADRS** の chain address の値を i とし，

$$sk_i \leftarrow \mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \text{skADRS}) \qquad pk_i \leftarrow \mathsf{chain}(sk_i, 0, w-1, \mathbf{PK}.\text{seed}, \mathbf{ADRS})$$

   とする。なお，skADRS は WOTS$^+$ 鍵生成アドレスであり，layer address, tree address, key pair address, chain addresss については **ADRS** と同じ値が用いられる。
2. $pk \leftarrow \mathbf{T}_{len}(\mathbf{PK}.\text{seed}, \text{wotspkADRS}, pk_0 \| \cdots \| pk_{len-1})$ とする。ここで，wotspkADRS は WOTS$^+$ 公開鍵圧縮アドレスであり，layer address, tree address, key pair address については **ADRS** と同じ値が用いられる。

公開鍵は pk である。秘密鍵は $sk := (sk_0, sk_1, \ldots, sk_{len-1})$ である。

■署名アルゴリズム　入力は byte 長 n のメッセージ M，**SK**.seed，**PK**.seed，WOTS$^+$ ハッシュアドレス **ADRS** である。**ADRS** の layer address, tree address, key pair address で指定される WOTS$^+$ の秘密鍵を用いて署名が生成される。なお，**ADRS** の chain address, hash address の値はいずれも 0 である。

1. M をそれぞれ長さ lg_w bits の len_1 個のブロックに分割し，先頭から順に $msg_0, msg_1, \ldots, msg_{len_1-1}$ とする。これらを整数とみなすと，$0 \leq i \leq len_1 - 1$ について，$msg_i \in \{0, 1, \ldots, w-1\}$ である。
2. $csum \leftarrow \sum_{i=0}^{len_1-1} (w - 1 - msg_i)$ とする。
3. $csum \cdot 2^{(8-(len_2 \cdot lg_w \bmod 8)) \bmod 8}$ を長さ $\lceil (len_2 \cdot lg_w)/8 \rceil$ Bytes の系列とみなし，それぞれ長さ lg_w bits の len_2 個のブロックに分割し，先頭から順に $msg_{len_1}, msg_{len_1+1}, \ldots, msg_{len-1}$ とする。
4. $0 \leq i \leq len - 1$ について，**ADRS** の chain address の値を i とし，

$$sk_i \leftarrow \mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \text{skADRS}) \qquad sig_i \leftarrow \mathsf{chain}(sk_i, 0, msg_i, \mathbf{PK}.\text{seed}, \mathbf{ADRS})$$

   とする。なお，skADRS は WOTS$^+$ 鍵生成アドレスであり，layer address, tree address, key pair address, chain addresss については **ADRS** と同じ値が用いられる。

メッセージ M に対する署名は $sig_0, sig_1, \ldots, sig_{len-1}$ である。

■検証アルゴリズム　SLH-DSA では WOTS$^+$ が単独で使用されることが想定されていないため，検証アルゴリズムは明示されておらず，Winternitz one-time signature の検証に必須の，メッセージと署名の組から対応する公開鍵の候補を計算するアルゴリズムが示されている。なお，このアルゴリズムは，鍵生成と署名のアルゴリズムより容易に導出される。詳細は NIST FIPS 205 [40] を参照のこと。

7.3.3.2　XMSS

XMSS はマークル木を用いた署名方式であり，WOTS$^+$ を用いて構成される。

■鍵生成アルゴリズム　XMSS では，WOTS$^+$ の公開鍵を各葉にもつ高さ h' のマークル木（XMSS 木）を構成することにより，公開鍵が生成される。XMSS 木の高さ $z\,(\geq 0)$ の左から $i\,(\geq 0)$ 番目の節点を $node_{i,j}$ と表記する。入力は **SK**.seed, **PK**.seed, **ADRS** である。

1. $0 \leq i \leq 2^{h'} - 1$ について，$node_{i,0} \leftarrow pk_i$ とする。ここで，pk_i は **SK**.seed, **PK**.seed を用いて計算される WOTS$^+$ の公開鍵である。なお，pk_i の計算に用いられるアドレスの layer address と tree address の値は **ADRS** のそれらと等しく，key pair address の値は $[i]_4$ である。

2. $1 \leq z \leq h'$ について，それぞれ，$0 \leq i \leq 2^{h'-z} - 1$ について，

$$node_{i,z} \leftarrow \mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, node_{2i,z-1} \| node_{2i+1,z-1})$$

とする。ここで，**ADRS** はハッシュ木アドレスであり，tree height の値は $[z]_4$，tree index の値は $[i]_4$ である。

公開鍵は XMSS 木の根 $node_{0,h'}$ であり，秘密鍵は $2^{h'}$ 個の WOTS$^+$ の秘密鍵である。

なお，XMSS の単独での使用が想定されていないことから，NIST FIPS 205 [40] では，鍵生成アルゴリズムは明示されておらず，XMSS 木の各節点を計算する再帰的アルゴリズムが示されている。

■署名アルゴリズム　SLH-DSA では，XMSS で署名されるメッセージは XMSS の公開鍵あるいは FORS の公開鍵のみである。入力は $M, \mathbf{SK}.\text{seed}, idx, \mathbf{PK}.\text{seed}, \mathbf{ADRS}$ である。M は長さ n Bytes のメッセージ，idx は M の署名に使用される WOTS$^+$ の鍵の key pair address である。

WOTS$^+$ の idx 番目の秘密鍵を用いて M に署名し，XMSS 木の idx 番目の葉の認証パスを計算する。

■検証アルゴリズム　NIST FIPS 205 [40] では，検証に必須の，メッセージと署名の組から対応する公開鍵の候補を計算するアルゴリズムが示されている。このアルゴリズムは鍵生成と署名のアルゴリズムより容易に導出されるので，詳細は NIST FIPS 205 [40] を参照のこと。

7.3.3.3　Hypertree

SLH-DSA では，hypertree と呼ばれる XMSS 木の階層構造が用いられる。hypertree は d 層の XMSS 木からなり，すべての XMSS 木の高さは等しい。第 $(d-1)$ 層と第 0 層はそれぞれ hypertree の根と葉に相当する。第 x 層の左から y 番目の XMSS 木の構成で使用される WOTS$^+$ ハッシュアドレス，WOTS$^+$ 公開鍵圧縮アドレス，WOTS$^+$ 鍵生成アドレス，ハッシュ木アドレスの layer address と tree address はそれぞれ $[x]_4$，$[y]_{12}$ である。

hypertree の公開鍵は第 $(d-1)$ 層の XMSS の公開鍵である。hypertree の署名，検証の各アルゴリズムについての詳細は NIST FIPS 205 [40] を参照のこと。

7.3.3.4　FORS

FORS は，数回署名方式 HORS [43] に基づく HORST [8] の改良版である。FORS は $k, t := 2^a$ をパラメータとし，長さ ka bits の系列に署名を行う。

■鍵生成アルゴリズム　入力は **SK**.seed, **PK**.seed, FORS 木アドレス **ADRS** である。なお，**ADRS** の layer address, tree address, key pair address は，生成された FORS の公開鍵の署名に用いられる WOTS$^+$ の鍵の生成に用いられるアドレスのそれらの値と等しい。

1. $0 \leq i \leq kt - 1$ について，

$$sk_i \leftarrow \mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \text{skADRS}) \qquad node_{i,0} \leftarrow \mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, sk_i)$$

とする。ここで，skADRS は FORS 鍵生成アドレスであり，layer address, tree address, key pair address については **ADRS** と同じ値が用いられ，tree index の値は $[i]_4$ である。また，**ADRS** の tree index の値は $[i]_4$ である。

2. $1 \leq z \leq a$ について，それぞれ，$0 \leq i \leq k \cdot 2^{a-z} - 1$ について，

$$node_{i,z} \leftarrow \mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, node_{2i,z-1} \| node_{2i+1,z-1})$$

とする。ここで，**ADRS** の tree height の値は $[z]_4$，tree index の値は $[i]_4$ である。

3. $pk \leftarrow \mathbf{T}_k(\mathbf{PK}.\mathrm{seed}, \mathrm{forspkADRS}, node_{0,a} \| \cdots \| node_{k-1,a})$ とする。ここで，forspkADRS は FORS 木根圧縮アドレスであり，layer address, tree address, key pair address については **ADRS** と同じ値が用いられる。

このアルゴリズムにより，$node_{0,a}, node_{1,a}, \ldots, node_{k-1,a}$ を根とする k 個のマークル木が構成されている。公開鍵は pk である。秘密鍵は $sk_0, sk_1, \ldots, sk_{kt-1}$ である。

■署名アルゴリズム 長さ $k \cdot a$ bits のメッセージダイジェスト md をそれぞれ長さ a bits の k 個のブロック $md_0, md_1, \ldots, md_{k-1}$ に分割する。すなわち，$md = md_0 \| md_1 \| \cdots \| md_{k-1}$ である。さらに，md_i を 2 進数表記の非負整数とみなす。md の署名は $sk_{0 \cdot t + md_0}, sk_{1 \cdot t + md_1}, \ldots, sk_{(k-1)t + md_{k-1}}$ と，$0 \leq i \leq k-1$ について，$node_{i,a}$ を根とするマークル木の $node_{i \cdot t + md_i, 0}$ の認証パスである。詳細は NIST FIPS 205 [40] を参照のこと。

■検証アルゴリズム NIST FIPS 205 [40] では，検証に必須の，メッセージと署名の組から対応する公開鍵の候補を計算するアルゴリズムが示されている。詳細は NIST FIPS 205 [40] を参照のこと。

7.3.3.5 SLH-DSA

前節までの構成要素を用いて SLH-DSA の署名が構成される。SLH-DSA のパラメータは以下のとおりである。

- セキュリティパラメータ n（単位は Byte）
- hypertree のパラメータ $h, d, h'(= h/d)$
- FORS のパラメータ a, k
- WOTS$^+$ のパラメータ lg_w
- メッセージダイジェストの byte 長 $m = \lceil (h - h')/8 \rceil + \lceil h'/8 \rceil + \lceil (k \cdot a)/8 \rceil$

■鍵生成アルゴリズム $\mathbf{SK}.\mathrm{seed}, \mathbf{SK}.\mathrm{prf} \in \mathbb{B}^n$ はいずれも無作為に選択される。$\mathbf{PK}.\mathrm{seed} \in \mathbb{B}^n$ は無作為に選択される。$\mathbf{PK}.\mathrm{root} \in \mathbb{B}^n$ は hypertree の第 $(d-1)$ 層の XMSS 木の根である。秘密鍵は $\mathbf{SK}.\mathrm{seed}, \mathbf{SK}.\mathrm{prf}, \mathbf{PK}.\mathrm{seed}, \mathbf{PK}.\mathrm{root}$ である。公開鍵は $\mathbf{PK}.\mathrm{seed}, \mathbf{PK}.\mathrm{root}$ である。したがって，秘密鍵，公開鍵のサイズはそれぞれ，$4n$ Bytes，$2n$ Bytes である。

■署名アルゴリズム メッセージ M の署名は以下のように生成される。

1. $R := \mathbf{PRF}_{msg}(\mathbf{SK}.\mathrm{prf}, opt\_rand, M)$ とする。ここで，opt_rand を $\mathbb{B}^n$ の乱数とすることがデフォルトとされており，特にサイドチャネル攻撃が懸念される場合については強く推奨されているが，乱数生成器が利用可能でない場合は $opt\_rand = \mathbf{PK}.\mathrm{seed}$ とすることが許容されている。
2. $digest := \mathbf{H}_{msg}(R, \mathbf{PK}.\mathrm{seed}, \mathbf{PK}.\mathrm{root}, M)$ とする。$digest$ の最初の $\lceil (k \cdot a)/8 \rceil$ Bytes，次の $\lceil (h - h')/8 \rceil$ Bytes，その次の $\lceil h'/8 \rceil$ Bytes をそれぞれ md，整数 idx_{tree} の 2 進数表記，整数 idx_{leaf} の 2 進数表記とする。
3. hypertree の第 0 層の左から idx_{tree} 番目の XMSS 木の左から idx_{leaf} 番目の葉に対応する FORS の鍵を用いて md の先頭 $k \cdot a$ bits に対する署名を生成する。
4. 上の署名で用いられた FORS の公開鍵への hypertree による署名を生成する。

M の署名は R，md への FORS による署名，md への署名の検証に用いられる FORS の公開鍵への hypertree による署名からなる。したがって，署名のサイズは $(1 + k(a+1) + h + d \cdot len)n$ Bytes である。

SLH-DSA では，署名アルゴリズムに与えられるメッセージ M を署名対象の内容から生成する二つの方法が定められている。これらは pure 版，pre-hash 版と呼ばれている。署名アルゴリズムに対して，pure 版ではコンテクストと署名対象の内容とが与えられ，pre-hash 版ではコンテクストと署名対象の内容のハッシュ値とが与えられる。なお，コンテクストは長さが高々 255 Bytes の系列であり，デフォルトでは空列である。詳細については NIST FIPS 205 [40]

を参照のこと。

■検証アルゴリズム　署名アルゴリズムより容易に導出されるので，詳細については NIST FIPS 205 [40] を参照のこと。

7.3.3.6　パラメータの設定と安全性

SLH-DSA については，表 7.8 の 12 個のパラメータセットが示されている。この表の最左欄の名称は，使用されるハッシュ関数とセキュリティパラメータ n の bit 長を単位とした値を示している。さらに，s と f はそれぞれ，署名サイズ，計算時間が小さくなるよう定められたパラメータセットであることを示している。また，安全性レベルは NIST PQC 標準化プロジェクトの Call for Proposals に記された安全性強度のカテゴリである。これらのパラメータセットは，一組の公開鍵と秘密鍵により高々 2^{64} 個のメッセージが署名される場合の選択文書攻撃に対する存在偽造不能性を考慮して定められている。

表 7.8: SLH-DSA のパラメータセット。公開鍵長，署名長の単位は Byte である。

名称	n	h	d	h'	a	k	lg_w	m	安全性レベル	公開鍵長	署名長
SLH-DSA-SHA2-128s SLH-DSA-SHAKE-128s	16	63	7	9	12	14	4	30	レベル 1	32	7,856
SLH-DSA-SHA2-128f SLH-DSA-SHAKE-128f	16	66	22	3	6	33	4	34	レベル 1	32	17,088
SLH-DSA-SHA2-192s SLH-DSA-SHAKE-192s	24	63	7	9	14	17	4	39	レベル 3	48	16,224
SLH-DSA-SHA2-192f SLH-DSA-SHAKE-192f	24	66	22	3	8	33	4	42	レベル 3	48	35,664
SLH-DSA-SHA2-256s SLH-DSA-SHAKE-256s	32	64	8	8	14	22	4	47	レベル 5	64	29,792
SLH-DSA-SHA2-256f SLH-DSA-SHAKE-256f	32	68	17	4	9	35	4	49	レベル 5	64	49,856

Hülsing と Kudinov [23] は，SPHINCS$^+$ が選択文書攻撃に対する存在偽造不能性 (EUF-CMA) を満たすことをハッシュ関数 $\mathbf{T}_\ell$，$\mathbf{H}_{msg}$，擬似ランダム関数 $\mathbf{PRF}, \mathbf{PRF}_{msg}$ の以下の安全性に帰着している。

- $\mathbf{T}_\ell$ が以下の性質を満たすこと
  - single-function, multi-target collision resistance (SM-TCR)
  - single-function, multi-target preimage resistance (SM-PRE)
  - single-function, multi-target decisional second preimage resistance (SM-DSPR)
  - single-function, multi-target undetectability (SM-UD)
- $\mathbf{H}_{msg}$ が interleaved target subset resilience (ITSR) を満たすこと
- $\mathbf{PRF}, \mathbf{PRF}_{msg}$ が擬似ランダム関数 (PRF) であること

ここで，SM-TCR, SM-DSPR, ITSR は，$\mathbf{T}_\ell, \mathbf{H}_{msg}$ の構成に用いられるハッシュ関数の第二原像攻撃に対する安全性に基づく性質であり，SM-PRE は原像攻撃に対する安全性に基づく性質である。一方，SM-UD, PRF は，秘密鍵入力を有するハッシュ関数が擬似ランダム関数であることを要求する。

Barbosa ら [3] は，SPHINCS$^+$ で用いられている XMSS について，コンピュータで検証された安全性証明を与えて

いる。

さらに，NIST FIPS 205 [40] には，SLH-DSA の実装をサイドチャネル攻撃 [27] や故障攻撃 [1, 11, 19, 45] から保護するための注意が払われなければならないことが記されている。

7.3.3.7 ハッシュ関数の実現法

SLH-DSA の関数はすべて，SHAKE256，SHA-2 のうちのいずれかを用いて構成される。これらの構成は，SPHINCS$^+$ で simple な実現と呼ばれる構成であり，7.2.4 節で述べられたビットマスクは用いられていない。

SHAKE256 を用いた構成は以下のとおりである。

$$\begin{aligned}
\mathbf{H}_{msg}(R, \mathbf{PK}.\text{seed}, \mathbf{PK}.\text{root}, M) &:= \text{SHAKE256}(R \| \mathbf{PK}.\text{seed} \| \mathbf{PK}.\text{root} \| M, 8m) \\
\mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \mathbf{ADRS}) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \| \mathbf{ADRS} \| \mathbf{SK}.\text{seed}, 8n) \\
\mathbf{PRF}_{msg}(\mathbf{SK}.\text{prf}, \mathit{opt_rand}, M) &:= \text{SHAKE256}(\mathbf{SK}.\text{prf} \| \mathit{opt_rand} \| M, 8n) \\
\mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_1) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \| \mathbf{ADRS} \| M_1, 8n) \\
\mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_2) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \| \mathbf{ADRS} \| M_2, 8n) \\
\mathbf{T}_\ell(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_\ell) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \| \mathbf{ADRS} \| M_\ell, 8n)
\end{aligned}$$

安全性レベル 1 に対する SHA-2 を用いた構成は以下のとおりである。

$$\begin{aligned}
\mathbf{H}_{msg}(R, \mathbf{PK}.\text{seed}, \mathbf{PK}.\text{root}, M) &:= \text{MGF1-SHA-256}(R \| \mathbf{PK}.\text{seed} \| \text{SHA-256}(R \| \mathbf{PK}.\text{seed} \| \mathbf{PK}.\text{root} \| M, m)) \\
\mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \mathbf{ADRS}) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| \mathbf{SK}.\text{seed})) \\
\mathbf{PRF}_{msg}(\mathbf{SK}.\text{prf}, \mathit{opt_rand}, M) &:= \mathsf{Trunc}_n(\text{HMAC-SHA-256}(\mathbf{SK}.\text{prf} \| \mathit{opt_rand} \| M)) \\
\mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_1) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| M_1)) \\
\mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_2) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| M_2)) \\
\mathbf{T}_\ell(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_\ell) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| M_\ell))
\end{aligned}$$

安全性レベル 3, 5 に対する SHA-2 を用いた構成は以下のとおりである。

$$\begin{aligned}
\mathbf{H}_{msg}(R, \mathbf{PK}.\text{seed}, \mathbf{PK}.\text{root}, M) &:= \text{MGF1-SHA-512}(R \| \mathbf{PK}.\text{seed} \| \text{SHA-512}(R \| \mathbf{PK}.\text{seed} \| \mathbf{PK}.\text{root} \| M, m)) \\
\mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \mathbf{ADRS}) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| \mathbf{SK}.\text{seed})) \\
\mathbf{PRF}_{msg}(\mathbf{SK}.\text{prf}, \mathit{opt_rand}, M) &:= \mathsf{Trunc}_n(\text{HMAC-SHA-512}(\mathbf{SK}.\text{prf} \| \mathit{opt_rand} \| M)) \\
\mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_1) &:= \mathsf{Trunc}_n(\text{SHA-256}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 64-n) \| \mathbf{ADRS}^c \| M_1)) \\
\mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_2) &:= \mathsf{Trunc}_n(\text{SHA-512}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 128-n) \| \mathbf{ADRS}^c \| M_2)) \\
\mathbf{T}_\ell(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_\ell) &:= \mathsf{Trunc}_n(\text{SHA-512}(\mathbf{PK}.\text{seed} \| \mathsf{toByte}(0, 128-n) \| \mathbf{ADRS}^c \| M_\ell))
\end{aligned}$$

ここで，MGF1-SHA-256，MGF1-SHA-512 は RFC 8017 [37] の Appendix B.2.1 に記載されている MGF1 であり，HMAC-SHA-256，HMAC-SHA-512 は FIPS 198-1 [41] の HMAC である。また，$\mathsf{Trunc}_l(x)$ は byte 系列 x の左端から l Bytes を出力する関数であり，$\mathsf{toByte}(z, l)$ は整数 z を長さ l Bytes の byte 系列に変換する関数である。さらに，$\mathbf{ADRS}^c$ は $\mathbf{ADRS}$ の layer address，tree address，type をそれぞれ 1 Byte，8 Bytes，1 Byte に短縮した長さ 22 Bytes のアドレスである。

SPHINCS$^+$ では，当初，SHA-2 を用いた実現で SHA-256 のみが用いられていたが，SHA-256 を用いた実現では安全性のレベル 5 が達成できないことを示す攻撃 [42] が示されたことから，SPHINCS$^+$ v.3.1 では，安全性レベル 3, 5 について，$\mathbf{H}_{msg}$, $\mathbf{PRF}_{msg}$, $\mathbf{H}$, $\mathbf{T}_\ell$ が SHA-512 を用いて実現されることとなり，SLH-DSA でもそれに従っている。

7.4 ハッシュ関数に基づく署名技術に関するまとめ

本章では，ハッシュ関数に基づく署名技術として，Lighton-Micali hash-based signatures, XMSS, SLH-DSA を取り上げた。これらはいずれも 7.2 節で述べた代表的なハッシュ関数に基づく署名方式に基づく構造を有する。Lighton-

Micali hash-based signatures [33] と XMSS [21] は NIST の推奨アルゴリズムであり [12]，SLH-DSA は NIST PQC 標準化プロジェクトで選出された SPHINCS$^+$[2] に基づく標準アルゴリズムである。

ハッシュ関数に基づく署名技術の安全性はハッシュ関数の第二原像攻撃に対する安全性に依存しているが，XMSS, SLH-DSA については，秘密鍵入力を有するハッシュ関数が擬似ランダム関数であることにも依存する。さらに，ビットマスクの生成についてはハッシュ関数がランダムオラクルであることが仮定される。また，偽造攻撃の計算量は，ハッシュ関数がランダムオラクルであることを仮定して見積もられている。

ハッシュ関数に基づく署名技術については，stateful であること，すなわち，各メッセージの署名に用いられる 1 回署名の秘密鍵を 2 回以上使用することのないよう管理しなければならないことが問題であった。Lighton-Micali hash-based signatures と XMSS はいずれもハッシュ関数に基づく stateful な署名方式であり，それらを推奨アルゴリズムとする NIST SP 800-208 [12] には，ハッシュ関数に基づく stateful な署名方式は一般的な使用には適するものでなく，近い将来に実装が必要であり，その実装が長期間の使用を予定されており，かつ，使用開始後に他の署名方式への移行が実用的でないような応用での使用が意図されていると述べられている。

SLH-DSA は XMSS の設計で得られた知見に基づいて設計されており，HSS, XMSS^{MT} と同様の構造を有するが，各メッセージの署名に一つの秘密鍵で数回署名可能な FORS を用いることによって署名可能な回数を増加させることにより，stateless であることを達成している。

第 7 章の参照文献

[1] D. Amiet, L. Leuenberger, A. Curiger, P. Zbinden. FPGA-based SPHINCS$^+$ Implementations: Mind the Glitch. DSD. IEEE, 2020, pp. 229–237.

[2] J.-P. Aumasson et al. SPHINCS$^+$ Submission to the NIST post-quantum project, v.3.1. `https://sphincs.org/data/sphincs+-r3.1-specification.pdf`. 2022-06. (2024-03-08 閲覧).

[3] M. Barbosa, F. Dupressoir, B. Grégoire, A. Hülsing, M. Meijers, P.-Y. Strub. Machine-Checked Security for XMSS as in RFC 8391 and SPHINCS$^+$. CRYPTO (5). Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 421–454.

[4] E. Barker, J. Kelsey. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. NIST SP 800-90A Rev. 1, `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf`. 2015-06.

[5] E. Barker, J. Kelsey, K. McKay, A. Roginsky, M. S. Turan. Recommendation for Random Bit Generator (RBG) Constructions. NIST SP 800-90C (4th public draft), `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90C.4pd.pdf`. 2024-07. (2025-02-17 閲覧).

[6] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, P. Schwabe. The SPHINCS$^+$ Signature Framework. CCS. ACM, 2019, pp. 2129–2146.

[7] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, P. Schwabe. The SPHINCS+ Signature Framework. Cryptology ePrint Archive, Paper 2019/1086. 2019. `https://eprint.iacr.org/2019/1086`.

[8] D. J. Bernstein et al. SPHINCS: Practical Stateless Hash-Based Signatures. EUROCRYPT (1). Vol. 9056. Lecture Notes in Computer Science. Springer, 2015, pp. 368–397.

[9] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche. Sponge functions. ECRYPT Hash Workshop. 2007.

[10] J. Buchmann, E. Dahmen, A. Hülsing. XMSS – A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions. PQCrypto. Vol. 7071. Lecture Notes in Computer Science. Springer, 2011, pp. 117–129.

[11] L. Castelnovi, A. Martinelli, T. Prest. Grafting Trees: A Fault Attack Against the SPHINCS Framework. PQCrypto. Vol. 10786. Lecture Notes in Computer Science. Springer, 2018, pp. 165–184.

[12] D. Cooper, D. Apon, Q. Dang, M. Davidson, M. Dworkin, C. Miller. Recommendation for Stateful Hash-Based Signature Schemes. NIST SP 800-208, `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-208.pdf`. 2020-10.

[13] E. Dahmen, K. Okeya, T. Takagi, C. Vuillaume. Digital Signatures Out of Second-Preimage Resistant Hash Functions. PQCrypto. Vol. 5299. Lecture Notes in Computer Science. Springer, 2008, pp. 109–123.

[14] I. Damgård. A Design Principle for Hash Functions. CRYPTO. Vol. 435. Lecture Notes in Computer Science. Springer, 1989, pp. 416–427.

[15] W. Diffie, M. E. Hellman. New directions in cryptography. IEEE Trans. Inf. Theory. Vol. 22, Num. 6 (1976), pp. 644–654.

[16] C. Dods, N. P. Smart, M. Stam. Hash Based Digital Signature Schemes. IMACC. Vol. 3796. Lecture Notes in Computer Science. Springer, 2005, pp. 96–115.

[17] E. Eaton. Leighton-Micali Hash-Based Signatures in the Quantum Random-Oracle Model. SAC. Vol. 10719. Lecture Notes in Computer Science. Springer, 2017, pp. 263–280.

[18] S. Fluhrer. Further Analysis of a Proposed Hash-Based Signature Standard. Cryptology ePrint Archive, Paper 2017/553. 2017. `https://eprint.iacr.org/2017/553`.

[19] A. Genêt. On Protecting SPHINCS+ Against Fault Attacks. IACR Trans. Cryptogr. Hardw. Embed. Syst. Vol. 2023, Num. 2 (2023), pp. 80–114.

[20] L. K. Grover. A fast quantum mechanical algorithm for database search. STOC. ACM, 1996, pp. 212–219.

[21] A. Huelsing, D. Butin, S.-L. Gazdag, J. Rijneveld, A. Mohaisen. XMSS: eXtended Merkle Signature Scheme. RFC 8391, `https://www.rfc-editor.org/info/rfc8391`. 2018-05.

[22] A. Hülsing. W-OTS$^+$ – Shorter Signatures for Hash-Based Signature Schemes. AFRICACRYPT. Vol. 7918. Lecture Notes in Computer Science. Springer, 2013, pp. 173–188.

[23] A. Hülsing, M. A. Kudinov. Recovering the Tight Security Proof of SPHINCS$^+$. ASIACRYPT (4). Vol. 13794. Lecture Notes in Computer Science. Springer, 2022, pp. 3–33.

[24] A. Hülsing, L. Rausch, J. Buchmann. Optimal Parameters for XMSS MT. CD-ARES Workshops. Vol. 8128. Lecture Notes in Computer Science. Springer, 2013, pp. 194–208.

[25] A. Hülsing, J. Rijneveld, F. Song. Mitigating Multi-target Attacks in Hash-Based Signatures. Public Key Cryptography (1). Vol. 9614. Lecture Notes in Computer Science. Springer, 2016, pp. 387–416.

[26] P. Kampanakis, S. Fluhrer. LMS vs XMSS: Comparion of two Hash-Based Signature Standards. Cryptology ePrint Archive, Paper 2017/349. 2017. `https://eprint.iacr.org/2017/349`.

[27] M. J. Kannwischer, A. Genêt, D. Butin, J. Krämer, J. Buchmann. Differential Power Analysis of XMSS and SPHINCS. COSADE. Vol. 10815. Lecture Notes in Computer Science. Springer, 2018, pp. 168–188.

[28] J. Katz. Analysis of a Proposed Hash-Based Signature Standard. SSR. Vol. 10074. Lecture Notes in Computer Science. Springer, 2016, pp. 261–273.

[29] J. Kelsey, B. Schneier. Second Preimages on n-Bit Hash Functions for Much Less than 2^n Work. EUROCRYPT. Vol. 3494. Lecture Notes in Computer Science. Springer, 2005, pp. 474–490.

[30] P. Lafrance, A. Menezes. On the security of the WOTS-PRF signature scheme. Adv. Math. Commun. Vol. 13, Num. 1 (2019), pp. 185–193.

[31] L. Lamport. Constructing digital signatures from a one-way function. SRI International Technical Report, CSL-98. 1979-10.

[32] F. T. Leighton, S. Micali. Large provably fast and secure digital signature schemes based on secure hash functions. 1995-07. US Patent 5,432,852.

[33] D. McGrew, M. Curcio, S. Fluhrer. Leighton-Micali Hash-Based Signatures. RFC 8554, `https://www.rfc-editor.org/info/rfc8554`. 2019-04.

[34] R. Merkle. Secrecy, Authentication, and Public Key Systems. PhD thesis. Stanford University, 1979. `https://www.merkle.com/papers/Thesis1979.pdf`.

[35] R. C. Merkle. A Certified Digital Signature. CRYPTO. Vol. 435. Lecture Notes in Computer Science. Springer, 1989, pp. 218–238.

[36] R. C. Merkle. One Way Hash Functions and DES. CRYPTO. Vol. 435. Lecture Notes in Computer Science. Springer, 1989, pp. 428–446.

[37] K. Moriarty, B. Kaliski, J. Jonsson, A. Rusch. PKCS #1: RSA Cryptography Specifications Version 2.2. RFC 8017, https://www.rfc-editor.org/info/rfc8017. 2016-11.

[38] NIST. Secure Hash Standard (SHS). NIST FIPS 180-4, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf. 2015-08.

[39] NIST. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. NIST FIPS 202, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf. 2015-08.

[40] NIST. Stateless Hash-Based Digital Signature Standard. NIST FIPS 205, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf. 2024-08.

[41] NIST. The Keyed-Hash Message Authentication Code (HMAC). NIST FIPS 198-1, https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.198-1.pdf. 2008-07.

[42] R. A. Perlner, J. Kelsey, D. A. Cooper. Breaking Category Five SPHINCS$^+$ with SHA-256. PQCrypto. Vol. 13512. Lecture Notes in Computer Science. Springer, 2022, pp. 501–522.

[43] L. Reyzin, N. Reyzin. Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying. ACISP. Vol. 2384. Lecture Notes in Computer Science. Springer, 2002, pp. 144–153.

[44] M. S. Turan, E. Barker, J. Kelsey, K. McKay, M. Baish, M. Boyle. Recommendation for the Entropy Sources Used for Random Bit Generation. NIST SP 800-90B, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf. 2018-01.

[45] A. Wagner, V. Wesselkamp, F. Oberhansl, M. Schink, E. Strieder. Faulting Winternitz One-Time Signatures to Forge LMS, XMSS, or SPHINCS$^+$ Signatures. PQCrypto. Vol. 14154. Lecture Notes in Computer Science. Springer, 2023, pp. 658–687.

耐量子計算機暗号の研究動向調査報告書
[CRYPTREC TR-****-****]

発行日：20**年 3 月 31 日（第 1 版）
発行者
〒184-8795
- 東京都小金井市貫井北町四丁目 2 番 1 号

国立研究開発法人情報通信研究機構
（サイバーセキュリティ研究所　セキュリティ基盤研究室）
NATIONAL INSTITUTE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
4-2-1 NUKUI-KITAMACHI, KOGANEI
TOKYO, 184-8795 JAPAN
〒113-6591
東京都文京区本駒込二丁目 2 8 番 8 号
独立行政法人情報処理推進機構
（セキュリティセンター セキュリティ技術評価部 暗号グループ）
INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN
2-28-8 HONKOMAGOME, BUNKYO-KU
TOKYO, 113-6591 JAPAN

資料３－５

量子コンピュータが共通鍵暗号の安全性に及ぼす影響（外部評価）

1　背景

（1）2019 年度、暗号技術調査ワーキンググループ（暗号解析評価）における活動として「量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価」を外部評価により実施し、本報告書（以下、「2019 年度外部評価報告書（CRYPTREC EX-2901-2019)」という）を CRYPTREC の技術調査報告書として公開した。

（2）2022 年度、暗号技術調査ワーキンググループ（耐量子計算機暗号）における活動として「CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）（CRYPTREC GL-2004-2022）」と「耐量子計算機暗号の研究動向調査報告書（CRYPTREC TR-2001-2022）」（以下、「PQC ガイドライン等」という）を作成した。

（3）2019 年度外部評価報告書が公開されていることを踏まえ、PQC ガイドライン等では PQC として共通鍵暗号を含まず、公開鍵暗号のみを示す言葉としている。つまり、PQC ガイドライン等では共通鍵暗号の耐量子安全性については触れていない。

（4）2023 年度第 2 回暗号技術評価委員会で共通鍵暗号の耐量子安全性に関する議論が行われた。具体的には、共通鍵暗号の耐量子安全性に関する技術動向調査を実施し、2019 年度外部評価報告書に調査内容を反映させる形で更新することはどうか、ということについて議論が行われた。本件について、事務局で対応を検討することが確認された。

（5）2024 年度第 1 回暗号技術評価委員会において、量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価を外部評価で実施し、本結果を 2019 年度外部評価報告書に反映させる形で更新することが承認された。

2　実施概要

細山田 光倫 様（日本電信電話株式会社）に外部評価を依頼した。選出理由と依頼内容は次のとおり。

（1）選出理由

共通鍵暗号の耐量子安全性に関する広い知見をお持ちであり、当該分野に関する数多くの実績をお持ちであるとともに、2019 年度外部評価報告書の執筆者であるため。

（2）依頼内容

量子コンピュータが共通鍵暗号の安全性に及ぼす影響について、公開されている解析手法やその影響範囲などについてまとめ、考察などを行い、2019 年度外部評価報告書に最新の情報を反映させて、報告書（以下、「2024 年度外部評価報告書」という）を作成する。

3　外部評価報告書の概要【報告事項】

（1）目次

2024 年度外部評価報告書の目次は表 1 のとおり。表内において、2019 年度版外部評価報告書から大きく加筆・修正した箇所を青字、追加された箇所を赤字で示す。

表 1　2024 年度外部評価報告書の目次

章	章タイトル	概要
1	はじめに	導入、2019 年度版との差異
2	準備	1. Grover のアルゴリズム 2. Simon のアルゴリズム
3	攻撃のモデル： 古典クエリと量子クエリ	1. 古典攻撃モデル 2. Q1 モデル（古典クエリ攻撃モデル） 3. Q2 モデル（量子クエリ攻撃モデル） 4. Q1 モデルと Q2 モデルの比較 5. ハッシュ関数への攻撃のモデル
4	攻撃コスト評価方法に 関する議論	1. 古典的衝突探索と誕生日のパラドクス 2. 最初の量子衝突探索アルゴリズム：BHT 3. BHT のアルゴリズムの効率性をめぐる議論 4. 量子ビット数の観点で効率的なアルゴリズム：CNS 5. その他の議論
5	汎用量子攻撃	1. Grover のアルゴリズムによる鍵回復・原像攻撃 2. 衝突探索および関連する問題 3. 多重原像探索 4. タイムメモリトレードオフとレインボーテーブル 5. ノストラダムス攻撃 6. 汎用量子攻撃の具体的なコスト
6	量子クエリ攻撃（Q2）	1. Even-Mansour（EM）暗号への鍵回復攻撃 2. Feistel 暗号（Luby-Rackoff 構成）への識別攻撃 3. Crypto 2016 における Kaplan らの結果 4. Grover のアルゴリズムと Simon のアルゴリズムの組み合わせ 5. 隠れシフト問題と Kuperberg のアルゴリズム 6. 線形化攻撃 7. その他の古典攻撃の高速化
7	古典クエリ攻撃（Q1）	1. 桑門・森井による EM 暗号への鍵回復攻撃 2. オンライン-オフライン中間一致攻撃 3. 量子クエリ無しでの Simon のアルゴリズムの応用 4. 古典的に$2k$ビット安全ならkビット耐量子安全か？ 5. その他の古典攻撃の高速化 6. 古典安全性証明の結果が Q1 モデルへ持ち上がる場合
8	ハッシュ関数への（汎用でない）攻撃	1. 衝突攻撃 2. 原像攻撃
9	考察とまとめ	CRYPTREC 暗号リスト、NIST　LWC 最終選考方式 Ascon の耐量子安全性に関する考察

（２）調査結果の概要

調査結果について、主に新規追加事項（表１の赤字箇所）を概説する。

① 準備：攻撃モデル（3 章）

攻撃者は量子計算機を持っており、秘密鍵が埋め込まれた攻撃対象のオラクル（暗号化／復号／認証タグ生成オラクル）へクエリ可能である。

・古典クエリ攻撃（Q1）モデル：オラクルへのクエリが古典情報

・量子クエリ攻撃（Q2）モデル：オラクルへのクエリが量子重ね合わせ状態

なお、ハッシュ関数への攻撃を考える場合はオラクルを使用する必要がないため、本資料では単に量子攻撃モデルと記載する。

② 汎用量子攻撃：タイムメモリトレードオフとレインボーテーブル（5.4 節）

ランダムな関数$H:\{0,1\}^n \to \{0,1\}^n$の原像探索にかかるオンライン計算量$T$は、使用可能なメモリサイズ$S$によって変動することが知られている。最も有名な手法には、Hellman のタイムメモリトレードオフ攻撃と Oechslin のレインボーテーブルがある。いずれも、時間とメモリのトレードオフ$T = O((2^n/S)^2)$が与えられる。

2024 年に Dunkelman らは、量子計算機を用いることで時間とメモリのトレードオフを$T = O((2^n/S)^{1.5})$まで改善できることを示した。攻撃アイデアの根幹は Hellman のタイムメモリトレードオフ攻撃と Oechslin のレインボーテーブルと同じである。

量子計算リソースは、多項式サイズの小さい計算用量子プロセッサと指数的に大きなサイズの QRAM がある、と仮定している。

③ 汎用量子攻撃：ノストラダムス攻撃（5.5 節）

Merkle-Damgard 構造のハッシュ関数Hに対する汎用量子攻撃である。具体的には、攻撃者は以下の問題を解く。

Step 1. 攻撃者は何らかの値yを事前に計算する。

Step 2. Xが選ばれ、攻撃者に与えられる。

Step 3. 攻撃者は$H(X\|R) = y$を満たすRを求める。

古典攻撃モデルでは、nビットハッシュ関数に対し、$O(2^{2n/3})$回の圧縮関数評価で攻撃が実行可能であると知られている。2022 年に Benedikt らは、量子攻撃モデルにおいて評価回数を$O(2^{3n/7})$回まで削減できることを示した。

④ Q2 モデルにおける量子クエリ攻撃：線形化攻撃（6.6 節）

Q2 モデルにおける量子クエリ攻撃のアイデアは、周期関数を作って Simon のアルゴリズムを適用する、ということが大部分を占めている。例えば、EM 暗号、Feistel 暗号、GCM や CBC-MAC を含むブロック暗号利用モード、などへの攻撃がある（報告書 6.1-6.3 節）。ブロック暗号利用モードへの攻撃は 2016 年の Kaplan らによって報告されたが、ISO 標準の LightMAC を含むいくつかのブロック暗号利用モードには同様のアイデ

アを適用できないという問題があった。

2021 年に Bonnetain らは、量子線形化攻撃を提案してこの問題を解決した。攻撃手法の詳細は省略するが、既存のアイデアと同様、攻撃対象の内部構造を詳細に分析して周期関数を作り、Simon のアルゴリズムを適用することにより、多項式時間での識別攻撃を可能にした。

⑤　古典攻撃モデルにて$2k$ビット安全であればkビット耐量子安全か？（7.4 節）

Grover のアルゴリズムにより、kビット鍵の全数探索に必要な計算量が2^kから$2^{k/2}$まで落ちることが知られている。量子計算機の実用化後に共通鍵暗号の安全性を現在と同程度に保つためには鍵長を 2 倍以上にする必要がある、と言われるのはこのためである。

一方、この逆の「古典攻撃モデルにて$2k$ビット安全であれば Q1/Q2 モデルにてkビット安全である」という主張について考察すると、必ずしもこの主張が成り立つとは限らない。例えば、Q2 モデルでは EM 暗号に対する多項式時間攻撃（6.1 節）があり、Q1 モデルでも FX 構成の拡張版である 2XOR 構成と呼ばれる構造のブロック暗号に対し、上記の主張を破る攻撃が 2022 年に Bonnetain らによって報告された。

⑥　古典的な安全性証明の結果が Q1 モデルでもそのまま成り立つ場合（7.6 節）

古典的な安全性証明がランダムオラクルモデルなどのプリミティブを理想化した条件で与えられるのではなく、反証可能な標準的仮定（CTR モードであればブロック暗号が擬似ランダム置換という仮定）のみに依存している場合、古典的な安全性証明の結果が Q1 モデルでの安全性証明としてそのまま成り立つ。

⑦　ハッシュ関数に対する（汎用的でない）量子攻撃（8 章）

衝突攻撃の攻撃可能段数に関して、古典攻撃モデルよりも量子攻撃モデルの方が優れている例がいくつか報告されている。例えば、電子政府推奨暗号リスト掲載の SHA-256、SHA-512、そして SHA3-256 が該当し、結果の詳細は表 2 の通りである。

表 2　古典・量子攻撃モデルでの SHA2 と SHA3 に対する衝突攻撃の比較

対象	出力長	段数	攻撃可能段数	
			古典	量子
SHA-256	256	64	31	38
SHA-512	512	80	31	39
SHA3-224	224	24	5	6
SHA3-256	256	24	5	6

一方、原像攻撃の攻撃可能段数に関して、現状において古典攻撃モデルよりも量子攻撃モデルの方が優れている例は報告されていない。

（３）考察結果の概要

Q2 モデルでの攻撃、Q1 モデルでの攻撃、ハッシュ関数への攻撃に分け、種々の主要な方式の安全性への影響を考察する。より具体的には、CRYPTREC 電子政府推奨暗号リスト掲載方式、そして NIST 標準軽量暗号 Ascon に焦点を当てる。

① Q2 モデルでの攻撃

古典攻撃モデルにおいて安全性が保証されている共通鍵暗号技術（GCM、CBC-MAC、など）に対して多項式時間で実行可能な攻撃が存在するが、Q2 モデルでは攻撃対象が量子回路上に実装されている必要がある。これは非常に特殊な状況であり、現状では既存の共通鍵暗号技術に Q2 モデルでの攻撃の影響が及ぶことは無いと考えられる。特に、CRYPTREC 電子政府推奨暗号リスト掲載方式や NIST 標準軽量暗号 Ascon の安全性を評価する上で Q2 モデルでの攻撃を考慮する必要はない。

② Q1 モデルでの攻撃

Q2 モデルでの攻撃とは異なり、古典攻撃モデルにおいて安全性が保証されている共通鍵暗号技術に対して多項式時間で実行可能な攻撃は存在しない。ただし、古典攻撃モデルにて$2k$ビット以上の安全性があったとしても、Q1 モデルでの安全性がkビット以下になる例も示されているため、暗号技術ごとに確認が必要である。

Q1 モデルにおいて、ハッシュ関数を除く CRYPTREC 電子政府推奨暗号リスト掲載方式や NIST 標準軽量認証暗号の Ascon-AEAD/Ascon-80pq の安全性に量子計算機が与える影響は、“Grover のアルゴリズムを用いるとkビット鍵の全数探索が$O(2^{k/2})$で実行できるため、長期的に保護したいデータには鍵長が 192 ビットや 256 ビットの暗号技術を使用した方が賢明である”と考えられる。これらの暗号技術について、Q1 モデルで安全性が期待できる範囲を表３でまとめる。

③ ハッシュ関数への攻撃

SHA-256、SHA-512、SHA3-256 では、量子計算機が使用可能になると衝突攻撃の攻撃可能段数が古典攻撃モデルと比べて伸びることが知られている。表２で示すように、安全性マージンは十分に確保されているものの、今後の動向を注視する必要がある。その他、CRYPTREC 電子政府推奨暗号リスト掲載のハッシュ関数や NIST 標準軽量ハッシュ関数の Ascon-Hash256/Ascon-XOF128 の安全性に量子計算機が及ぼす影響は、汎用量子攻撃（特に、BHT のアルゴリズム）のみを考慮すれば十分である。これらの暗号技術について、BHT のアルゴリズムを適用するのに必要な計算時間と量子メモリの概算値を表４でまとめる。

表３ ハッシュ関数を除く電子政府推奨暗号リスト掲載方式、Ascon-AEAD、そして Ascon-80pq について、Q1 モデルで安全性が期待できる範囲（Grover のアルゴリズム：$\leq 2^{k/2}$）

技術分類	方式名	鍵長k（ビット）	安全性が期待できる範囲
ブロック暗号	AES Camellia	128 192 256	時間 $\leq 2^{64}$ 時間 $\leq 2^{96}$ 時間 $\leq 2^{128}$
ストリーム暗号	KCipher-2	128	時間 $\leq 2^{64}$
秘匿モード	CBC, CFB, CTR, OFB, XTS	k	時間 $\leq 2^{k/2}$ かつ 古典的に安全性が保証される範囲
認証付き 秘匿モード	CCM, GCM	k	時間 $\leq 2^{k/2}$ かつ 古典的に安全性が保証される範囲
メッセージ 認証コード	CMAC, HMAC	k	時間 $\leq 2^{k/2}$ かつ 古典的に安全性が保証される範囲
認証暗号	ChaCha20- Poly1305	256	時間 $\leq 2^{128}$ かつ 古典的に安全性が保証される範囲
	Ascon- AEAD128	128	時間 $\leq 2^{64}$ かつ 古典的に安全性が保証される範囲
	Ascon-80pq	160	時間 $\leq 2^{80}$ かつ 古典的に安全性が保証される範囲

表４ 電子政府推奨暗号リスト掲載のハッシュ関数、Ascon-Hash256、Ascon-XOF128 に対して BHT のアルゴリズムを適用するのに必要な計算時間と量子メモリの概算値：$\min(2^{c/3}, 2^{h/3})$

方式名	キャパシティc	出力長h	計算時間	量子メモリ
SHA-256 SHA-512/256 SHA3-256	- - 512	256	$2^{85.3}$	$2^{85.3}$
SHA-384 SHA3-384	- 768	384	2^{128}	2^{128}
SHA-512 SHA3-512	- 1024	512	$2^{170.7}$	$2^{170.7}$
SHAKE128	256	$\ell \geq 256$	$\min(2^{85.3}, 2^{\ell/3})$	$\min(2^{85.3}, 2^{\ell/3})$
SHAKE256	512	$\ell \geq 256$	$\min(2^{170.7}, 2^{\ell/3})$	$\min(2^{170.7}, 2^{\ell/3})$
Ascon-Hash256	256	256	$2^{85.3}$	$2^{85.3}$
Ascon-XOF128	256	$\ell > 0$	$\min(2^{85.3}, 2^{\ell/3})$	$\min(2^{85.3}, 2^{\ell/3})$

（４）2019 年度外部評価報告書における結論との差異

具体的な方式の実用面での安全性評価について、2019 年度外部評価執筆時からの大きな差異は、ハッシュ関数の衝突攻撃可能段数が古典攻撃モデルの場合に比べて量子攻撃モデルの場合に伸びることが明らかになってきたということである。

このような状況の変化に応じ、2019 年度版外部評価報告書の結論を表５で示すように変更した。その他の結論部分について大きな差異はない。

表５　ハッシュ関数に関する技術動向の変化

2019 年度外部評価報告書執筆時	現在
古典的に 128 ビット安全性のあるハッシュ関数の安全性に量子攻撃が現実的な脅威を直接及ぼすとは現状考えづらい。	重要な用途に供するハッシュ関数の出力長（スポンジ構造の場合は出力長に加えてキャパシティ長）は BHT のアルゴリズムの計算量$O(2^{n/3})$を基準にして 384 ビットや 512 ビットのものを用いた方が無難であると考えられる。

４　CRYPTREC 文書としての公開可否について【審議事項】

（１）2024 年度外部評価報告書（資料 3-5 別紙）は、今年度の調査対象である共通鍵暗号の耐量子安全性に関する技術動向調査として十分な内容を含んでいると考えられる。このため、本報告書を CRYPTREC の技術調査報告書としてよろしいかご審議いただきたい。

（２）2024 年度外部評価報告書（資料 3-5 別紙）から、CRYPTREC 電子政府推奨暗号リスト掲載方式と NIST 標準軽量暗号 Ascon の安全性に量子計算機が及ぼす影響は、汎用量子アルゴリズム（特に、Grover のアルゴリズムと BHT のアルゴリズム）のみを考慮すれば十分であるという結論を得た。本結論を暗号技術評価委員会としての見解とし、CRYPTREC Report 2024 暗号技術評価委員会報告として公開してよろしいかご審議いただきたい。

以上

量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価 2024 年度版

NTT 社会情報研究所 / NTT 理論量子情報研究センタ
細山田 光倫

2025 年 1 月

エグゼクティブサマリー

量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価を行った．文献調査により，次のことを確認した．

- 量子コンピュータを用いた攻撃のモデル，特にハッシュ関数以外の（秘密鍵を用いる）共通鍵暗号技術への攻撃のモデルには Q1 モデルと Q2 モデルの二種類のモデルが存在する．Q1 モデルにおいては鍵の埋め込まれたオラクルは古典的な攻撃モデルと同じ古典オラクルだが，Q2 モデルにおいては鍵の埋め込まれたオラクルが量子オラクルとなり，攻撃者はオラクルへの量子重ね合わせクエリを行える．Q2 モデルの攻撃を実行するには攻撃対象の暗号技術が（秘密鍵を埋め込んだうえで）量子回路上に実装されている必要がある．
- Q2 モデルにおいては，古典的に安全とされている共通鍵暗号技術（CBC-MAC や GCM など）に多項式時間の攻撃が存在する．多項式時間の攻撃には Simon の量子アルゴリズムが用いられる．
- Q1 モデルにおいては，古典的に安全とされている共通鍵暗号技術に多項式時間の攻撃は現在の所存在しない．しかし従来より認識されていた Grover のアルゴリズムによる鍵全数探索の高速化のみならず，暗号技術の構造に依存した様々な攻撃が存在する．Even-Mansour 暗号および類似の構造を持つ暗号技術に対しては，Q1 モデルであっても Simon のアルゴリズムを活用して古典的攻撃より効率的な攻撃が実行できる．更に，古典的に $2k$ ビット以上の安全性があっても Q1 モデルでの安全性が k ビットを下回る例が示されている．
- 古典的な安全性証明は，ideal permutation model などプリミティブを理想化したモデルでなく反証可能な標準的仮定（ブロック暗号の PRP 安全性など）に依拠するものであれば，Q1 モデルでそのまま通用する．つまり，古典的な安全性証明がついていれば（ブロック暗号などのプリミティブに対する攻撃の影響を考慮する必要はあるが）データ量やクエリ回数などについて安全性が保障される範囲は古典的設定と Q1 モデルで変わらない．Ideal permutation model や ideal cipher model での安全性証明は Q1 モデルでも通用するとは限らず，方式ごとに安全性を再精査する必要がある．
- 既存研究において使用可能と想定されている量子計算のリソースは論文によって異なり，攻撃コストの評価方法も様々である．特にハッシュ関数への汎用攻撃 (衝突探索など) については，使用可能な量子計算のリソースに関する想定や攻撃コストの評価方法に応じて最良の攻撃が異なる．
- 多くのハッシュ関数について，量子計算機が使えるようになれば衝突攻撃の攻撃可能段数が伸びることが示されている．攻撃可能段数が伸びるものには，CRYPTREC の電子政府推奨暗号リストにある SHA-256 と SHA-512 および SHA3-256 が含まれるが，破れているのはそれぞれ 64 段中 38 段，80 段中 39 段，および 24 段中 6 段で，まだ余裕がある．段数削減なしで衝突耐性が破れる心配は今の所無いと考えられるが，今後も研究の進展を注視する必要がある．

また調査した文献の内容に考察を加えた結果，次のような結論を得た．

- ある関数を計算するための古典計算機向けのプログラムコードがあった場合その関数を量子

回路上に実装することが可能になるため，Q2 モデルにおいて多項式時間の攻撃が可能な暗号技術については，例え難読化処理等を施しても，その関数 (例えば CBC-MAC でメッセージからタグを計算する関数) を実装して秘密鍵を埋め込んだコードを，量子コンピュータを持った攻撃者に手渡すべきではない．しかし，攻撃対象となる暗号技術が量子回路上に実装されているような（あるいは量子回路上に移植可能となるような）非常に特殊な状況でない限り，既存の共通鍵暗号技術，特に CRYPTREC の電子政府推奨暗号リストにある共通鍵暗号技術や最近 NIST 標準として選ばれた Ascon に，Q2 モデルの攻撃の影響が及ぶことは現状では無いと考えられる．

- 従来から指摘されていた通り，Grover のアルゴリズムによって k ビット鍵の全数探索が時間 $O(2^{k/2})$ で実行可能になるため，長期的に保護したいデータには秘密鍵の鍵長が 128 ビットの暗号技術でなく 192 ビットや 256 ビットの暗号技術を使用するのが賢明であると考えられる．またハッシュ関数の衝突攻撃可能段数が古典より伸びることが判明していることも考慮すると，重要な用途に供するハッシュ関数の出力長（スポンジ構造の場合は出力長に加えキャパシティ長）は BHT のアルゴリズムの計算量 $O(2^{n/3})$ を基準にして 384 ビットや 512 ビットのものを用いた方が無難であると考えられる．
- CRYPTREC の電子政府推奨暗号リストの共通鍵暗号技術および Ascon の安全性に量子コンピュータが直接与える影響は，Grover のアルゴリズムや BHT のアルゴリズム以上のものは現状では無いと考えられる．しかし Even-Mansour 暗号への Q1 モデルにおける攻撃のように安全性に現実的な影響を直接及ぼす可能性のある攻撃が今後も発見される可能性があり，また種々のハッシュ関数で古典より攻撃可能段数が伸びているため，研究の動向には今後も注意を払っておく必要がある．

2019 年度版のまとめとの差異

具体的な方式の実用面での安全性評価について，2019 年度版執筆時から今回までの間で一番大きな差異は，ハッシュ関数の衝突攻撃可能段数が古典計算機しか使えない場合に比べて伸びることが明らかになってきたということである．これを踏まえ，2019 年度版ではハッシュ関数について「古典的に 128 ビット安全性のあるハッシュ関数の安全性に量子攻撃が現実的な脅威を直接及ぼすとは現状考えづらい」と結論付けていたものを「重要な用途に供するハッシュ関数の出力長（スポンジ構造の場合は出力長に加えキャパシティ長）は BHT のアルゴリズムの計算量 $O(2^{n/3})$ を基準にして 384 ビットや 512 ビットのものを用いた方が無難であると考えられる」と変更した．まとめの他の部分については，2019 年度とこの 2024 年度版で大きな差異は無い．

目次

1 はじめに

Shor の量子アルゴリズム [Sho94] によって現在広く利用されている公開鍵暗号技術が効率的に破れてしまうということが判明して以来，大規模な汎用量子コンピュータが実現してからも安全性を担保できる耐量子公開鍵暗号技術の研究が盛んに行われている．NIST では標準化プロセスが進み，既に電子署名と KEM のいくつかが FIPS 203-205 として標準化されている [NIS24b, NIS24a, NIS24c].

一方共通鍵暗号技術の安全性については量子コンピュータが及ぼす影響は非常に限定的であると考えられていたが，従来は気づかれていなかった攻撃の存在を示す研究結果がこの 10 年程で多数発表されている．本報告書では，量子コンピュータが共通鍵暗号技術の安全性に及ぼす影響について，主に攻撃アルゴリズムの面から既存文献の調査と評価を報告する[*1].

1.1 共通鍵暗号技術に対する量子攻撃の研究の重要性

便利かつ安全な通信は，公開鍵暗号技術と共通鍵暗号技術を組み合わせて初めて実現される．また複数の暗号技術を組み合わせて保護された通信やデータの安全性は使用されている暗号技術のうち最も弱いものによって決まる．ゆえに，量子コンピュータを持った攻撃者から通信やデータを保護するためには，公開鍵暗号技術はもちろん，共通鍵暗号技術も量子コンピュータを用いた攻撃から安全である必要がある．

ブロック暗号やハッシュ関数などの共通鍵暗号プリミティブの耐量子性は，それらに対して有効な量子攻撃が存在するか否かのみによって評価され得る．また古典的に安全性証明がついている方式も，量子計算機に対してどれだけ安全かはわからない．ゆえに，量子コンピュータが共通鍵暗号技術の安全性へ及ぼす影響を把握するためには，量子アルゴリズムを用いる攻撃を研究することが重要となる．

公開鍵暗号技術・共通鍵暗号技術ともに，耐量子性の研究は大規模な汎用量子コンピュータが実現するよりもかなり早い段階で進めておく必要がある．これは主に次の二つの理由による：第一の理由は，現在量子コンピュータを保持していない攻撃者であっても，例えば数十年後に量子コンピュータを入手できた際に解読できるようになることを期待して，現在入手できる限りの暗号文を手に入れようとしている可能性が有る，というものである．このような潜在的脅威を念頭に置くと，数十年単位で長期間安全に保護したいデータはなるべく早い段階から耐量子暗号技術で保護しておくことが望ましい．第二の理由は，基礎研究で知見が蓄えられてから耐量子暗号技術が広く使用されるようになるまでには 10 年単位の時間がかかる，というものである．例えば以前米国の標準暗号であった DES への最初の理論攻撃 [BS92] が発表されてから次の世代の暗号である AES の標準化が公式に発表されるまで 10 年近い時間がかかっている [NIS01]．よって大規模な汎用量子コンピュータが実現される前から，なるべく早く研究を進めておく必要がある．

[*1] 本報告書では「量子コンピュータ」あるいは「量子計算機」とは，ゲート型量子計算機のことを指すものとする．

1.2 本報告書の構成

本報告書の構成は以下の通りである．2 章では，報告書全体を通して必要となる記法等について述べ，共通鍵暗号技術への量子攻撃に欠かせない Grover のアルゴリズムと Simon のアルゴリズムの概要を記述する．3 章では，ハッシュ関数以外の（秘密鍵を利用する）共通鍵暗号技術への 2 つの攻撃モデル（Q1 モデルと Q2 モデル）を紹介する．4 章では，攻撃アルゴリズムのコスト評価方法に関する議論を概観する．特に，使用可能な量子計算のリソースに関する想定に応じて最良の衝突探索アルゴリズムが変わるということを説明する．5 章では，暗号技術の内部構造によらず適用可能な汎用攻撃について，既存研究を概観する．6 章および 7 章ではそれぞれ主に，Q2 モデルと Q1 モデルにおける既存の量子攻撃の研究結果を紹介する．8 章では，特定のハッシュ関数の内部構造を利用した（汎用でない）攻撃について説明する．9 章において，本報告書全体についての考察とまとめを与える．

なお 3 章から 7 章までの内容は主に，共通鍵暗号技術への（古典）攻撃の研究に明るい方が量子攻撃の既存研究を概観するために利用されることを意識して書かれている．本報告書の目的は理論の詳細を議論することでなく既存研究を広く調査し概観することであるため，理論的な厳密性より簡潔な説明を優先する．

1.3 2019 年度版との差異

この報告書は，2019 年度暗号技術関連の調査報告「量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価」[細 20] の改訂版である．特に大きく加筆・修正をした部分は，3.4 節，5.2 節，5.4 節，5.5 節，6.6 節，7.2 節，7.4 節，7.6 節，8 章，および 9 章である．他にも適宜，図表の追加・修正や表現の細かい見直しを行った．

2 準備

本報告書では量子計算のモデルとして量子回路モデル [NC10] を採用し，量子回路は全て Clifford+T ゲートで構成されているものとする．量子オラクルへのクエリが許される場合，オラクルクエリのための特別なゲートが用意され，回路に組み込まれているものとする（注意 2.1）．深さ D_C の量子回路 C が暗号技術 P への量子攻撃で用いられる際は，他に断りの無い限り，C が入力を得てから最終的な出力を計算し終わるまでの時間は D/D_P であるとみなす．ここで D_P は攻撃対象の暗号技術 P を実装するために必要な量子回路の深さである[*2]．また他に断りの無い限り，量子計算に関するすべての操作は誤り無しで実行されるものとし，量子誤り訂正に関連するコストは考慮に入れないものとする．量子状態の観測というと計算基底での観測を指すこととする．表記を簡潔にするため，計算量を示す際はパラメータの多項式程度の因子を省略することがある．

$x, y \in \{0,1\}^n$ に対して $x \oplus y$ は x と y の排他的論理和を表すとする．また $x \in \{0,1\}^m$ と $x' \in \{0,1\}^n$ に対して $x||x'$ は x と x' を結合した $(m+n)$ ビットのビット列を表すとする．集合 $\{0,1\}^n$ は演算 $\oplus$ について群を成すが，この群を $\mathbb{F}_2$ 上の n 次元ベクトル空間 $\mathbb{F}_2^n$ と同一視する．また $x = x_1||\cdots||x_n, y = y_1||\cdots||y_n \in \mathbb{F}_2^n$ $(x_i, y_i \in \{0,1\})$ に対して $x \cdot y$ は x と y のドット積 $x_1y_1 \oplus \cdots \oplus x_ny_n$ を表すとする．二つの n ビット列 x と y が直交するとは，$x \cdot y = 0$ が成り立つこととする．古典ビット列を出力する（量子）アルゴリズム $\mathcal{A}$ に対して $x \leftarrow \mathcal{A}$ と書いたとき，これは $\mathcal{A}$ を実行した結果その出力が x になることを意味する．

注意 2.1. 一般に，関数 $f : \{0,1\}^m \to \{0,1\}^n$ の（古典）オラクルと言うと，任意の入力 $x \in \{0,1\}^m$ に対して値 $f(x)$ を返すブラックボックスのことを指す．一方，関数 f の量子オラクルは

$$U_f : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$$

で定義されるユニタリ作用素としてモデル化される．

■量子ランダムアクセスメモリ（QRAM）　保存されたデータに量子重ね合わせ状態でランダムアクセスすることができるようなメモリを量子ランダムアクセスメモリ（QRAM）と言う [GLM08].

N 個のデータ $D_1, \ldots, D_N$ を格納している古典的な RAM にアドレス i を渡すと，対応するデータ D_i を効率的に取得できる．このデータ取得を量子重ね合わせで実行できるようにするのが QRAM である．即ち，アドレスの量子状態 $\sum_i c_i |i\rangle$ を QRAM に渡すと，対応するデータの量子重ね合わせ $\sum_i c_i |i\rangle |D_i\rangle$ を返す．

データの取得のみでなくデータの書き込みをも量子重ね合わせで行えるような QRAM を考えることもできる．前段落のようにデータの取得のみ量子重ね合わせを許容するものを QRACM，データの書き込みも量子重ね合わせを許容するものを QRAQM と呼んで区別することがある．攻撃アルゴリズムによっては QRAQM を必要とするものがあるが，本稿で QRAM といえば基本的に QRACM を指すとする。

[*2] 計算時間を D_C でなく D_C/D_P と見積もるのは，攻撃時間評価が攻撃対象の暗号技術をどう量子回路上に実装するかに依存せず決まるようにするため，また共通鍵暗号技術の研究における古典的な攻撃時間評価の慣習と整合性を取るためである．

議論を簡単にするため，本稿では RAM・QRAM ともにアクセスに要する時間は定数時間であると仮定する．

2.1 Grover のアルゴリズム

問題 2.1 (データベース探索)**.** t を正の整数 ($t \leq 2^n$) とする．関数 $f : \{0,1\}^n \to \{0,1\}$ について $|f^{-1}(1)| = t$ が成り立っているとする．f が (量子) オラクルとして与えられたとき，$f(x) = 1$ を充たす x を 1 つ見つけよ．

古典計算機でこの問題を解くには $\Omega(2^n/t)$ 回の古典クエリが必要であるが，量子計算機では Grover のアルゴリズム (あるいはその一般化) [Gro96, BBHT98] を使用すると $O(\sqrt{2^n/t})$ 回の量子クエリで解けることが知られている．アルゴリズムに用いられる量子回路は幅 $O(n)$，深さ $O(2^{n/2})$ となる (f へのクエリが時間 1 で実行されるとすると，アルゴリズムの実行に必要な時間も $O(\sqrt{2^n/t})$ となる)．

また Grover のアルゴリズムの簡単な応用として，以下の問題を解くアルゴリズムを作ることができる [HSX17].

問題 2.2 (ランダム関数の (多重) 原像探索)**.** t を正の整数 ($t \leq 2^n$) とする．$F : \{0,1\}^n \to \{0,1\}^n$ をランダム関数，L を $\{0,1\}^n$ の部分集合とし，$|L| = t$ とする．F が (量子) オラクルとして与えられるとき，$F(x) \in L$ となる x を一つ求めよ．

この問題を古典計算機で解くには F への古典クエリが $\Omega(2^n/t)$ 回必要である．しかし，F が量子オラクルとして与えられていれば，以下のような簡単な量子アルゴリズムを実行すると $O(\sqrt{2^n/t})$ 回の F への量子クエリで問題 2.2 を解くことができる：

Grover のアルゴリズムを用いた自明な (多重) 原像探索アルゴリズム

1. $f_L^F : \{0,1\}^n \to \{0,1\}$ を，$F(x) \in L$ であるときかつその時に限り $f_L^F(x) = 1$，と定義する．
2. f_L^F に Grover のアルゴリズムを適用する．

ステップ 1 の関数 f_L^F は幅 $\tilde{O}(|L|)$・深さ $O(1)$ の量子回路上に実装できる．F がランダム関数であることから $f_L^F(x) = 1$ となる x はおおむね t 個存在し，よって上記アルゴリズムに必要な量子回路は幅 $\tilde{O}(|L|)$・深さ $O(\sqrt{2^n/t})$ となる．(F へのクエリが時間 1 で実行されるとすると，アルゴリズムの実行に必要な時間も $O(\sqrt{2^n/t})$ となる)．

注意 2.2. ランダム関数 (ハッシュ関数) の原像探索問題やブロック暗号の鍵全数探索問題は自明に問題 2.2 の $t = 1$ の場合に帰着される．上記アルゴリズムにより，n ビット出力ハッシュ関数の原像探索は概ね時間 $2^{n/2}$ で，また k ビット鍵ブロック暗号の鍵全数探索は概ね時間 $2^{k/2}$ で，それぞれ実行可能となる．詳細は 5.1 節を参照されたい．

2.2 Simon のアルゴリズム

問題 2.3. 関数 $f : \{0,1\}^n \to \{0,1\}^n$ と $s \in \{0,1\}^n$ があって，以下の条件を満たすとする：

$$x = y \oplus s \text{ であるとき，かつそのときに限り } f(x) = f(y). \tag{1}$$

f が (量子) オラクルとして与えられたとき，s を求めよ．

条件 (1) は特に f が周期 s を持つ周期関数であるということを示しており，この問題は周期関数の周期を探索する問題である．

この問題を古典計算機で解くには $\Omega(2^{n/2})$ 回の古典クエリが必要であるが，Simon の量子アルゴリズムを用いると $O(n)$ 回の量子クエリで解くことができる [Sim94]．アルゴリズムの概要を以下に示す：

Simon のアルゴリズム

1. 下記のサブルーチン **SSub** を cn 回繰り返して n 個の元 $y_1, \ldots, y_n \in \{0,1\}^n$ を得る (c は適当な定数，例えば $c = 2$)．
2. $\{0,1\}^n$ を $\mathbb{F}_2$ 上の n 次元ベクトル空間とみなしたときの $y_1, \ldots, y_n$ の張るベクトル空間の次元 d を計算する．
3. $d \neq n-1$ なら，アルゴリズムは失敗したとして終了する．
4. $d = n-1$ なら，$y_1, \ldots, y_n$ に直交するベクトル s' を計算して出力する．

サブルーチン SSub

1. $2n$ 量子ビットの量子状態
$$|0^n\rangle |0^n\rangle \tag{2}$$
を用意する．
2. 状態 (2) の左 n 量子ビットに Hadamard 変換 $H^{\otimes n}$ をかけ，
$$\sum_x \sqrt{1/2^n} |x\rangle |0^n\rangle \tag{3}$$
を得る．
3. f への量子クエリを行い (ユニタリ作用素 $U_f : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$ を状態 (3) に作用させ)，
$$\sum_x \sqrt{1/2^n} |x\rangle |f(x)\rangle \tag{4}$$
を得る．
4. 状態 (4) の左 n 量子ビットに Hadamard 変換 $H^{\otimes n}$ をかけ，
$$\sum_{x,y} (-1)^{x \cdot y}/2^n |y\rangle |f(x)\rangle \tag{5}$$
を得る．

5. 状態 (5) の左 n 量子ビットを観測し，結果 (n ビットのビット列 y) を出力する．

条件 (1) を使うと，サブルーチン **SSub** は ($\{0,1\}^n$ を $\mathbb{F}_2$ 上の n 次元ベクトル空間と見たとき)s に直交するベクトルを一様ランダムに出力することがわかる．ゆえに，Simon のアルゴリズムのステップ 2 において非常に高い確率で $d = n-1$ となり，$s' = s$ となることがわかる．サブルーチン **SSub** は幅 $2n$・深さ $O(1)$ の量子回路を用いて実行することができ，f へ 1 回だけクエリを行う．また Simon のアルゴリズムのステップ 2 と 4 はガウス消去法を用いて，時間 $O(n^3)$ で実行できる．よって Simon のアルゴリズム全体として，f へ行う量子クエリは $O(n)$ 回，使用する量子回路は幅 $O(n)$ かつ深さ $O(1)$，また実行に必要な時間は f へのクエリが時間 1 で実行されるとした場合 $O(n^3)$ となる．

■条件 (1) の緩和　共通鍵暗号技術への攻撃に Simon のアルゴリズムを応用しようとする際，アルゴリズムを適用しようとする関数 f について

$$y = x \oplus s \text{ ならば } f(x) = f(y)$$

が成り立っていても，その逆

$$f(x) = f(y) \text{ ならば } y = x \oplus s \tag{6}$$

が成り立つとは限らない．しかし Kaplan らは，条件 (6) が成り立たずとも f が s を周期に持つことを除いてほぼランダムな関数である場合，Simon のアルゴリズムを適用することで s を計算できるということを示した [KLLN16a, Theorem 2].

3　攻撃のモデル：古典クエリと量子クエリ

本章では，量子計算機を用いた共通鍵暗号技術への攻撃を考察する際の攻撃モデルについて述べる．

秘密鍵を使用する共通鍵暗号技術（つまりハッシュ関数以外の共通鍵暗号技術）への量子計算機を用いた攻撃には，攻撃者がアクセスできる鍵の埋め込まれたオラクルの種類に応じて二つのモデルがある．一つはオラクルが古典攻撃と同じであるモデル（Q1 モデル），もう一方はオラクルへのクエリおよびオラクルの出力が量子重ね合わせ状態になることを許容するモデル（Q2 モデル）である [KLLN16b]．

本章の構成は次のとおりである．まず 3.1 節で古典的な攻撃のモデルを振り返り，3.2 節と 3.3 節で Q1 モデルと Q2 モデルを説明する．その後 3.4 節で二つのモデルを比較する．ハッシュ関数への攻撃のモデルについては 3.5 節で補足する．

3.1　古典的攻撃モデル

（ハッシュ関数を除く）共通鍵暗号技術への攻撃の典型的なモデルは，攻撃者が計算機を持っており（あるいは，攻撃者自体がアルゴリズムであるとモデル化し），ランダムに生成された秘密鍵の埋め込まれたオラクル（暗号化オラクル・復号オラクルや認証タグ生成オラクル）へメッセージを自由にクエリしてその結果を得られる，というものである．

例えばブロック暗号 E_K（K は秘密鍵）に対する選択平文攻撃による鍵回復について考える際は，平文 M をクエリすると $E_K(M)$ を時間 1 で返してくれるオラクルの存在を前提とする．攻撃者はオラクルへ様々な平文をクエリしつつ，自らの所持する計算機上で秘密鍵を推測するための計算を行う (図 1 を参照)．

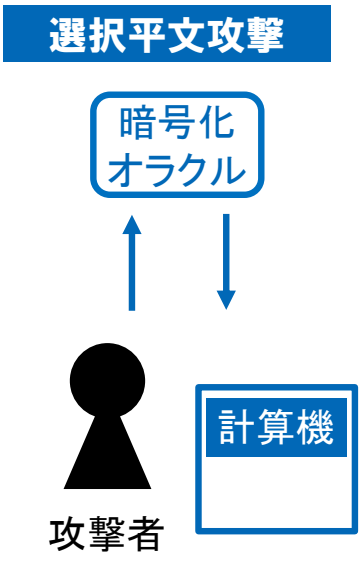


図 1　古典攻撃モデルの例（選択平文攻撃）

3.2　Q1 モデル（古典クエリ攻撃モデル）

このモデルにおいては，攻撃者の計算機が量子計算機になる（あるいは，攻撃者自体が量子アルゴリズムであるとモデル化する）が，それ以外の設定は基本的に古典的攻撃モデルと同じである．攻撃者はオラクルへ様々なデータをクエリしてその結果を取得しつつ，自らの所持する量子計算機上で攻撃に必要な計算を行う．

量子計算機は様々な問題を古典計算機より高速に解けるため，古典的攻撃モデルに比べて高速な攻撃が可能になる (図 2 を参照).

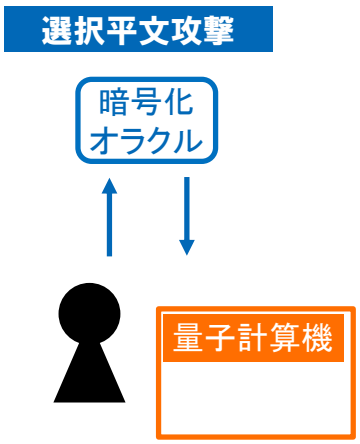


図 2　Q1 攻撃モデルの例（選択平文攻撃）

3.3　Q2 モデル（量子クエリ攻撃モデル）

このモデルでは，攻撃者の計算機が量子計算機であることに加え，鍵が埋め込まれたオラクルの入出力も量子重ね合わせ状態になる．つまり，攻撃者に量子オラクルが与えられる，という設定を考える（図 3 参照）.

例えばブロック暗号 E_K に対する量子選択平文攻撃では，平文 $|M\rangle$ をクエリすると対応する暗号文 $E_K(M)$ が得られるのみでなく，二つの平文 M_1 と M_2 の量子重ね合わせ状態 $\sqrt{1/2}\,|M_1\rangle\,|0^n\rangle + \sqrt{1/2}\,|M_2\rangle\,|0^n\rangle$ を E_K をクエリすることが許される．この状態をクエリすると，M_1 と M_2 が同時に暗号化され，量子重ね合わせ状態 $\sqrt{1/2}\,|M_1\rangle\,|E_K(M_1)\rangle + \sqrt{1/2}\,|M_2\rangle\,|E_K(M_2)\rangle$ が返される．

3.4　Q1 モデルと Q2 モデルの比較

Q2 モデルでは，鍵の埋め込まれたオラクルへ量子重ね合わせクエリを攻撃者が行える状況を想定している．例えばブロック暗号への量子選択平文攻撃であれば，秘密鍵の埋め込まれた暗号化関数が量子回路上に実装されており，その量子回路へ攻撃者が自由に入力を与えて出力を得られるという状況を想定している．

図 3　Q2 攻撃モデルの例（量子選択平文攻撃）

一般に，計算の過程で量子重ね合わせ状態を扱える場面が増えれば増えるほど攻撃者の能力が強くなる．よって攻撃者の能力は，Q1 モデルより Q2 モデルのほうが強い．実際，Q1 モデルでは多項式時間攻撃が発見されていないが Q2 モデルでは多項式時間攻撃が可能になる，というような暗号技術がいくつも存在する（Q2 モデルにおける攻撃についての詳細は 6 章を参照）．

また，Q1 モデルでは鍵の埋め込まれたオラクルが古典計算機上に実装されている状況を想定しているため，Q1 モデルの方が Q2 モデルに比べてより実現可能性が高い．

しかし，Q2 モデルが「非現実的なモデル」というわけではない [HS18a]．例えば以下のような状況では Q2 モデルが現実的なモデルになる：

a) 通信や情報処理の多くが量子状態で行われているような未来
b) 攻撃対象の暗号技術を実装し鍵の埋め込まれた（古典計算機用の）プログラムコードを攻撃者が入手可能な状況

a) の状況において Q2 モデルが現実的なモデルとなるのは明らかである．また古典計算機のプログラムは原理上量子計算機へ移植可能であるため，b) の状況においても Q2 モデルが妥当なモデルとなる．ここで，b) の状況は，攻撃対象の暗号技術（何らかの鍵付きの暗号学的関数）F_K を実装したプログラムコードに何らかの方法で難読化処理が施されたものを攻撃者が保持している場合にも発生し得ることに注意されたい：難読化処理後のプログラムコードを $\mathcal{C}$ とおく．すると，たとえ古典攻撃で鍵 K などの秘密情報を抽出することが困難であっても，もし Q2 モデルにおいて F_K への効率的な鍵回復攻撃が存在するなら，攻撃者はプログラムコード $\mathcal{C}$ を量子計算機上に移植することで F_K の量子オラクルをシミュレートし，効率的に鍵を回復することができる．このような量子攻撃の可能性は Q1 モデルのみでは捉えることができない．

加えて，いくつかの Q1 モデルにおける攻撃は，Q2 モデルにおける攻撃を元に考案されている（例えば 7.3 節で紹介する [BHN+19]）．より実現可能性が高い Q1 モデルでの攻撃を発見する前段階として，Q2 モデルにおける攻撃の研究は有用である．

更に，鍵長が入力長より十分長いときは，量子オラクルをシミュレートすることにより Q2 モデ

ルの攻撃を Q1 モデルの攻撃へ変換できる．例えば，あるブロック暗号 E_K の入力長が n ビット，鍵長が k ビットで，$k > 2n$ であるとする（例：SKINNY [BJK+16]）．この E_K について，古典暗号化オラクルが与えられており（つまり Q1 モデル），また Q2 モデルにおいては Grover のアルゴリズムを用いた鍵全数探索より効率的な鍵回復攻撃が存在する，と仮定する．このときまず，E_K の full-codebook を得て QRAM に保存することを考える．即ち，ありうる全ての平文 x について，x を暗号化オラクルにクエリして $E_K(x)$ を得てペア $(x, E_K(x))$ を QRAM に保存しておく．するとこの QRAM を用いて $E_K(x)$ の量子オラクルを効率的にシミュレートでき，よって Q2 モデルの攻撃を実行できる．Full-codebook を QRAM に保存するにはクエリ・計算量ともに 2^n が必要であるが，$k > 2n$ であれば 2^n は Grover のアルゴリズムを用いた鍵全数探索（5.1 節を参照）の計算量 $2^{k/2}$ よりも小さい．ゆえに，鍵全数探索よりも効率的な Q1 モデルの攻撃が得られる．

以上の理由により，Q2 モデルにおける攻撃の研究は，共通鍵暗号技術の耐量子性を評価する上で Q1 モデルにおける攻撃の研究と同様に重要である．

■Full-codebook を用いる攻撃についての注意　Full-codebook を用いる攻撃では全ての平文 x に対応する暗号文 $E_K(x)$ をクエリして保存する．これはかなりコストのかかる処理であり，かつ full-codebook を得てしまえば鍵を知らずとも任意の暗号文を復号できる．それでもなお，コストが（Grover のアルゴリズムを用いた）鍵全数探索を下回るような鍵回復攻撃が見つかった場合，暗号 E は理論上破られたと見做される．

なぜこのような考え方をするかという理由は幾つかあるが（そして研究者によって言うことが若干変わるのではないかと考えるが），まず挙げられる理由は「たとえ full-codebook を用いるものであったとしても，鍵全数探索より効率的な鍵回復攻撃が存在するということは，E が他の暗号に無い弱点を持つことを示す」というものである：入出力長や鍵長，そして処理効率が E と同じ別のブロック暗号 E' があったとする．鍵全数探索より効率的な鍵回復攻撃が E には見つかっていて E' には見つかっていないと仮定すると，E の内部構造は E' のそれと比べて暗号文に偏りを生み出しやすいと考えられる．E と E' の処理効率などが同じなのであれば，安全性の観点から E を使う理由は無く E' を用いるべきである．

3.5　ハッシュ関数への攻撃のモデル

SHA-2 や SHA-3 などのハッシュ関数は仕様が公開された決定的アルゴリズムであり，モードやプロトコルの構成部品として使われない限り秘密鍵を入力に取らない．ゆえにハッシュ関数への攻撃，特に衝突攻撃や原像攻撃などを考える際は，量子計算機の有無に関わらずオラクルが登場しない．攻撃者は，自分で用意した計算資源のみを用いて衝突や原像を探索する．

量子計算機が利用可能な場合，攻撃者はハッシュ関数を量子計算機上に実装して攻撃に利用できる．必要とあれば関数の量子オラクルを自分でシミュレートすることも可能である．その意味では，ハッシュ関数の攻撃は Q2 モデルに近いとも考えられる．（実際，量子ランダムオラクルモデル [BDF+11] においては，ハッシュ関数がランダム関数の量子オラクルであるとモデル化して様々な暗号技術の安全性証明が与えられる．）

4　攻撃コスト評価方法に関する議論

本章では，攻撃アルゴリズムのコスト（あるいは効率性）をどう評価すべきかということに関して暗号研究者の間でなされている既存の議論を，ハッシュ関数に対する汎用衝突探索アルゴリズム[*3]の研究の進展を軸に説明する．汎用でない衝突攻撃，つまり具体的なハッシュ関数に対してその内部構造を活用するような衝突攻撃については，のちほど 8 章で詳述する．

量子計算機の研究自体がまだまだ発展途上であることから，攻撃コストの評価方法について暗号研究者の間で統一的な合意が取れているわけではない．既存研究において使用可能と想定されている量子計算のリソースは論文によって異なり，攻撃コストの評価方法も様々である．特にハッシュ関数への攻撃については，使用可能な量子計算のリソースに関する想定や攻撃コストの評価方法に応じて最良の攻撃が異なる．

なるべく中立的な立場から既存の議論の概要を紹介することに努め，各々の主張が妥当であるか否かの判断には立ち入らないこととする．今後の研究の進展や技術の発展によって，着目すべきコスト評価方法が大きく変化する可能性があることに留意されたい．

4.1　古典的衝突探索と誕生日のパラドクス

関数 $h : X \to Y$ の衝突とは，X の要素のペア (x, x') であって $x \neq x'$ かつ $h(x) = h(x')$ を充たすものである．h が暗号学的ハッシュ関数（例えば SHA-2 や SHA-3）である場合 X のサイズは Y 以上で，また h は完全にランダムに振る舞うと見做して差し支えない．以下簡単のため，h はランダム関数，また $X = Y = \{0,1\}^n$ であるとする．

古典的には，有名な誕生日のパラドクスにより，以下の命題が成り立つことがわかる：

命題 4.1. $S \subset \{0,1\}^n$ をサイズ $\sqrt{2^n}$ の任意の部分集合とする．このとき $h|_S$ には確率 $\Theta(1)$ で衝突が存在する[*4]．

この命題を利用するとランダム関数 h の衝突を $O(2^{n/2})$ のメモリを使用して時間 $O(2^{n/2})$ で発見する自明なアルゴリズムが得られる．更に，より洗練されたアルゴリズム (rho 法) を使用すると，時間は $O(2^{n/2})$ そのままに，メモリ使用量を $O(1)$ に減らして衝突を発見できることが知られている [Pol75]．また任意の確率的アルゴリズムについて，ランダム関数 h の衝突を確率 $\Theta(1)$ で探索するには（並列計算を考慮に入れなければ）時間 $\Theta(2^{n/2})$ が必要であることが容易に証明される．より正確に言うと，h の評価 (h がオラクルとして与えられるときの，h へのクエリ回数) が $\Theta(2^{n/2})$ 回必要であることが証明される．

[*3] すなわち，ハッシュ関数の具体的な内部構造によらず適用できる衝突探索アルゴリズム．

[*4] 確率は関数 h を一様ランダムに選ぶ試行について定義される．

4.2 最初の量子衝突探索アルゴリズム：BHT

Brassard らは 1997 年，Grover のアルゴリズムを応用し，n ビットハッシュ関数の衝突探索を時間 $O(2^{n/3})$ で探索するアルゴリズムを発表した [BHT97][*5]．以下このアルゴリズムを，考案者らの頭文字をとって BHT のアルゴリズムと呼ぶ．前節で述べたように，古典的な衝突探索アルゴリズムは時間 $\Omega(2^{n/2})$ を要するため，BHT のアルゴリズムを用いると $O(2^{n/6})$ の高速化が得られている．アルゴリズムの概要を以下に示す．

1. サイズ $2^{n/3}$ の部分集合 $S \subset \{0,1\}^n$ をとる．全ての $x \in S$ について $h(x)$ を計算し，ペア $(x, h(x))$ をリスト L に保存する（リスト L は QRAM に格納する）．
2. $x' \in \{0,1\}^n \setminus S$ と $(x, h(x)) \in L$ の組であって $h(x') = h(x)$ を満たすようなものを，2.1 節で紹介した（Grover のアルゴリズムを自明に適用することによって得られる）多重衝突探索アルゴリズムを用いて見つける．
3. (x, x') を出力する．

なお，h の量子オラクルが攻撃者に与えられていると考え，h への 1 回のクエリは時間 1 で行えると仮定する．

リスト L のサイズが $2^{n/3}$ であるため，ステップ 1 に要する時間および h へのクエリ回数は $O(2^{n/3})$ である．ゆえにアルゴリズム全体で要する時間は $O(2^{n/3})$ となる．

ここで，ステップ 2 において 2.1 節の重衝突探索アルゴリズムを用いる際，サイズ $2^{n/3}$ のリスト L へ量子重ね合わせアクセスが必要となることに留意されたい．

BHT のアルゴリズムは，サイズ $O(2^{n/3})$ の量子メモリ (QRAM) を使用し，時間 $O(2^{n/3})$ でランダム関数 h の衝突を見つけるアルゴリズムである．h の評価回数 (h へのクエリ回数) もおよそ $2^{n/3}$ である．

なお，Zhandry によりランダム関数 h の衝突探索に必要な h の評価回数は $\Omega(2^{n/3})$ 以上であるということが証明されているため，h の評価回数という観点からは BHT のアルゴリズムは最良のアルゴリズムである [Zha15].

注意 4.1. 二つの関数 $h, g : \{0,1\}^n \to \{0,1\}^n$ に対して，ペア (x, x') であって $h(x) = g(x')$ を充たすものを関数 h と g の claw と呼ぶ．h と g がランダムであるとき，BHT のアルゴリズムを適用すると h と g の claw を時間 $2^{n/3}$ で見つけることができる（ステップ 1 とリスト L はそのままにして，ステップ 2 において多重原像探索アルゴリズムを L と g に適用すればよい）．ここで関数 h の評価は古典的に行えれば十分で，h を計算する量子回路 (または h の量子オラクル) は必要が無いことに注意する．

[*5] 正確に言うと現論文においてランダム（とみなせる）関数の衝突探索が議論されているわけではないが，ランダム関数の衝突探索にも適用できることが容易に示せる [HSTX19].

4.3 BHT のアルゴリズムの効率性をめぐる議論

前節で述べたように，BHT のアルゴリズムはランダム関数 h の衝突をサイズ $O(2^{n/3})$ の量子メモリ（QRAM）を用いて時間 $O(2^{n/3})$ で発見する．一見すると BHT のアルゴリズムが Grover のアルゴリズムを用いた自明な衝突探索アルゴリズム[*6]や古典アルゴリズムより効率的であることに議論の余地は無いように思われる．しかし，BHT のアルゴリズムが $O(2^{n/3})$ という非常に大きな量子メモリ（QRAM）を必要とすることから，Grover と Rudolph および Bernstein は BHT のアルゴリズムが Grover のアルゴリズムを用いた自明な衝突探索アルゴリズムや古典アルゴリズムより効率的とは言えないと主張した [GR04, Ber09].

まず Grover と Rudolph は，メモリの量子ビットと計算用の量子ビットは古典計算機におけるメモリと CPU のように明確に区別できるものではなく，よって大きさ $O(Q)$ の量子メモリを必要とする BHT の効率性を他の衝突探索アルゴリズムの効率性と比較する際は $O(Q)$ 個の量子ビットを全て演算に用いる（並列）量子アルゴリズムを比較対象に入れるのが妥当であると主張した [GR04]. 特に，Grover のアルゴリズムを用いた自明な衝突探索アルゴリズムを約 $2^{n/3}$ 量子ビットを用いて並列化すれば BHT のアルゴリズムと同じく時間 $O(2^{n/3})$ で衝突を発見可能であり，よって BHT のアルゴリズムの効率性が並列化した自明な衝突探索アルゴリズムの効率性と変わらないと主張した[*7].

更に Bernstein は，そもそもサイズ $2^{n/3}$ の古典計算機がある（$2^{n/3}$ 個の CPU があって，それらが互いに通信し合い協調して計算を行える）場合は，古典アルゴリズム (並列 rho 法 [vOW94]) を用いて時間 $O(2^{n/6})$ で衝突を発見できることを示した [Ber09][*8]．$2^{n/3}$ 個の量子ビットを利用できる量子計算機はサイズおよそ $2^{n/3}$ の古典計算機として使用できるため，実行時間と使用するハードウェアの大きさとのトレードオフの観点からは古典アルゴリズムの方が BHT の量子アルゴリズムより効率的であると主張した.

■通信コストに関する議論　Bernstein は [Ber09] において，攻撃アルゴリズムの効率を評価する際，通信コストを考慮に入れるべきだと主張している．ここでの通信コストとは，量子計算機を構成する量子ビットの間で情報をやり取りするのに必要なコスト，あるいは小さな（例えば定数サイズまたは多項式サイズの）量子計算機の集合が互いに量子通信を行い協調して計算を行うことで大規模な（例えば指数的に大きなサイズの）量子計算機を実現しているような状況における通信のコストを指す．以下，大規模な量子計算機が小さな量子計算機の集合として実現されているとき，小さな量子計算機のことを量子プロセッサと呼ぶことにする.

量子回路モデルにおいては，任意の量子ビットのペアを 2 量子ビット入出力の量子ゲートの入力に取ることが可能である．これは大規模な量子計算機が小さな量子プロセッサの集合として実現さ

[*6] x をランダムに取って $h(x)$ を計算し，次に Grover のアルゴリズムで $h(x') \neq h(x)$ なる x' を探索する．すると時間 $O(2^{n/2})$ で h の衝突を発見できる.

[*7] なお，$\tilde{O}(2^{n/3})$ 個の量子ビットを用いて並列化した自明な衝突探索アルゴリズムは単位時間あたり $\tilde{O}(2^{n/3})$ 回の h の評価を独立して行うため，h の評価回数は合計で $\tilde{O}(2^{n/3}) \times \tilde{O}(2^{n/3}) = \tilde{O}(2^{2n/3})$ となって BHT のアルゴリズムが h を評価する回数 $\tilde{O}(2^{n/3})$ を大幅に上回る.

[*8] Bernstein の指摘した手法によっても h の評価回数は $O(2^{n/2})$ であり，h の評価回数という観点からは BHT のアルゴリズムの方が優れている.

れている状況において，任意の量子プロセッサ同士が時間 $O(1)$ で通信可能であることに対応する．

しかし現実世界で大規模な量子計算機を実現する際には，量子ビット（あるいは，小さな量子プロセッサたち）が二次元メッシュ状に並べられていて隣り合った量子ビット同士（隣り合ったプロセッサ同士）のみが直接通信できると想定するのが妥当である，と Bernstein は主張した [Ber09]．2^s 個のプロセッサが $\sqrt{2^s} \times \sqrt{2^s}$ の二次元格子状に配置されており隣り合った量子プロセッサ同士の通信にかかる時間が $O(1)$ のとき，最も離れたプロセッサ同士が通信をしようと思うと $O(\sqrt{2^s})$ だけの時間を要することになる．

Bernstein が [Ber09] において示した古典衝突探索アルゴリズム（並列 rho 法）は，量子ビット（あるいは小さな量子プロセッサ）を二次元格子状に配置した構造の量子計算機でも前述の計算量で衝突探索を実行できる．特に，サイズ 2^s の量子計算機を用いた際に衝突を発見するのに要する時間は $O(2^{n/2-s})$ である．

なお Grover と Rudolph が指摘した自明な衝突探索アルゴリズムの並列化は，多項式サイズの小さな量子プロセッサたちが互いに独立して計算を行うように並列化を行う．ゆえに，プロセッサ間の量子通信は発生しない．なお 2^s 個の多項式サイズの小さな量子プロセッサが利用可能なとき衝突探索に要する時間は $O(2^{(n-s)/2})$ となる．

4.4 使用量子ビット数の観点から効率的なアルゴリズム：CNS

量子計算機は古典計算機に比べて実現が非常に難しいという事実を鑑みると，攻撃者の使用可能なリソースとして大規模な古典計算機[*9]と多項式サイズ程度の小さな量子計算機がある，と想定することは妥当である．

このような設定ではもはや BHT のアルゴリズムはもちろんのこと Grover と Rudolph の並列原像探索や Bernstein の指摘した並列 rho 法も衝突探索に時間 $O(2^{n/2})$ を要する．しかし Chailloux らはこのような設定においても，古典メモリ $O(2^{n/5})$ とサイズ $O(\mathrm{poly}(n))$ の量子計算機を用いて時間 $\tilde{O}(2^{2n/5})$ で衝突を発見するアルゴリズムが存在することを示した [CNS17]．以下このアルゴリズムを，考案者の頭文字を取って CNS のアルゴリズムと呼ぶことにする．

Chailloux らは [CNS17] において CNS のアルゴリズムを並列化した際の実行時間評価も与えている．CNS のアルゴリズムを 2^s 個の量子プロセッサを用いて並列化すると，時間 $\tilde{O}(2^{2n/5-3s/5})$ で衝突を発見する．なお使用する古典メモリのサイズは $\tilde{O}(2^{n/5+s/5})$ となる．またこの計算量は $s \leq n/4$ のときのみ有効であり，Grover と Rudolph らが指摘した並列アルゴリズムと同様，各量子プロセッサは独立して計算を行うためプロセッサ間の量子通信はしない．

4.5 ここまでのまとめ

本章でこれまでに説明したことを総合すると，ハッシュ関数の汎用衝突探索アルゴリズムに関する既存研究において，使用可能とされる量子計算リソースの設定には様々なものがあり，以下のよ

[*9] 古典攻撃の研究における典型的な設定に従い，CPU は一つしか持たず並列計算はできないが指数的に大きなメモリを持つと想定する．

うに分類できる[*10]：

Case 0　小さいサイズの計算用の量子プロセッサと，指数的に大きなサイズの量子メモリ（QRAM）から成る量子計算機があるという想定
Case 1a 小さいサイズの計算用量子プロセッサが大量に使用可能であり，任意のプロセッサのペア同士が時間 $O(1)$ で通信できる．
Case 1b 小さいサイズの計算用量子プロセッサが大量に使用可能で 2 次元格子点状に配置されており，隣り合ったプロセッサ同士のみが（時間 $O(1)$ で）通信できる．
Case 1c 小さいサイズの計算用の計算用量子プロセッサが大量に使用可能であり，それらは互いに通信することなく独立して計算を行う．
Case 2　小さいサイズの計算用量子計算プロセッサが 1 つだけ使用可能である．

なお，小さいサイズというのは高々 n の多項式程度のサイズを指すものとする．また全てのケースにおいて，量子計算機とは別に，計算用プロセッサ（CPU）とメモリを備えた古典計算機が 1 つ追加で使用可能であると想定する．（この古典計算機は並列計算を行わないものとし，メモリは指数的に大きなものが使用可能であるとする．また設定によっては古典計算機のサイズも宣言する．）

またそれぞれの設定において最良の汎用衝突探索アルゴリズムは異なる．Case 0 において最も速い汎用衝突探索アルゴリズムは BHT のアルゴリズムである（4.2 節）．Case 2 における現状で最良の汎用衝突探索アルゴリズムは CNS のアルゴリズムである（4.4 節）．Case 1a-1c では，実行時間と使用する計算機のサイズのトレードオフによって効率性が評価される．利用可能な量子計算機および古典メモリのサイズが同一という条件下では，Case 1a および Case 1b における現状で最も速い汎用衝突探索アルゴリズムは並列 rho 法である（4.3 節）．また Case 1c においては利用可能な量子計算機および古典メモリのサイズに応じて最良のアルゴリズムが変化する．表 1 に，計算量のうち重要なものをまとめておく．

表 1　各ケースにおける衝突攻撃に必要な計算リソース．オーダー記号は省略している．Case 1a-1c については，のちに 8 章で紹介する特定のハッシュ関数に対する（汎用でない）衝突攻撃の議論において Case 1a が最も重要になるため，Case 1a のみ記載した．

設定	時間	計算用量子プロセッサの大きさ・数	古典メモリ	量子メモリ
Case 0	$2^{n/3}$	$\mathrm{poly}(n)$	$2^{n/3}$	$2^{n/3}$
Case 1a	$2^{n/2}/S$	S	S	S
Case 2	$2^{2n/5}$	$\mathrm{poly}(n)$	$2^{n/5}$	$\mathrm{poly}(n)$

今後量子コンピュータの研究開発がどのように進展していくかはわからないということ，また共通鍵暗号技術では実際のところ安全性パラメータ n は $n = 128$ などに固定されており，“n について指数的に大きいサイズ” と “n について高々多項式的程度の小さいサイズ” の区別も曖昧である（例

[*10] この分類は Case 0 以外，CT-RSA 2018 における細山田と佐々木の分類 [HS18a] に従っている．細山田と佐々木の分類は多重原像探索攻撃の効率性評価を念頭においたものであるが，衝突探索攻撃の効率性評価でも同じ分類を使うことができる．

えば $n = 128$ なら $2^{n/3} \approx n^6$ である）ことから，できるだけ様々な状況を想定して攻撃の研究をし安全性を評価しておくことが望ましい．

4.6 その他の議論

ここまで紹介した既存研究では簡単のため量子誤り訂正のコストや実際の物理的ハードウェアの実現法を無視し，量子回路の実行時間が回路の深さに比例するとみなして実行時間について論じられていた．しかし，量子誤り訂正のコストや実際の物理的ハードウェアの実現法を考慮に入れると暗号に対する量子アルゴリズムを用いた攻撃の実行コストは量子回路中で使用される量子ゲートの個数あるいは量子回路の幅と深さの積で図るべきである，という議論も存在する．このような議論の詳細については，例えば Jaques と Schanck の論文 [JS19] を参照されたい．

5　汎用量子攻撃

本章では，暗号技術の内部構造に関わらず適用できる[*11]ような汎用量子攻撃について，既存の主な研究結果を紹介する．

5.1　Grover のアルゴリズムを用いた鍵回復攻撃と原像探索

2 章の注意 2.2 で触れた，Grover のアルゴリズムを用いた鍵回復攻撃と原像探索の詳細について述べる．

原像探索問題はほぼ問題 2.2 の $t = 1$ の場合そのものであるため，n ビット出力ハッシュ関数の原像探索は時間 $2^{n/2}$ で実行可能である．

以下，秘密鍵の全数探索について，ブロック暗号の場合を例に取って説明する．E を鍵長 k ビット，ブロック長 n ビットのブロック暗号とする．まず，平文 P と対応する暗号文 $C = E_k(P)$ のペア (P, C) を $\ell := \lceil k/n \rceil$ 個集める．集めたペアを $(P_1, C_1), \ldots, (P_\ell, C_\ell)$ とする．次に関数 $f : \{0,1\}^k \to \{0,1\}$ を，$E_X(P_i) = C_i$ が全ての $1 \leq i \leq \ell$ について成り立つとき，またその時に限って $f(X) = 1$ となるように定義する．ブロック暗号 E が十分にランダムであれば，$X = K$ のとき $f(X) = 1$，$X \neq K$ のとき $f(X) = 0$ となる．よって Grover のアルゴリズムを f に適用すれば，秘密鍵 K を時間 $O(2^{k/2})$ で発見できる．必要な量子ビットは $\tilde{O}(1)$ となる．計算量を表 2 にまとめる．

表 2　Grover のアルゴリズムを用いた鍵回復攻撃と原像探索に必要な計算量．k は秘密鍵の鍵長，n は関数の出力長である．原像探索のデータ・メモリで 1 と書いているのは，必要なデータは原像を求める値一つだけであり，またメモリはその値を蓄えるためのものだけであるという意味である．

攻撃の種類	適用先	時間	データ	（量子）メモリ
鍵回復	秘密鍵を用いる任意の暗号技術	$O(2^{k/2})$	$O(k)$ ビット	$O(k)$ ビット
原像探索	ハッシュ関数など	$O(2^{n/2})$	1	1

5.2　衝突探索および関連する問題

4 章で述べたように，衝突探索問題については使用可能な量子計算のリソースに関する想定に応じて様々な量子アルゴリズムが存在する．

n ビット出力の十分にランダムな関数の衝突を探索するとき，BHT のアルゴリズム (4.2 節) は時間 $O(2^{n/3})$ で衝突を発見し，関数を評価する回数（関数へクエリする回数）も $O(2^{n/3})$ であるが，大きさ $O(2^{n/3})$ の量子メモリを必要とする．

多項式サイズの小さな（古典または量子）計算用プロセッサが 2^s 個あって互いに通信を取り合い

[*11] スポンジ関数のキャパシティ部分の衝突を見つける攻撃（5.2 節）やノストラダムス攻撃（5.5 節）はハッシュ関数の内部構造を若干利用していると見れなくもないが，便宜上汎用攻撃に含めるものとする．

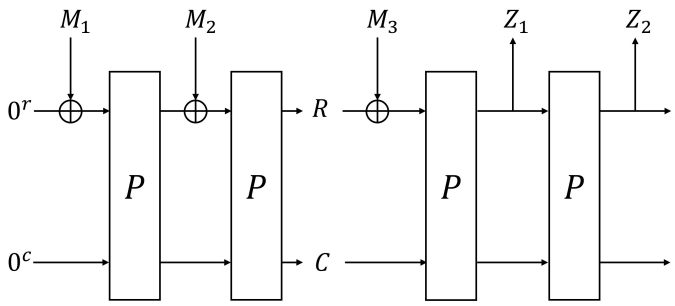


図 4　スポンジ構造．各 M_i と Z_i は r ビット．$Z_1||Z_2$ が出力であるとする．説明を簡単にするため，入力長は 3 ブロックで固定，パディングは無いものとする．

ながら並列計算を行える場合，時間 $O(2^{n/2-s})$ で衝突探索が可能であるが，関数を評価する回数は約 $O(2^{n/2})$ 回となる (4.3 節).

（通常の古典計算機に加えて）n の多項式サイズの小さい量子計算機のみが使える場合でも，CNS のアルゴリズムを用いると時間 $\tilde{O}(2^{2n/5})$ で衝突を探索することができる (4.4 節)．なお大きさ $\tilde{O}(2^{n/5})$ の古典メモリが必要である．

汎用衝突探索アルゴリズムの詳細は 4 章を参照されたい．

■スポンジ構造・XOF　SHA-3 などスポンジ構造を採用しているハッシュ関数は，内部状態のキャパシティの部分で衝突を見つけられれば出力の衝突を見つけることができる．例えば図 4 の関数において, C で示した部分の値が一致するような $M_1||M_2$ と $M_1'||M_2'$ を見つけられたとする．このとき対応するレート部分の値を R および R' とおくと，メッセージ $M = M_1||M_2||R$ と $M' = M_1'||M_2'||R'$ は関数の出力値が一致し，よってこのスポンジ関数の衝突となる．

キャパシティの部分で衝突を見つけるのに必要な古典計算量は $O(2^{c/2})$ である．よって，スポンジ構造の出力長を ℓ とすると，衝突を見つけるのに必要な古典計算量は $O(\min(2^{c/2}, 2^{\ell/2}))$ となる．これは特に出力長を自由に設定できる XOF において重要で，ℓ を非常に大きくしたとき衝突探索にかかる時間が $O(2^{\ell/2})$ ではなく $O(\min(2^{c/2}, 2^{\ell/2})) = O(2^{c/2})$ となる．

量子計算機を用いる場合も同様で，例えば BHT のアルゴリズムを用いる場合だと，スポンジ構造のハッシュ関数の衝突を探索するのに必要な計算時間と量子メモリは双方とも $O(\min(2^{c/3}, 2^{\ell/3}))$ になる．

5.2.1　大量の衝突を探索する問題

n ビット出力のランダム関数 f があってこの衝突を 2^k ペア探す，という問題を考える（定義域は十分大きく k はさほど多くないとする）．最も単純な探索の仕方は，単に衝突探索アルゴリズムを 2^k 回繰り返すというものであり，古典的には f の評価回数（クエリ回数）が $O(2^{n/2+k})$ 必要である．しかし，少し工夫を加えると f の評価回数を $O(2^{(n+k)/2})$ まで下げられる．これが量子計算機を用いると，f の評価回数を更に $O(2^{(n+2k)/3})$ まで下げられる [BCSS23].

5.2.2 多重衝突探索問題

関数 f の衝突というとペア (x_1, x_2) であって $x_1 \neq x_2$ かつ $f(x_1) = f(x_2)$ となるものを指すが，それを拡張した概念として関数 f の多重衝突がある．整数 $\ell \geq 2$ に対して関数 f の ℓ-多重衝突とは，組 $(x_1, \ldots, x_\ell)$ であって $i \neq j$ なる任意の i と j について $x_i \neq x_j$ が成り立ち，かつ $f(x_1) = \cdots = f(x_\ell)$ が成り立つものである．古典計算においてランダム関数 $f : \{0,1\}^m \to \{0,1\}^n$ の ℓ-多重衝突を探索するのに必要な（f への）クエリ回数は $\Theta(2^{(\ell-1)n/\ell})$ となることが知られている [STKT08]．これに対し，量子計算機を用いてランダム関数 f の ℓ-多重衝突を探索するのに必要な（量子）クエリの回数は $\Theta\left(2^{\frac{2^{\ell-1}-1}{2^\ell-1}n}\right)$ まで下がることが示されている [HSX17, HSTX19, LZ19].

5.2.3 k-XOR 問題

多重衝突探索問題に似た問題として k-XOR 問題(与えられた関数 f に対し, $f(x_1) \oplus \cdots \oplus f(x_k) = 0$ を満たす組 $(x_1, \ldots, x_k)$ を探す）があるが，これについても量子計算機を用いればある程度の高速化が得られることが示されている [CE05, GNS18, NS20, Sch21].

5.3 多重原像探索

2.1 節で紹介した多重原像探索問題（問題 2.2）を考える．つまり，ランダム関数 $F : \{0,1\}^n \to \{0,1\}^n$（量子オラクルとして与えられる）と $L \subset \{0,1\}^n$ が与えられたとき，$F(x) \in L$ となるような x を探索することを考える．

2.1 節で紹介した，Grover のアルゴリズムを自明に適用することによって得られる多重原像探索アルゴリズムは，F へのクエリ回数 $\sqrt{2^n/|L|}$, 時間 $\sqrt{2^n/|L|}$ で原像 x を見つけるというものであった．このアルゴリズムは $O(|L|)$ の大きさの量子メモリ（QRAM）を必要とする（4.5 節の分類で言うところの Case 0 にあたる）.

Bernstein と Banegas は Case 1a と Case 1b において，サイズ 2^s の量子計算機を用いた場合にそれぞれ時間 $O\left(\sqrt{\frac{2^n}{|L|\cdot 2^s}}\right)$ および $O\left(\sqrt{\frac{2^n}{|L|^{1/2}\cdot 2^s}}\right)$ で原像を発見できることを示した [BB17]．なおこの計算量は $2^s \geq |L|$ のときのみ有効である．

Chailloux らは Case 2（サイズが高々 n の多項式の小さな量子計算機が一つ利用可能）において，時間 $\tilde{O}\left(2^{n/2-\ell/6}\right)$ で原像を発見することが出来ることを示した [CNS17]．ここで $\ell := \log |L|$ である．またこの計算量評価は $\ell \leq 3n/7$ であるときに限り有効で，サイズ $\tilde{O}\left(2^{\ell/3}\right)$ の古典メモリを使用する．

また Chailloux らは Case 1c において，サイズが高々 n の多項式の小さな独立した量子計算機がそれぞれ 2^s 個使用可能であるとき，時間 $\tilde{O}\left(2^{n/2-\ell/6-s/2}\right)$ で原像を探索することが可能であることを示した [CNS17]．なおこの計算量評価は $\ell \leq (3n+3s)/7$ であるときに限り有効で，サイズ $\tilde{O}\left(2^{\ell/3}\right)$ の古典メモリを使用する．

5.4 Hellman の時間メモリトレードオフとレインボーテーブル

ランダムな関数 $H : \{0,1\}^n \to \{0,1\}^n$ の原像探索問題を解く，つまりランダムな y が与えられたときに $H(x) = y$ なる x を見つけるには，古典で時間 $\Omega(2^n)$ を要する．

しかし，これは攻撃者が事前に（y を与えられる前に）何も準備をしていなかった場合の話である．もしも攻撃者が y を与えられる前にペア $(x, H(x))$ を全ての x について計算しメモリに保存しておけば，原像探索問題は $O(1)$ で解ける．つまり，攻撃者が事前に何らかの計算をして H に依存する情報をメモリに保存しておけば，原像探索問題は時間 $O(2^n)$ よりもずっと早く解けるわけである．事前情報を蓄えるメモリサイズを S として，$S = 0$ が事前計算の無い通常の原像探索に対応し，$S = 2^n$ が全ての x に対して $(x, H(x))$ を計算し保存しておく状況に対応する．

攻撃に使えるメモリは少なくとも 0 ではないと想定するのが自然であるが，$S = 2^n$ ものメモリを使えるとも限らない．では使えるメモリのサイズ S が 0 と 2^n の間にある際，原像探索にかかる時間 T はどうなるだろうか？なお，メモリサイズ S の大小に限らず，事前計算に使える時間に制限は無いとする．

この問題を解くのに使える手法が，Hellman の時間メモリトレードオフ攻撃 [Hel80] と Oechslin のレインボーテーブル [Oec03] である．いずれも，時間とメモリのトレードオフ $T = O((2^n/S)^2)$ を与える．

Dunkelman らは，量子計算機を用いればトレードオフが上述の $T = O((2^n/S)^2)$ から $T = O((2^n/S)^{1.5})$ まで改善されることを示した [DKRS24]．以下，この高速化がどうやって得られるかについて説明を行う．高速化のアイデアの根っ子は，Hellman の攻撃とレインボーテーブルでほぼ同じである．レインボーテーブルの方が Hellman の攻撃より説明が簡潔に済むため，レインボーテーブルに焦点を当てる．

まずは古典的なレインボーテーブルを用いた手法の概要振り返ったのち，Dunkelman らのアイデアを説明する．最後に，時間とメモリだけでなくデータも含んだトレードオフについて紹介する．なお本節において量子計算リソースの設定は 4 章の Case 0，すなわち多項式サイズの小さい計算用量子プロセッサと指数的に大きなサイズの QRAM があると想定する．

5.4.1 レインボーテーブルによる時間メモリトレードオフ（古典）

まず，正の整数 t と m を適当に取る．$i = 1, \ldots, t$ に対して適当な可逆関数 $L_i : \{0,1\}^n \to \{0,1\}^n$ を取り（ビットの入れ替えなど），$f_i(x) := H(L_i(x))$ とおく．事前計算として，以下のプロセスを実行する．

■事前計算

1. $w_1, \ldots, w_m \in \{0,1\}^n$ をランダムに取る．
2. $i = 1, \ldots, m$ に対して $z_i = f_t(f_{t-1}(\cdots f_1(w_i)))$ を計算し，ペア (w_i, z_i) を保存する．

事前計算で，m 個のペアがメモリに蓄えられることになる．事前計算のあと，攻撃者は $y \in \{0,1\}^n$

を与えられる．メモリに蓄えたデータを以下のように利用し，以下の要領で $H(x) = y$ なる x を探索する．

■オンライン計算

1. $j = 1, \ldots, t$ に対して順に以下を実行：
   (a) $z' = f_t(f_{t-1}(\cdots f_j(y)))$ を計算する．
   (b) いずれかの i について $z' = z_i$ となる場合：高確率で $y = f_{j-1}(f_{j-2}(\cdots f_1(w_i)))$ が成り立ち，特に $x := L_{j-1}(f_{j-2}(\cdots f_1(w_i)))$ とおけば $H(x) = y$ を満たすはずである．そこで，この x を出力してアルゴリズムを終了する．
   (c) 全ての i について $z' \neq z_i$ の場合：何もせず次の j へ．

パラメータ m と t が $m \times t \approx 2^n$ を満たしていれば，上記の攻撃は高確率で成功する．メモリの大きさ S は m に等しく，またオンライン計算にかかる時間は $T \approx t^2$ である．ゆえに $m \times t \approx 2^n$ が満たされているとき，トレードオフ $T \approx (2^n/S)^2$ が成り立つ．例えば $t = 2^{n/3}$ かつ $m = 2^{2n/3}$ なら，$T = S = 2^{2n/3}$ となる．

5.4.2 Dunkelman らによる量子高速化 [DKRS24]

上述のアルゴリズムにおけるオンライン計算では，ステップ (a)-(c) を全ての $j = 1, \ldots, t$ について行っている．これは当然ながら，どの j が当たりか，つまり「ある i が存在して $z' = f_t(f_{t-1}(\cdots f_j(y))) = z_i$ が成り立つ」というような j が一体どれなのか，全通りチェックしないとわからないからである．

Dunkelman らのアイデアは，この j の探索に Grover のアルゴリズムを用いて，オンライン計算を高速化しようというものである（事前準備は古典のときと同じである）．より具体的には，Bool 関数 $F : \{1, \ldots, t\} \to \{0, 1\}$ を

$$F(j) := \begin{cases} 1 & (\text{ある } i \text{ が存在して } f_t(f_{t-1}(\cdots f_j(y))) = z_i \text{ が成り立つとき}) \\ 0 & (\text{それ以外のとき}) \end{cases}$$

と定義し，この F に Grover のアルゴリズムを適用する．関数 F 自体の計算は

1. $z' = f_t(f_{t-1}(\cdots f_j(y)))$ を計算する．
2. 事前計算でメモリに蓄えられたデータを検索し，$z' = z_i$ となる i があるかチェックする

とすれば時間 $O(t)$ で可能である．Grover のアルゴリズムを F に適用すると，F の定義域サイズが t なので，Grover のアルゴリズムが F を呼び出す回数は $O(\sqrt{t})$ 回となる．よって，オンライン計算にかかる時間は $T = O(t) \times O(\sqrt{t}) = O(t^{3/2})$ となる．

攻撃成功に必要な条件が $m \times t \approx 2^n$ であったこと，$S = m$ であること，および $T \approx t^{3/2}$ より，トレードオフとしては

$$S \times T^{2/3} \approx 2^n$$

あるいはこれを整理して

$$T = O((2^n/S)^{3/2})$$

が得られる．例えば $S = 2^{3n/5}$ のとき $T = O(2^{3n/5})$ となる．

注意 5.1. 本節における説明は Dunkelman らのアイデアの要点を手短に説明することが目的であり，様々な部分で細かい説明を省略している．成功確率の評価，distinguished point に関する議論，Grover のアルゴリズムで呼び出す関数 F の量子回路としての実装（入力によらず計算時間が同じで，かつ計算が可逆である必要がある）等々，詳細は本節で紹介した原論文や関連研究を参照されたい．

5.4.3 時間・メモリ・データのトレードオフ

ここまでは，時間とメモリのトレードオフを与える古典アルゴリズムおよびその量子計算機を用いた高速化について説明した．しかし原像探索について述べたのみで，多重原像探索については触れていなかった．

多重原像探索問題では複数のターゲット $y_1, \ldots, y_D$ が与えられ，どれか一つの y_i について $H(x) = y_i$ となる i を見つけられれば良い．このようにターゲットが複数ある場合は，時間とメモリのみならずデータ量（つまりターゲットの個数 D）も含んだトレードオフ

$$T = O\left(\left(\frac{2^n}{S \cdot D}\right)^2\right)$$

が古典的に得られることが知られている（ただし $T \geq D^2$ のときのみ有効である）．例えば $D = 2^{n/4}$ かつ $S = 2^{n/2}$ のとき $T = 2^{n/2}$ となる．トレードオフは，前節で触れた Hellman の手法あるいはレインボーテーブルを用いた手法を改良することで得られる [BS00, BMS05, BBS06].

Dunkelman らの論文 [DKRS24] はこのデータを含むトレードオフについても量子計算による高速化を示している．具体的には

$$T = O\left(\left(\frac{2^n}{S \cdot D}\right)^{3/2}\right)$$

となる（$T \geq D^{1.5}$ のときのみ有効である）．例えば $D = 2^{2n/7}$ かつ $S = 2^{3n/7}$ のとき $T = 2^{3n/7}$ となる．高速化のアイデアは前節と同様で，オンライン計算の一部を上手く全探索とみなして Grover のアルゴリズムを適用するというものである．詳細は原論文 [DKRS24] を参照されたい．

5.5 ノストラダムス攻撃

Merkle-Damgård 構成のハッシュ関数に対するノストラダムス攻撃というものについて説明する．アリスとボブという二人の人がおり，アリスは予知能力を持っていて，来月当選発表がある宝くじの 1 等の番号 X が予測できるとする．アリスはこの能力があることを知人のボブに証明したいが，自分で宝くじを買って儲けるつもりはなく，またボブがアリスの能力を使って儲けるのも嫌だとする．このとき，安全な暗号学的ハッシュ関数 H を以下のように使えば，アリスは予知能力があるこ

とを証明できるのではないかと考えられる[*12].

1. アリスはランダム文字列 R を選んで $X||R$ をハッシュ関数 H にかけ，出力 $y = H(X||R)$ を計算して予言の証拠としてボブに渡す.
2. 一か月後, 宝くじの当選番号が発表される．アリスは値 R をボブに渡し，ボブは対応するハッシュ値 $z = H(X||R)$ を計算する．$y = z$ ならアリスの予言は正しかったことが確かめられる.

ここで，ハッシュ関数 H が安全であるということが重要である．もし H が安全でなければ,「アリスは実は予知能力が無く，y としてでたらめな値を選んでボブに渡し，当選番号 X の発表後に $y = H(X||R)$ を満たす R を求めてボブに渡していた」という可能性が排除できなくなってしまうからである.

裏を返せば，そのような R を見つけられるのであれば，予知能力がないのに予言者のふりができるかもしれない．これがノストラダムス攻撃である．より正確には，以下の状況を考える.

1. 攻撃者は何らかの値 y を事前に計算する.
2. X が選ばれ，攻撃者に与えられる.
3. 攻撃者は $H(X||R) = y$ を満たす R を求める.

古典的な結果として，Merkle-Damgård 構成によって作られた n ビット出力ハッシュ関数の場合，圧縮関数の評価 $O(2^{2n/3})$ 回でこの攻撃が成功することが知られている [KK06, BSU12] が，量子計算の場合はこの評価回数が $O(2^{3n/7})$ まで下げられることが示されている [BFH22].

5.6 汎用量子攻撃の具体的なコスト評価

AES や SHA-2・SHA-3 などの代表的な共通鍵暗号技術に対して Grover のアルゴリズムを用いた鍵全数探索などの汎用攻撃を実行するのに必要なコストを，攻撃対象のプリミティブを量子回路上へ実装する際のコストも込みで具体的に見積もろうという研究もなされている [GLRS16, ASAM18, JNRV20, LPS20, AMG+16, LPZW23, LGQW23, ZWS+20, HS22, CLF+24, ZSWS24, Pre22, KHJ18, LKL+24, LLLC23]．例えば，深さ高々 2^{75}，幅高々 2^{13} 量子ビットの回路に，高々 2^{83} 個の Clifford+T ゲートを使用することで Grover のアルゴリズムを用いた AES-128 への鍵回復攻撃を実装できることが示されている [LPZW23].

[*12] これはいわゆるコミットメントをしようとしているわけであるが，この方式でコミットメント方式としての安全性証明がつくかどうかは考えない．あくまで攻撃を説明する都合上このような状況を例にとっているだけである.

6　量子クエリ攻撃 (Q2)

本章では Q2 モデルにおける攻撃，すなわち攻撃者が量子計算機を所有していることに加え秘密鍵の埋め込まれたオラクルへ量子クエリを行えるという状況下での攻撃について，これまでに発表されている主な結果を紹介する．

6.1　Even-Mansour 暗号への鍵回復攻撃

P を n ビットの公開置換とする．Even-Mansour 暗号 [EM91] の暗号化関数は，その暗号化関数が P および 2 つの n ビット鍵 K_1, K_2 を用いて $E_{K_1,K_2}(M) = P(M \oplus K_1) \oplus K_2$ と定義されるブロック暗号である（図 5 を参照）．P がランダム置換であるという理想化された状況において，

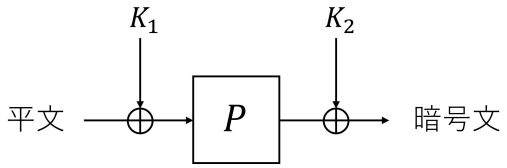


図 5　Even-Mansour 暗号

Even-Mansour 暗号は多項式時間の古典攻撃に対し安全な強擬似ランダム置換（SPRP）であることが証明されている．しかし桑門と森井は，Q2 モデルにおいては Even-Mansour 暗号の鍵を多項式時間で回復できるということを示した [KM12]．以下攻撃の概要を述べる．

まず関数 $f : \{0,1\}^n \to \{0,1\}^n$ を $f(x) := E_{K_1,K_2}(x) \oplus P(x)$ と定義する．すると，

$$\begin{aligned} f(x \oplus K_1) &= E_{K_1,K_2}(x \oplus K_1) \oplus P(x \oplus K_1) \\ &= P(x \oplus K_1 \oplus K_1) \oplus K_2 \oplus P(x \oplus K_1) \\ &= P(x) \oplus K_2 \oplus P(x \oplus K_1) \\ &= f(x) \end{aligned}$$

が成り立ち，f は秘密鍵 K_1 を周期に持つ周期関数である．Q2 モデルでは，攻撃者に暗号化関数 E_{K_1,K_2} の量子オラクルが与えられている．また P は公開置換であるので，攻撃者は置換 P の値を量子重ね合わせで評価することが出来る．よって攻撃者は関数 f の値も量子重ね合わせで評価することが出来る（f の量子オラクルをシミュレートすることが出来る）．P が十分にランダムであれば，攻撃者は Simon のアルゴリズムを適用することにより多項式時間で K_1 を回復することが出来る[*13]．一旦 K_1 を回復することができれば，$K_2 = E_{K_1,K_2}(x) \oplus P(x \oplus K_1)$ が全ての x について成り立つため，K_2 も容易に計算できる．以上が桑門と森井による Even-Mansour 暗号への鍵回復攻撃の概要である．

[*13] P が十分にランダムでないと Simon のアルゴリズムを適用しても K_1 を回復することはできない．たとえば P が恒等置換である場合，$f(x) = K_1 \oplus K_2$ が全ての x について成り立ってしまう．このとき Simon のアルゴリズムは K_1 を計算することができない．しかし P がランダムなら Simon のアルゴリズムは K_1 を返すということが示されている [KLLN16a].

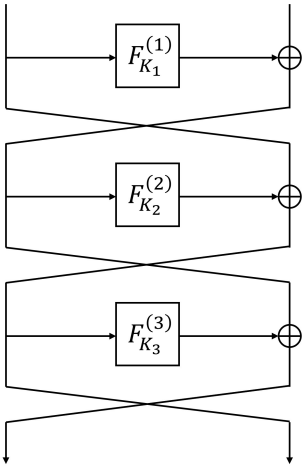


図 6　3 段 Feistel 暗号

Even-Mansour の構造を持つ暗号技術として，Chaskey [MMH+14] が挙げられる．Chaskey 自体はブロック暗号ではなくメッセージ認証コードだが，メッセージ長が短いときの構造は本質的に Even-Mansour 暗号であり，上記の攻撃を適用できる．

6.2　Feistel 暗号（Luby-Rackoff 構成）への識別攻撃

本節では桑門と森井による 3 段 Feistel 暗号（3 段 Luby-Rackoff 構成）[LR85] への識別攻撃 [KM10] の概要を述べる．

r を正整数とする．鍵付き関数 $F_{K_i}^{(i)} : \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ が $i = 1, \cdots, r$ について与えられているとき，r 段 Feistel 暗号（あるいは r 段 Luby-Rackoff 構成）は暗号化関数 $\mathrm{Enc}_{K_1,\ldots,K_r}$ が平文 $x_L||x_R \in \{0,1\}^n$（$x_L, x_R \in \{0,1\}^{n/2}$）に対し以下のようにして定められるブロック暗号である：

$$\mathrm{Enc}_{K_1,\ldots,K_r}(x_L, x_R) := \left(R_{K_r}^{(r)} \circ \cdots \circ R_{K_1}^{(1)}\right)(x_L, x_R),$$

ただしここで

$$R_{K_i}^{(i)}(x_L, x_R) = \left(x_R \oplus F_{K_i}^{(i)}(x_L)\right)||x_L.$$

$r = 3$ の場合の暗号化関数 $\mathrm{Enc}_{K_1,K_2,K_3}$ を図 6 に示す．Feistel 暗号の構造は DES [Nat77] や Camellia [AIK+00] を初めとした様々なブロック暗号に採用されている．以下，簡単のため鍵付き関数の鍵長は全て同じであるとする．

各 $F_{K_i}^{(i)}$ が多項式時間の古典攻撃に対して安全な擬似ランダム関数（PRF）のとき，3 段 Feistel 暗号は多項式時間の古典選択平文攻撃に対して安全な擬似ランダム置換（PRP）になり，また 4 段 Feistel 暗号は多項式時間の古典選択暗号文攻撃に対して安全な強擬似ランダム置換（SPRP）になることが証明されている [LR85]．一方桑門と森井は，たとえ各 $F_{K_i}^{(i)}$ が多項式時間の量子クエリ攻撃に対して安全な擬似ランダム関数であったとしても，3 段 Feistel 暗号を多項式時間の量子選択平文攻撃によって n ビットランダム置換から識別する攻撃アルゴリズムが存在する（つまり 3 段 Feistel

暗号は量子擬似ランダム置換（qPRP）ではない）ことを示した．以下，桑門と森井による量子識別攻撃の概要を述べる．

まず識別攻撃の設定を説明する．攻撃者には n ビット置換 Π の量子オラクルが与えられている．Π は 3 段 Feistel 暗号の暗号化関数 $\mathrm{Enc}_{K1,K2,K3}$ あるいは n ビットランダム置換 RP のいずれかである．攻撃者の目的は Π が $\mathrm{Enc}_{K1,K2,K3}$ と RP のいずれであるかを識別することである．

桑門と森井による攻撃では，まず $\alpha_0, \alpha_1 \in \{0,1\}^{n/2}$ であって $\alpha_0 \neq \alpha_1$ となるものを任意に取って固定し，関数 $f^{\Pi} : \{0,1\} \times \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ を

$$f^{\Pi}(b,x) := \Pi(\alpha_b, x)_R \oplus \alpha_b$$

と定義する．ただしここで $\Pi(\alpha_b,x)_R$ は $\Pi(\alpha_b,x)$ の下位 $n/2$ ビットである．

$\Pi = \mathrm{Enc}_{K_1,K_2,K_3}$ ならば，

$$f^{\Pi}(b,x) := \mathrm{Enc}_{K1,K2,K3}(\alpha_b,x)_R \oplus \alpha_b = F^{(2)}_{K_2}\left(F^{(1)}_{K_1}(\alpha_b) \oplus x\right)$$

であるから

$$f^{\Pi}\left((b,x) \oplus \left(1, F^{(1)}_{K_1}(\alpha_0) \oplus F^{(1)}_{K_1}(\alpha_1)\right)\right) = f(b,x)$$

が任意の $(b,x) \in \{0,1\} \times \{0,1\}^n$ に対して成り立つことがわかる．特に f^{Π} は，$\left(1, F^{(1)}_{K_1}(\alpha_0) \oplus F^{(1)}_{K_1}(\alpha_1)\right)$ を周期に持つ周期関数である．一方 $\Pi = \mathsf{PR}$ の場合，高確率で f^{Π} は周期的にならない．

よって，f^{Π} が周期をもつかを Simon のアルゴリズムを用いて調べることにより，Π が $\mathrm{Enc}_{K1,K2,K3}$ と RP のどちらであるか多項式時間で識別することができる．以上が桑門と森井による識別攻撃の概要である．

この攻撃は 4 段 Feistel 暗号への量子選択暗号文攻撃による識別攻撃 [IHM+19] や一般化 Feistel 暗号への攻撃にも拡張されている [DLW19, NIDI19, CHLS20, HKK20]．また関連する後続研究として，ラウンド関数 $F^{(i)}_{K_i}$ が特定の構造を持つ状況下での攻撃の研究や，識別攻撃を鍵回復攻撃へ拡張する研究などがある [BNS19a, DW18, HS18b].

また最近の関連する研究としては，この種の識別攻撃を切詰差分と関連させて段数を削減した LBlock[WZ11] や SIMON[BSS+13][*14]に対する識別攻撃を示している研究 [XWY+24]，5 段以上の Feistel 暗号に対する量子ウォークを用いた（指数時間）[CPT23, CCP24]，識別攻撃の発見を自動化する試み [CLS22] などがある．

6.3 CRYPTO 2016 における Kaplan らの結果

CRYPTO 2016 において Kaplan らは，Q2 モデルでは CBC-MAC（XCBC [BR00] や OMAC [IK03]，CMAC [NIS05] などの変種を含む）や GCM [MV04] など現在幅広く使用されている様々な共通鍵暗号技術，特にブロック暗号利用モードが多項式時間で破られることを示した [KLLN16a]．多項式時間で破られることが示された暗号技術は CBC-MAC や GCM の他に，

[*14] NSA が設計したブロック暗号である．Simon のアルゴリズムとは特に関係がない．

PMAC [BR02]，GMAC [MV04]，OCB [RBBK01, Rog04, KR11]，LRW 構成 [LRW02]，などがある．

攻撃のアイデアは Even-Mansour や Feistel への攻撃と同様，周期関数を作って Simon のアルゴリズムを適用するというものである．周期関数の作り方は攻撃対象によって異なるが，攻撃が直接操作できる値（平文，メッセージなど）に，秘密鍵に依存した値（Even-Mansour の場合は K_1，Feistel の場合は 1 段目のラウンド関数の出力）が XOR されていることを利用する．攻撃の詳細は原論文を参照されたい．

Kaplan らはまた同時に，古典的に指数時間を要するスライド攻撃 [BW99] が Q2 モデルにおいて多項式時間まで高速化可能であることも示した[*15]．

6.4 Grover のアルゴリズムと Simon のアルゴリズムの組み合わせ

本節では，Leander と May による FX 構成への Q2 モデルにおける攻撃 [LM17] の概要を紹介する．攻撃は，Grover のアルゴリズムと Simon のアルゴリズムの組み合わせにより実現される．

まず鍵長 k ビットの n ビットブロック暗号 $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ から作られる FX 構成 [KR96] とは，鍵長 $(k+2n)$ ビットの n ビットブロック暗号 $E' : \{0,1\}^{k+2n} \times \{0,1\}^n \to \{0,1\}^n$ であって，

$$E'(K_0, K_1, K_2, x) := E_{K_0}(x \oplus K_1) \oplus K_2 \tag{7}$$

により定義されるもののことである（ここで，$K_0 \in \{0,1\}^k$ かつ $K_1, K_2, x \in \{0,1\}^n$ である．図 7 参照）．古典的には，E が理想的にランダムなブロック暗号であれば，FX 構成 E' をランダム置換と識別するためには暗号化オラクルおよび復号オラクルへおよそ $2^{(m+n)/2}$ 回のクエリを行わねばならないことが証明される．

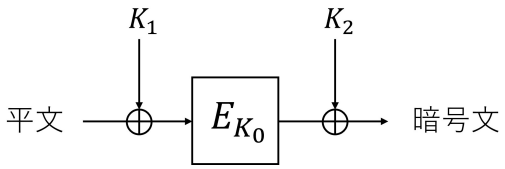


図 7　FX 構成

Q2 モデルにおいて FX 構成の鍵の回復を試みる際，まず自然な発想として思いつくのは，FX 構成が Even-Mansour 暗号に似ているから Simon のアルゴリズムで攻撃できないだろうか，ということである．実際，秘密鍵のうち K_0 がわかっていれば，残りの鍵は Even-Mansour への攻撃と同様 Siomn のアルゴリズムを用いて多項式時間で回復できる[*16]．

このアイデアを用いると，以下のようにして全ての秘密鍵を回復することができる．なお FX 構成 $E'_{K_0,K_1,K_2}(x) = E_{K_0}(x \oplus K_1) \oplus K_2$ の暗号化関数の量子オラクルが与えられているものとする．

1. 全ての $K'_0 \in \{0,1\}^k$ に対して以下のステップ a と b を実行する：

[*15] この結果はのちに advanced slide attack [BW00] の指数的高速化に拡張されている [BNS19a].
[*16] E_{K_0} を 6.1 節における置換 P とみなせばよい.

(a) 関数 $f_{K_0'}$ を $f_{K_0'}(x) := E'_{K_0,K_1,K_2}(x) \oplus E_{K_0'}(x)$ で定義する．($K_0' = K_0$ であれば $f_{K_0'}$ は周期関数になり，また E が理想的にランダムなブロック暗号であれば $K_0' \neq K_0$ のとき $f_{K_0'}$ は周期関数にならないため，$K_0' = K_0'$ かどうかを $f_{K_0'}$ か否かで判定できる．なお f_{K_0} の周期は K_1 である．)

(b) Simon のアルゴリズムを $f_{K_0'}$ へ適用し，$f_{K_0'}$ が周期関数か否か，すなわち $K_0' = K_0'$ であるか調べる．$K_0' = K_0'$ であればステップ 2 へ移る．

2. $f_{K_0'}$ に Simon のアルゴリズムを適用して K_1 を回復する．また $K_2 = E'_{K_0,K_1,K_2}(0^n) \oplus E_{K_0}(K_1)$ であることを用いて K_2 を回復する．

ステップ 1 では 2^k 個の鍵候補を全数探索しており，また Simon のアルゴリズムは多項式時間で実行できることから，この攻撃の実行時間は $\tilde{O}(2^k)$ となる．

ここで，次のアイデアが自然な発想として浮かんでくる：

> 2^k 個の鍵の全数探索を Grover のアルゴリズムで行えば攻撃時間を $\tilde{O}(2^{k/2})$ まで下げられるのではないか？

K_0 を Grover のアルゴリズムで探索するためには関数 $F : \{0,1\}^k \to \{0,1\}$ であって $F^{-1}(1) = \{K_0\}$ となるものを量子重ね合わせで評価できる量子回路を実装する必要がある．関数 F の実装として自然なものは先述したアルゴリズムのステップ 1a-1b，すなわち "$f_{K_0'}$ に Simon のアルゴリズムを適用し，$f_{K_0'}$ が周期関数のとき，またその時に限って $F(K_0') = 1$ と計算する" というものである．

しかしここで Simon のアルゴリズムが量子状態の観測を複数回行うことが問題になる：Grover のアルゴリズムを F に適用する際，F を実装する量子回路は途中での観測を行ってはならない．ところが Simon のアルゴリズムは複数回の量子状態の観測を含むため，F の量子回路をどう構築すれば良いかは自明ではない．

Leander と May は Asiacrypt 2017 において, Simon のアルゴリズムのサブルーチン **SSub**（2.2 節を参照）から最後の観測を除いたものを並列して走らせることで途中の観測なしで F を実装する量子回路を示し，また詳細な誤差解析を行って実際に FX 構成の秘密鍵を時間 $\tilde{O}(2^{k/2})$ で回復できることを証明した [LM17]．攻撃に必要な量子ビットの個数は高々 k と n の多項式で抑えられる．

Leander と May の論文は FX 構成への攻撃しか取り扱っていないが，Grover と Simon の二つのアルゴリズムを組み合わせて攻撃に用いたいという場面では基本的に Leander と May の手法が適用可能である．

6.5 隠れシフト問題と Kuperberg のアルゴリズム

本節では共通鍵暗号技術に対する量子攻撃と隠れシフト問題および Kuperberg のアルゴリズム [Kup05] について説明する．

ブロック暗号などの共通鍵暗号技術に Simon のアルゴリズムを用いた量子攻撃を行う際は, 秘密鍵に依存するようなある秘密情報 $s \in \{0,1\}^m$ と全ての $x \in \{0,1\}^m$ に対して $f_0(x) = f_1(x \oplus s)$ が成り立つような 2 つの関数 $f_0, f_1 : \{0,1\}^m \to \{0,1\}^n$ を, 鍵の埋め込まれたオラクルから構成することが

多い．なぜなら，このような関数 f_0, f_1 を構成できたとすると，関数 $F : \{0,1\} \times \{0,1\}^m \to \{0,1\}^n$ を $F(b,x) := f_b(x)$ と定義すれば F は $(1,s)$ を周期に持つ周期関数になり，Simon のアルゴリズムを F へ適用することで秘密情報 s を得られることが多いからである．

上記の関数 f_1 は関数 f_0 から隠れた（秘密の）情報 s だけ入力がシフトされた関数であると見ることができる．一般に G を有限群，X を集合とし，二つの関数 $f_0, f_1 : G \to X$ が次の条件を満たすとする：或る $g \in G$ があって任意の $x \in G$ に対して $f_0(g) = f_1(g \cdot s)$ が成り立つ[*17]．f_0 と f_1 のオラクルが与えられたときに s を求める問題を隠れシフト問題と呼ぶ．

隠れシフト問題は $G = (\mathbb{Z}/\mathbb{Z}_2)^n$ のときは上述のように Simon のアルゴリズムを用いて効率的に解くことができるが，G が巡回群 $\mathbb{Z}/2^n\mathbb{Z}$ の場合は多項式時間で解けるアルゴリズムが知られていない．$\mathbb{Z}/2^n\mathbb{Z}$ の場合，現時点での最良のアルゴリズムは Kuperberg のアルゴリズム [Kup05] であり，問題を解くのに要する計算量は $\tilde{O}\left(2^{\sqrt{2\log_2(3)n}}\right)$ である．

Alagic と Russell はこの事実に着目し，（ある条件下での）隠れシフト問題を多項式時間で解くことが困難であると仮定して，共通鍵暗号技術で使用される群演算を $(\mathbb{Z}/\mathbb{Z}_2)^n$ の演算（XOR 演算）から $\mathbb{Z}/2^n\mathbb{Z}$ の演算（Modular Addition）に変更すれば，本章でここまでに紹介したような多項式時間攻撃が効かなくなるということを示した [AR17]．

しかしのちに Bonnetain と Naya-Plasencia は，共通鍵暗号技術で実際に使用されるパラメータ n が小さい（ブロック暗号のブロック長としてよく用いられるのは $n = 128$）を考慮すると，Kuperberg のアルゴリズムの計算量 $\tilde{O}\left(2^{\sqrt{2\log_2(3)n}}\right)$ はさほど大きくなく，このような演算の変更は Q2 モデルにおける量子攻撃への対策として必ずしも効果的とは言えないということを指摘した [BN18]．

例えば n ビットブロックの Even-Mansour 暗号について，Simon のアルゴリズムを用いた攻撃（6.1 節参照）を防ぐために XOR 演算を Modular Addition に変更したとしても，Kuperberg のアルゴリズムを用いれば n が 5000 程度であれば時間 2^{128} を下回るような攻撃が可能であると示されている．Bonnetain と Naya-Plasenia は同時に，Kuperberg のアルゴリズムを応用するとメッセージ認証コード Poly1305 [Ber05] を攻撃できるということも示している．

6.6 線形化攻撃

前節で説明した Kaplan らの攻撃は様々なモードに対する多項式時間攻撃を示したが，ISO 標準である LightMAC [LPTY16, ISO19] などへの多項式時間攻撃は見つかっていなかった．

LightMAC はブロック暗号モードであり，構造は図 8 の通りである（赤字・赤枠の部分は一旦無視していただきたい）．パラメータ s は n より小さい値で，i_s は整数 i の s ビット表現であり，また各メッセージブロック M_i は $(n-s)$ ビットのビット列である．メッセージブロック数 ℓ には $\ell \leq 2^s$ の制限がある．説明を簡単にするため，入力メッセージ M の長さが $(n-s)$ の倍数の時のみを考える．M はまず $M = M_0||M_1||\cdots||M_\ell$ とメッセージブロックに分割され，各 M_i は i_s と結合して n ビットのビット列 $i_s||M_i$ に変換される．最後の M_ℓ だけは $1\underbrace{0\cdots0}_{(s-1)\text{ 個}}$ を結合させる．その後ブロック

[*17] ここでは共通鍵暗号技術に対する攻撃への応用を考えるため，f_0 と f_1 および s はランダムに選ばれる状況を考える．

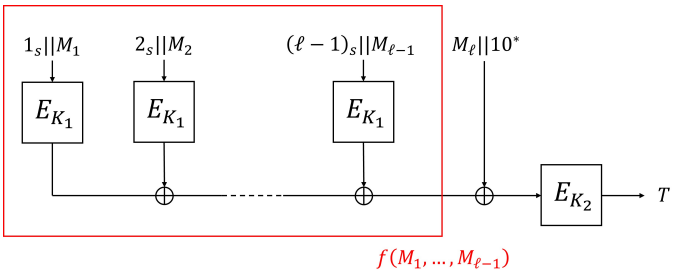


図 8　LigthtMAC.

暗号による暗号化や XOR 演算を図 8 の通りに行った出力が出力タグ T となる[*18].

図 8 を見ると，秘密鍵に依存する情報が最後のブロック M_ℓ へ XOR されているように見える．しかしこの M_ℓ を XOR するタイミングで 10^* の部分は攻撃者が値を操作できず，そのせいで Even-Mansour や Feistel 構造のときのような周期関数を構成できない．

Bonnetain らは，この問題を以下のようにして回避することができることを示した．まず図 8 の赤枠で囲った箇所（つまり $M_\ell||10^*$ を XOR する直前までの部分）を関数とみなして $f(M_1,\ldots,M_{\ell-1})$ とおく．次に任意の相異なる定数 $C_0, C_1 \in \{0,1\}^{n-s}$ を取り，$(\ell-1)$ ビットのビット列 x に対して

$$M_i(x) = \begin{cases} C_0 & x \text{ の } i \text{ ビット目が } 0 \text{ の場合} \\ C_1 & x \text{ の } i \text{ ビット目が } 1 \text{ の場合} \end{cases}$$

と置く．ここで $g : \{0,1\}^{\ell-1} \to \{0,1\}^n$ を

$$g(x) := f(M_1(x),\ldots,M_{\ell-1}(x))$$

で定めると，

$$g(x) = \left(\bigoplus_{1\le i\le \ell-1} (E_{K_1}(i_s||C_0) \oplus E_{K_1}(i_s||C_1)) \cdot x_i \right) \oplus g(0^{\ell-1})$$

が成り立ち，よって $g(x)$ はアフィン関数になることがわかる．特に，適当な行列 A と定数 c があって

$$g(x) = Ax \oplus c$$

と書ける．次に関数 $G : \{0,1\}^{\ell-1} \to \{0,1\}^t$ を

$$G(x) := \mathrm{LightMAC}(M_1(x),\ldots,M_{\ell-1}(x), C_0)$$

で定めると

$$G(x) = E_{K_2}(g(x) \oplus (C_0||10^*)) = E_{K_2}(Ax \oplus c')$$

となる（c' は何らかの定数）．ℓ を適当に大きくすれば（例えば $\ell = 2n$）線形写像 $x \mapsto Ax$ は必ず非自明なカーネルを持つ．カーネルの任意の元 $s \neq 0$ と任意の x に対して $A(x \oplus s) = Ax$ が成り立ち，ゆえに

$$G(x \oplus s) = E_{K_2}(A(x \oplus s) \oplus c') = E_{K_2}(Ax \oplus c') = G(x)$$

[*18] 本来であれば T のいくつかのビットを切り詰めたりするが，説明を簡単にするため省略する．

より，G は周期関数となる．ゆえに Simon のアルゴリズムを用いて LightMAC をランダム関数から識別できる．

Bonnetain らは線形化のアイデアを PMAC+ [IMPS17] や ZMAC [Yas11] などにも適用し，多項式時間攻撃を示している．また同じ論文で，Shor のアルゴリズム [Sho94] を応用した Poly1305 への攻撃なども示している．

6.7 関連鍵攻撃

古典的な関連鍵攻撃の設定として，E_K を k ビット鍵の n ビットブロック暗号としたとき（$K \in \{0,1\}^k$ は秘密鍵），入力 $(x, M) \in \{0,1\}^k \times \{0,1\}^n$ に対して $E^{-1}_{K\oplus x}(M)$ を返すオラクル $\mathcal{O}_K$ と，入力 $(x, C) \in \{0,1\}^k \times \{0,1\}^n$ に対して $E^{-1}_{K\oplus x}(C)$ を返すオラクル $\mathcal{O}^{-1}_K$ が攻撃者に与えられる，というものがある [WH87]．E が理想的にランダムなブロック暗号であれば，古典的にはこの設定で秘密鍵 K を回復するのに指数時間を必要とする．

一方 Rötteler と Steinwandt は，$\mathcal{O}_K$ の量子オラクルが与えられていれば以下のようにして秘密鍵 K を多項式時間で回復できることを示した [RS15]：$M \in 0, 1^n$ を任意に固定し，関数 f を $f(x) := \mathcal{O}_K(x, M) \oplus E_x(M) = E_{x\oplus K}(M) \oplus E_x(M)$ と定義する．すると f は明らかに秘密鍵 K を周期に持つ周期関数であり，Simon のアルゴリズムを適用することによって K を多項式時間で回復することが出来る．

この攻撃はほぼ全ての（古典的に安全な）ブロック暗号に適用可能なものであり，ゆえに理論上興味深いものではあるが，$\mathcal{O}_K$ の量子オラクルが攻撃者に与えられるような状況が現実的に起こることは想定しづらい．

Rötteler と Steinwandt による上記の攻撃は鍵 K 全体へ差分を自由に入れられるというものであるが，もう少し特殊な設定における関連鍵攻撃の研究も行われている [HA17, CHLS20].

6.8 その他の古典攻撃の高速化

量子計算機を用いると様々なアルゴリズムが高速化され得るため，代表的な古典攻撃が量子計算機を用いた際どれだけ高速化できるかということは，たとえ指数的高速化が得られずとも重要な研究の対象となる．量子クエリが行える状況下（Q2 モデル）で，古典攻撃の高速化に関する前節までに挙げたもの以外の主な研究結果としては，差分解読法・線型解読法の高速化 [KLLN16b] などが挙げられる．なお [KLLN16b] で論じられている差分解読法・線型解読法の量子版は，対応する古典攻撃でかかる時間を T としたとき，大雑把に言って $\sqrt{T}$ あるいはそれ以上の時間を要する．

また最近の研究の流れとして，線形解読法や高速相関攻撃などの古典的な攻撃手法に現れる離散フーリエ変換をうまく量子フーリエ変換に対応づけようというものがある [Sch23, Hos23, Hos24].

7　古典クエリ攻撃 (Q1)

本章では Q1 モデルにおける攻撃，すなわち攻撃者が量子計算機を所有しているが攻撃者に与えられる鍵の埋め込まれたオラクルは古典オラクルであるという状況下での攻撃について，これまでに発表されている主な研究結果を紹介する．

7.1　桑門・森井による Even-Mansour 暗号への鍵回復攻撃

6.1 節で紹介した Q2 モデルにおける Even-Mansour 暗号への多項式時間攻撃は秘密鍵の埋め込まれたオラクルへの量子クエリを必要とするため，Q1 モデルでは実行できない．しかし桑門と森井は，Q1 モデルにおいても量子衝突探索アルゴリズム[*19]を用いれば時間 $\tilde{O}(2^{n/3})$ で鍵を回復できることを示した [KM12]．以下その概要を述べる．

Even-Mansour 暗号の暗号化関数は，公開置換 P と秘密鍵 K_1, K_2 を用いて $E_{K1,K2}(M) := P(M \oplus K_1) \oplus K_2$ と定義されるのであった．まず, 関数 $h : \{0,1\}^n \to \{0,1\}^n$ を $h(x) := E_{K_1,K_2}(x) \oplus E_{K_1,K_2}(\bar{x})$ で定義する．ここで $\bar{x}$ はビット列 x の各ビットを反転したもの，つまり $\bar{x} = x \oplus 1^n$ である．更に関数 $g : \{0,1\}^n \to \{0,1\}^n$ を $g(x) := P(x) \oplus P(\bar{x})$ で定義する．すると $h(x \oplus K_1) = g(x)$ が全ての x について成り立つ．更に，P が十分にランダムであれば $h(x) = g(y)$ のとき高確率で $x = y \oplus K_1$ または $x = \bar{y} \oplus K_1$ となることが期待できる．よって，$h(x) = g(y)$ となるペア (x, y) を見つければ（つまり関数 h と g の claw を見つければ）K_1 を回復することができる．そのようなペアは BHT のアルゴリズムにより，時間 $\tilde{O}(2^{n/3})$ で探索することができる．（注意 4.1 を参照．今は Q1 モデルにおける攻撃を考えているため暗号化オラクルは古典オラクルであり関数 h の評価は古典的にしか行えない．しかし P が公開置換なので，関数 g の評価は量子重ね合わせで行える．）一旦 K_1 を回復すれば，K_2 は容易に計算できる．なおこの攻撃は BHT のアルゴリズムを用いるため大きさ $\tilde{O}(2^{n/3})$ の量子メモリを必要とする．

7.2　オンライン-オフライン中間一致攻撃

細山田と佐々木は，前節で紹介した桑門と森井の Q1 攻撃がオンライン計算とオフライン計算の中間一致攻撃とみなせることに着目し，使用可能な量子計算のリソースに関する想定（4 章参照）に応じてトレードオフが変化すること，また Even-Mansour 暗号以外にも FX 構成などに同種のオンライン-オフライン中間一致攻撃を適用できることを示した [HS18a]．以下，オンライン-オフライン中間一致攻撃およびその量子版の概要を述べる．

まず，攻撃対象の暗号技術の暗号化関数等から，次のような性質を充たす関数 $f_s, f_p : \{0,1\}^n \to \{0,1\}^n$ を構成できるという状況を考える：

1. f_s は秘密鍵に依存する関数であり，鍵の埋め込まれたオラクルへのクエリをしないと値を計算できない．

[*19] より正確には claw 探索アルゴリズム．

2. f_p は秘密鍵に依存しない関数であり，鍵の埋め込まれたオラクルへのクエリなしで，オフラインで計算できる関数である．
3. f_s と f_p の間の claw[*20]を発見すれば何らかの秘密情報（秘密鍵等）を抽出できる．

7.1 節の攻撃で言うと，f_s と f_p が h と g にそれぞれ対応する．以下簡単のため f_s と f_p はランダム関数であるとみなす．また，各 x に対する値 $f_s(x)$ の計算は，鍵の埋め込まれたオラクルへのクエリを $O(1)$ 回行えば時間 $O(1)$ で可能と仮定し，また各 x に対する値 $f_p(x)$ の計算は時間 $O(1)$ で可能とする．

古典的な設定（攻撃者が古典計算機のみを所持している設定）では，以下のようにして f_s と f_p の claw(x, y) を発見し，何らかの秘密情報を抽出することができる：

1. 鍵の埋め込まれたオラクルへのクエリ（オンラインクエリ）を行い $(x, f_s(x))$ の形のペアを異なる D 個の x について計算してリスト L に保存する．
2. L（の各要素の第二成分たち）を原像探索の標的として f_p について（古典）多重原像探索を行う．

ステップ 2 に要する計算時間（関数 f_p の評価回数）を T とすると，$T = \tilde{O}(2^n/D)$ が成り立つ．換言すれば，オンラインクエリの回数 D とオフラインの計算時間 T について $T \cdot D = \tilde{O}(2^n)$ のトレードオフが得られる．

この攻撃は鍵の埋め込まれたオラクルへのオンラインクエリを行うことによってのみ計算できる関数 f_s とオフラインで計算できる関数 f_s の値が一致しているペア (x, y) を探索する攻撃であることから，オンライン-オフライン中間一致攻撃と呼ばれる．

次に Q1 モデルにおける攻撃を考える．関数 $f_s(x)$ の値を計算をするためには古典的攻撃と同様各 x に対して $O(1)$ 回ずつ鍵の埋め込まれたオラクルへ古典クエリを行わざるを得ないが，f_p は鍵に依存しないため攻撃者が量子計算機を用いてオフラインで計算できる．特に，先述した古典攻撃のうち，ステップ 2 における f_p についての多重原像探索を量子計算機を用いて高速化することができる．

例えば 4.5 節でいうところの Case 0 の設定（QRAM が使用可能な状況）では，2.1 節で紹介した Grover のアルゴリズムを直接応用した多重原像探索アルゴリズムを用いることにより，時間 $T = O(\sqrt{2^n/D})$ のオフライン量子計算によって f_s と f_p の claw を発見できる．換言すれば，T と D について $T^2 \cdot D = O(2^n)$ のトレードオフを得られる．7.1 節の攻撃は，この例で $f_s = h, f_p = g, D = 2^{n/3}$ と設定した場合とみなすことができる．

細山田と佐々木は 4.5 節の他の Case についても，それぞれの設定で最良の多重原像探索アルゴリズム（5.3 節参照）を用いた場合に得られる T と D のトレードオフを示している．詳細は原論文 [HS18a] を参照されたい．

いくつかの共通鍵暗号技術は，（古典）オンライン-オフライン中間一致攻撃が最良の攻撃であるという前提で安全性を見積もっている．例えば Chaskey（$n = 128$）の設計者たちは，$D \leq 2^{48}$ であ

[*20] $f_s(x) = f_p(y)$ を充たす 9 ペア (x, y).

る限り実行時間が 2^{80}（$= 2^n/2^{48}$）を下回るような攻撃は存在しない，と主張している [MMH+14]．しかし Q1 モデルにおいて上述のように量子多重原像探索アルゴリズムを用いると，その主張は 4.5 節のいずれのケースにおいても成り立たないことになる．たとえば Case 2（通常の古典計算リソースに加えて量子ビットが高々 n の多項式個の小さい量子計算機を 1 つ使用可能）の場合であっても，およそ 2^{48} 回程度の古典クエリをしておけば，時間およそ 2^{56} のオフライン計算により秘密鍵を回復可能であることが示される．

■ストリーム暗号への時間・（メモリ）・トレードオフ攻撃の可能性？ [HS18a] では触れられていないが，この中間一致攻撃の設定は一部のストリーム暗号に対する古典的な時間・（メモリ）・データトレードオフ攻撃 [Bab95, Gol97, BS00, HS05] の設定に非常に近い．この攻撃はストリーム暗号の内部状態を回復するもので，内部状態のビット長を b としたときにトレードオフ $T \cdot D = O(2^b)$ が得られる．特に $D = 2^{b/2}$ とすれば攻撃時間は $T = 2^{b/2}$ となる．この攻撃が秘密鍵全探索より速くなってしまわないよう，一部のストリーム暗号は $b/2 \geq k$（k は秘密鍵の鍵長）が成り立つよう設計してある．

一方 Q1 モデル，例えば Case 0 で上記の量子アルゴリズムを適用すれば，$T = D = 2^{b/3}$ を満たすストリーム暗号への攻撃が得られる蓋然性が高い．ゆえに，上記のアルゴリズムを適用すれば鍵全数探索より速い攻撃が得られてしまうのではないかという懸念が生じる．

しかし，そもそも $b/2 \geq k$ ならば $2^{b/3} (\leq 2^{2k/3})$ は Grover のアルゴリズムによる鍵探索の計算量 $2^{k/2}$ を上回るため，上記の量子アルゴリズムが鍵全数探索より速くなることはない．（なお，ここで紹介したストリーム暗号への攻撃のトレードオフは 5.4.3 節で紹介したトレードオフに古典・量子ともに拡張されると考えられるが，同様の理由によって Grover のアルゴリズムを用いた鍵全探索より速くなることは無い．）

7.3 量子クエリ無しでの Simon のアルゴリズムの応用

6 章で述べたように，古典的に安全とされる共通鍵暗号技術であっても Simon のアルゴリズムを用いると多項式時間で破れてしまう場合があるという研究結果が近年複数報告されているが，それらの攻撃は全て鍵付きオラクルへの量子クエリを前提とする攻撃（Q2 モデルにおける攻撃）である．

Q2 モデルにおいては Simon のアルゴリズムにより各種攻撃の指数的高速化が可能となる一方で，鍵の埋め込まれたオラクルが古典オラクルである Q1 モデルにおいて Simon のアルゴリズムの恩恵を受けることができるかどうかは不明であった．しかし Bonnetain らは Asiacrypt 2019 において，Q1 モデルでの攻撃でも Simon のアルゴリズムを応用した攻撃が可能なことを示した [BHN+19]．

Bonnetain らの攻撃は，大雑把に言って

1. Q2 モデルにおいて Simon のアルゴリズム（または Simon のアルゴリズムと別の量子アルゴリズムの組み合わせ）を用いた攻撃が可能
2. 7.2 節のオンライン-オフライン中間一致攻撃が適用可能

という二つの条件が満たされるような共通鍵暗号技術に対し，高々多項式個の量子ビットを使うような小さい量子計算機のみを用いて（4.5 節での Case 2 に対応），既存の攻撃より高速な攻撃を実現

するものである．

攻撃を適用可能な共通鍵暗号技術としては Even-Mansour 暗号や FX 構成が挙げられる．例えば Bonnetain らの攻撃を Even-Mansour へ適用すると，鍵回復攻撃を多項式サイズの量子メモリおよび古典メモリのみを用いて $O(2^{n/3})$ 古典クエリ・時間 $\tilde{O}(2^{n/3})$ で実行可能である．他の攻撃との比較は表 3 を参照されたい．

表 3　Even-Mansour 暗号への Q1 モデルにおける攻撃の比較．多項式の因子およびオーダー記号は省略している．Bonnetain らの攻撃の計算量は最下段に赤字で示されている．また計算量はクエリ回数と時間が（Case 1a-1c については更にメモリも）バランスする点のみを示している．

4.5 節の Case	時間	クエリ	量子ビット (量子メモリ)	古典メモリ	出典
Case 0	$2^{n/3}$	$2^{n/3}$	$2^{n/3}$	$2^{n/3}$	[KM12]
Case 1a	$2^{n/4}$	$2^{n/4}$	$2^{n/4}$	$2^{n/4}$	[HS18a]
Case 1b	$2^{2n/7}$	$2^{2n/7}$	$2^{2n/7}$	$2^{2n/7}$	[HS18a]
Case 1c	$2^{3n/10}$	$2^{3n/10}$	$2^{3n/10}$	$2^{3n/10}$	[HS18a]
Case 2	$2^{3n/7}$	$2^{3n/7}$	$\mathrm{poly}(n)$	$2^{n/7}$	[HS18a]
Case 2	$2^{n/3}$	$2^{n/3}$	$\mathrm{poly}(n)$	$\mathrm{poly}(n)$	[BHN+19]

Bonnetain らの攻撃ではまず，指数回の古典クエリを鍵の埋め込まれたオラクルへ行い，クエリを一回行うごとにクエリの結果に応じて（多項式サイズの）量子メモリに保存されている量子状態を少しずつ変化させていく．必要な古典クエリが終ったのち，量子メモリに保存された量子状態 $|\phi\rangle$ を用いて，Simon のアルゴリズムと Grover のアルゴリズムを組み合わせたオフライン計算により秘密情報を回復する．量子メモリに保存する $|\phi\rangle$ をうまく取ることによって Simon のアルゴリズムを活用することが可能となる．攻撃の技術的詳細は原論文 [BHN+19] を参照されたい．

なお Q1 モデルにおける Even-Mansour 暗号への攻撃については，Bonnetain らの攻撃が最良で，それ以上効率的な攻撃は不可能であることが（quantum ideal permutation model で）証明されている [ABKM22].

注意 7.1. 6.3 節で Q2 モデルにおいては CBC-MAC，GCM，PMAC，GMAC，OCB，LRW 構成等が多項式時間で破られるという結果を紹介したが，これらの技術に本節の Q1 モデルにおける攻撃は適用できない．

7.4　古典的に $2k$ ビット安全なら k ビット耐量子安全か？

Grover のアルゴリズムを使うと，k ビット鍵の全数探索に必要な計算量が 2^k から $2^{k/2}$ まで落ちる．よく「量子計算機が出来た後も共通鍵暗号の安全性を今と同程度に保つには，鍵長を 2 倍以上にしないといけない」と言われるはこのためである．しかしその逆，つまり以下の主張は成り立つだろうか．

主張．鍵長が $2k$ 以上かつ，最も効率的な古典攻撃の計算量が 2^{2k} 以上ならば，量子計算機で

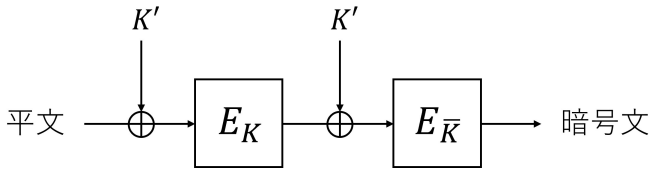


図 9　2XOR 構成．K' は K と独立した鍵で，$\bar{K}$ は K を適当な置換で変換したものである．

も破るのに時間 2^k がかかる．つまり，古典的に $2k$ ビット安全な共通鍵暗号技術は量子計算機に対して k ビット安全である．

Q2 モデルであれば，6.1 節などで挙げた多項式時間攻撃が可能になるため，上記の主張は明らかに成り立たない．更に Q1 モデル，つまりクエリが古典でもこの主張は成り立たないということが Eurocrypt 2022 において示された [BSS22].

ブロック暗号の鍵を伸ばすための構成としては，図 7 の FX 構成の他に 2XOR 構成 [GT12] というものがある（図 9）．この構成は，κ ビット鍵の n ビットブロック暗号 E から $(2n+\kappa)$ ビット鍵の n ビットブロック暗号を作るものである．E が理想的にランダムだというモデル（ideal cipher model）で，古典的に破るには時間 $O(2^{\kappa+n/2})$ が必要だということが証明されている [GT12]．例えば $\kappa = 2n$ なら，破るのに時間 $O(2^{5n/2})$ が必要である．

しかし $\kappa = 2n$ のとき，前節で紹介した Simon のアルゴリズムを用いる手法を応用すれば，Q1 モデルでも時間 $\tilde{O}(2^n)$ で破れてしまうことが示される [BSS22]．特に $k := 5n/4$ と置くと「2XOR 構成は古典的に $2k$ ビット安全だが，Q1 モデルでは k ビット安全ではない」ということになる．ゆえに先述の主張は Q1 モデルでも成り立たない．

7.5　その他の古典攻撃の高速化

Q2 モデルと同様 Q1 モデルにおいても，量子計算機を用いると様々なアルゴリズムが高速化され得るため，代表的な古典攻撃が量子計算機を用いた際どれだけ高速化できるかということは重要な研究の対象となる[*21]．Q1 モデルにおける古典攻撃の高速化に関する前節までに挙げたもの以外の主な研究結果としては，繰り返し構造を持つブロック暗号への中間一致攻撃の高速化 [Kap16] や差分解読法・線型解読法の高速化 [KLLN16b]，積分攻撃の高速化 [BNS19b]，Demiric-Selçuk 中間一致攻撃の高速化 [HS18b, BNS19b]，などが挙げられる．なおいずれの攻撃も，対応する古典攻撃でかかる時間を T としたとき，大雑把に言って $\sqrt{T}$ あるいはそれ以上の時間を要する．

また Q1 モデルにおいても，線形解読法や高速相関攻撃などの古典的な攻撃手法に現れる離散フーリエ変換をうまく量子フーリエ変換に対応づけようという研究が行われている [Sch23].

[*21] Q1 モデルにおける攻撃はそのまま Q2 モデルにおける攻撃として成立するため，ここで挙げた攻撃は全て Q2 モデルにおける攻撃とみなすこともできる．

7.6 古典的安全性証明の結果が Q1 モデルへ持ち上がる場合

古典的安全性証明の結果は Q1 モデルにほぼそのまま持ち上がって有効になることがよくある．本節では，古典的な結果がいつ Q1 モデルに持ち上がるか，注意すべき点は何か，などについて説明する．なお古典的な議論との整合性を取るため，この節では量子計算機のリソースの想定として 4.5 節の Case 0（小さい多項式サイズの量子計算機が 1 つ使えて，QRAM は必要な分だけ大きなものを使える）を仮定する．

■CTR モードの古典的安全性証明　まず n ビットブロック暗号 E_K を用いる CTR モードを例に取って説明する．簡単のため高々定数個ブロック分の長さのメッセージしか暗号化しないと仮定する．（何か定数 c があって平文は常に $M = M_1||\cdots||M_c$ の形を取るとする．各 M_i は n ビット．）また CTR モードの定義にも流儀が色々あるが，各平文 M の暗号化が以下のように処理されるものを考える．

1. $IV \in \{0,1\}^n$ をランダムに取る．
2. $Z := E_K(IV)||E_K(IV+1)||\cdots||E_K(IV+c)$ を計算する．
3. $C := M \oplus Z$ を暗号文として出力する．

この暗号化処理の結果を $\mathrm{CTR}^{E_K}(M) = C$ と書くことにする．

任意のオラクルつきアルゴリズム（攻撃者）$\mathcal{A}$ に対し，共通鍵暗号（モード）$\mathcal{E}$ の IND$-CPA 安全性[*22]に関する識別利得 $\mathsf{Adv}^{\text{IND\$-CPA}}_{\mathcal{E}}(\mathcal{A})$ は

$$\mathsf{Adv}^{\text{IND\$-CPA}}_{\mathcal{E}}(\mathcal{A}) := \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathcal{E}_K(\cdot)}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\$(\cdot)}\right]\right|$$

で定義される．ただし $\$(\cdot)$ は任意の入力に対してランダムな cn ビットの値を返すオラクルであり，秘密鍵 K は一様ランダムに取られるとする．また同様に，ブロック暗号 E の PRP 安全性に関する識別利得 $\mathsf{Adv}^{\text{PRP}}_{E}(\mathcal{A})$ は

$$\mathsf{Adv}^{\text{PRP}}_{E}(\mathcal{A}) := \left|\Pr\left[1 \leftarrow \mathcal{A}^{E_K}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{RP}}\right]\right|$$

で定義される．ここで RP は n ビット入出力のランダム置換であり，また秘密鍵 K は一様ランダムに取られるとする．このとき，CTR モードの安全性証明で示される結果は以下のようになる．

命題 7.1. $\mathcal{A}$ を CTR モードの IND$-CPA 安全性に関する任意の攻撃者として，その計算量が高々 t，クエリ回数が高々 q とする．このときブロック暗号 E の PRP 安全性に関する攻撃者 $\mathcal{B}$ であって計算量とクエリ回数がそれぞれ $O(t)$ と $O(q)$ であるようなものが存在し，

$$\mathsf{Adv}^{\text{IND\$-CPA}}_{\text{CTR}}(\mathcal{A}) \leq O(q^2/2^n) + \mathsf{Adv}^{\text{PRP}}_{E}(\mathcal{B})$$

が成り立つ．

[*22] 細かいことをいうとこの定義の流儀は本来 [Rog02] においてナンスベース暗号を念頭に導入されたものであるが，説明を簡単にするため今着目しているランダム IV のモード向けに改変して用いる．

この証明を示す上で重要になるのが以下の事実である（Game-playing proof technique [BR06] などを用いれば容易に証明できる）.

補題 7.1. RP を n ビットのランダム置換とし，CTR モードのうちブロック暗号の部分を RP に変えたものを $\mathrm{CTR}^{\mathsf{RP}}$ とおく．$\mathcal{A}$ を CTR モードの IND$-CPA 安全性に関する任意の攻撃者として，クエリ回数が高々 q とする．**なお $\mathcal{A}$ の計算時間には一切制限をつけないものとする．**このとき

$$\left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathrm{CTR}^{\mathsf{RP}}(\cdot)}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\$(\cdot)}\right]\right| \leq O(q^2/2^n)$$

が成り立つ.

この補題を用いると，命題 7.1 は以下のように証明される.

Proof. $\mathcal{B}$ を以下のようなアルゴリズムとする：$\mathcal{A}$ を走らせ，$\mathcal{A}$ がクエリしてきたときは，$\mathcal{B}$ に与えられたオラクル（E_K か RP）を用いて CTR モード（つまり CTR^{E_K} か $\mathrm{CTR}^{\mathsf{RP}}$）をシミュレートして返答する．$\mathcal{A}$ が最終的に出力したものを，$\mathcal{B}$ 自身の出力とする.

すると

$$\mathsf{Adv}_E^{\mathrm{PRP}}(\mathcal{B}) = \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathrm{CTR}^{E_K}(\cdot)}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathrm{CTR}^{\mathsf{RP}}(\cdot)}\right]\right| \tag{8}$$

が成り立つ．この等式と補題 7.1 より，

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{E}}^{\text{IND\$-CPA}}(\mathcal{A}) &= \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathcal{E}_K(\cdot)}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\$(\cdot)}\right]\right| \\
&\leq \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathrm{CTR}^{E_K}(\cdot)}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathrm{CTR}^{\mathsf{RP}}(\cdot)}\right]\right| \\
&\quad + \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathrm{CTR}^{\mathsf{RP}}(\cdot)}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\$(\cdot)}\right]\right| \\
&\leq O(q^2/2^n) + \mathsf{Adv}_E^{\mathrm{PRP}}(\mathcal{B})
\end{aligned}$$

が成り立つ. □

■Q1 モデルへの持ち上げ　上述の議論をよく吟味すると，アルゴリズムが量子になっても Q1 モデルなら証明がそのまま通用することがわかる：そのまま通用するかどうか一見非自明なのは補題 7.1 の部分であるが，補題では $\mathcal{A}$ の計算時間に制限をつけていない．有名な事実として，任意の量子アルゴリズムの挙動は，時間に制限をつけず効率を度外視すれば古典アルゴリズムでシミュレートできる．ゆえに補題 7.1 は量子アルゴリズムに対しても（Q1 モデルで，クエリが古典である限り）適用できてしまう．よって命題 7.1 も Q1 モデルでそのまま成り立つことがわかる.

注意 7.2. Q2 モデルではこのようなことは成り立たない．補題 7.1 は古典クエリが前提になっており，Q2 モデルで量子重ね合わせクエリが発生すると有効ではなくなってしまうからである．ゆえに，6.2 節で紹介した 3 段 Feistel 暗号への量子選択平文攻撃のように，古典的に安全性証明があるにも関わらず Q2 モデルで破れてしまうという事態が発生する.

命題そのものは Q1 モデルでも有効になるが，実際の安全性が保証される範囲については注意が必要である．命題 7.1 に現れるブロック暗号 E_K の識別利得 $\mathsf{Adv}_E^{\mathrm{PRP}}(\mathcal{B})$ は，E_K が十分安全かつア

ルゴリズムが全て古典なら，計算時間が 2^k に達するまで非常に小さいままである．しかしアルゴリズムが量子になると，$\mathsf{Adv}_E^{\mathrm{PRP}}(\mathcal{B})$ は時間 $2^{k/2}$ で（Grover の鍵全数探索により）ほぼ 1 になりうる．ゆえに，Q1 モデルで CTR モードの安全性が保証される範囲を具体的に述べると「クエリ数などは古典的に安全性が保証される範囲に収まり（つまり $q \ll 2^{n/2}$），かつ時間 $\leq 2^{k/2}$」となる．

■議論の一般化　ここまでの議論は CTR モードについてのものであったが，重要なのは

> （★）安全性証明がランダムオラクルモデル，ideal permutation model，ideal cipher model などプリミティブが理想化されたモデル下で与えられるのでなく，反証可能な標準的仮定（CTR モードなら E_K が PRP という仮定）のみに依存している

という点である．CTR モードで無くても，（★）が満たされている限り，古典的安全性証明の結果が Q1 モデルでもそのまま成り立つ．（ただし安全性が保証される範囲を具体的に述べようとすると，先述のように Grover のアルゴリズムで鍵全数探索等の影響を考慮する必要は発生する．）

一方，安全性が ideal permutation model などプリミティブが理想化されたモデル下で与えられている場合は，Q1 モデルであっても証明は有効にならない．これは理想化されたプリミティブへの量子クエリが発生してしまい，古典的な証明が通用しなくなるためである（例えば Even-Mansour の置換 P）．7.2 節（と 7.3 節）で触れた，Chaskey の古典的な安全性の見積もりが Q1 モデルで成り立たなくなるという結果は，本質的にこれが原因である（Chaskey の古典的安全性証明は ideal permutation model で与えられている）．

8　ハッシュ関数への（汎用でない）攻撃

本章では，ハッシュ関数への攻撃であって，（汎用攻撃とは違い）特定の圧縮関数や置換などの内部構造を利用するようなものについて紹介する．3 章で触れたように，ハッシュ関数は秘密鍵を用いないため鍵の埋め込まれたオラクルというものが存在しない．攻撃者はハッシュ関数あるいはその一部を量子計算機上に自由に実装することができる．

8.1　衝突攻撃

衝突攻撃は，量子計算機があれば攻撃可能段数が伸びることが多々あることがわかっている．これは細山田・佐々木の論文 [HS20] で初めて指摘されたもので，SHA-2 を初めとした様々なハッシュ関数（あるいはハッシュ関数の部品として使う圧縮関数）において衝突攻撃の攻撃可能段数が伸びることが判明している [HS21, DSS+20, CKS21, LH24, GLST22, FLN+20]．以下，なぜ攻撃可能段数が伸びるかということの概要を説明する．

8.1.1　古典的に「意味のある」衝突攻撃

攻撃のための古典アルゴリズムを見つけたとき，それが（少なくとも学術的に）意味があるとみなされるための条件は，対応する汎用攻撃よりも効率的なことである．出力長が n ビットのハッシュ関数に対する衝突攻撃であれば，対応する汎用攻撃は計算量 $2^{n/2}$ の誕生日攻撃（4.1 節）である．例えば出力長が 256 ビットである SHA-256 に衝突攻撃を思いついたとして，その計算量が 2^{120} などであれば，その攻撃は少なくとも学術的には意味があると見做される．

ここで例に挙げた 2^{120} というのは現実的には実行不可能な計算量である．しかし多くの攻撃の研究は，まず「少なくとも学術的には意味がある」と見做せるような攻撃が見つかって，その計算量がどんどん改善されていくという順序を辿る．例えば SHA-1 に対する衝突攻撃も，CRYPTO 2006 報告でされた最初の攻撃の計算量は 2^{69} 程度であったが [WYY05]，その後削減が進み CRYPTO 2017 で実際の衝突が報告されるに至った [SBK+17]．学術的に意味があると見做せる攻撃が見つかれば，アルゴリズムの移行を考え始めたほうが安全である．

■攻撃可能段数　ハッシュ関数やブロック暗号などの共通鍵プリミティブは同じような処理を何度も繰り返すような構造になっている．たとえば AES-128 が平文から暗号文を計算する際は，特定の変換をしてから秘密鍵（から計算された値）を足すという操作を 10 段繰り返す．SHA-256 の圧縮関数なら 64 段である．

ハッシュ関数にしろブロック暗号にしろ，急に破れるということはない．攻撃を研究する研究者はまず，その段数を削減したものを攻撃することを試みる．たとえば SHA-256 であれば「元の 64 段のものは破れないから段数を 20 段にまで削減したものを考えて，この 20 段の関数に対して汎用攻撃よりも効率的な衝突攻撃を試みよう」ということになる．実際に効率的な衝突攻撃が見つかれば「20 段まで破れた」と言い，次は 21 段を破る攻撃を探す，という具合に研究は進んでいく．元の段数のうち何段まで破れているかということも安全性指標の一つとみなせる．例えば元々 30 段ある

ハッシュ関数が 29 段まで破れてしまえば，元の（30 段の）関数もそろそろ危ないのではないかという気になってくる．

なお攻撃可能段数はセキュリティパラメータ（ハッシュ関数であれば出力長 n）と独立した指標であることに注意されたい．例えば，10 ビット出力のハッシュ関数があって，元々 40 段の段数のうち 5 段までしか破られていないとする．するとこのハッシュ関数は攻撃可能段数の面からは安心できるが，出力長がそもそも短すぎるので安全ではない．逆に 256 ビット出力・80 段のハッシュ関数があって，79 段まで衝突耐性が破れてしまっているなら，セキュリティパラメータは十分長いが攻撃可能段数という面では若干不安が出てくる．

8.1.2 意味のある量子衝突攻撃とは？

量子計算を用いた攻撃の話に戻る．量子計算機を用いた衝突攻撃でまず初めに見つかったのは BHT のアルゴリズム（4.2 節）である．しかしこのアルゴリズムは Shor のアルゴリズムと違って指数的な高速化が得られるわけではない．ゆえにハッシュ関数の出力長を少し長くしておけば特に何の影響も出ないだろう，というのが支配的な見方であった．

しかし，ハッシュ関数の安全性を考える上で重要になのは BHT などの汎用攻撃だけではない．古典的な安全性を評価する上で重要なことの一つは，先述のように，汎用攻撃よりも効率的な攻撃があるか，そして何段まで破れるかということであった．この考え方を量子計算を用いた衝突攻撃にも持ち込むとどうなるかということを考える．

まずは 4.5 節の Case 0，すなわち小さいサイズの計算用量子プロセッサと指数的に大きな QRAM が利用可能であるという設定を考える．このとき（現在見つかっている中で）最も効率的な汎用衝突攻撃は BHT のアルゴリズムで，計算量は $2^{n/3}$ である．古典的な「衝突攻撃が意味を持つかどうかは汎用衝突攻撃（計算量 $2^{n/2}$）より効率的かどうかで決まる」という考え方と整合性を持たせようとすると，「Case 0 において衝突攻撃が意味を持つかどうかは BHT より効率的かどうかで決めるべきである」と言える．

ここで着目すべきは，意味があるかどうかの閾値が古典（誕生日攻撃の $2^{n/2}$）と量子（BHT の $2^{n/3}$）でそこまで大きく変わらないということである．秘密鍵の全数探索は Grover のアルゴリズムによって古典的計算量の平方根まで落ちるが，BHT を使っても計算量の下げ幅は平方根まで落ちない．

一方，特定のブロック暗号やハッシュ関数の内部構造を利用する攻撃は，量子計算機を使うと元の平方根まで落ちることがよくある．これはなぜかというと，差分解読法を初めとする古典攻撃の重要な部分の多くが全数探索と見做せて，そこに Grover のアルゴリズムを適用できるからである．

一旦状況を整理すると以下のとおりである：量子計算機がある世界では，衝突攻撃の意味があるかどうかの閾値が古典と比べて大きく変わらない．一方，特定のハッシュ関数の内部構造を利用した攻撃は，量子アルゴリズムを用いて比較的大きな高速化が得られる．

これは即ち，量子計算機のある世界では，特定のハッシュ関数に対する攻撃の威力が相対的に高まるということである．古典的な計算量が閾値を上回っていて意味がないとされるような攻撃も，Grover のアルゴリズムなどを使って高速化すれば，量子計算機がある世界では閾値を下回って意味

表 4　SHA-2 および SHA-3 に対する衝突攻撃のうち，古典攻撃よりも攻撃可能段数が大きいもの．いずれも Case 1a における攻撃で，計算機サイズ S は古典メモリも含めたものである．

攻撃対象	出力長	攻撃段数 / 全段数	時間	計算機サイズ（汎用攻撃より速くなる範囲）	出典
任意の関数	n	-	$2^{n/2}/S$	S	[Ber09]（汎用攻撃）
SHA-256	256	38 / 64	$2^{122}/\sqrt{S}$	$S(\leq 2^{12})$	[HS21]
SHA-512	512	39 / 80	$2^{252.7}/\sqrt{S}$	$S(\leq 2^{6.6})$	[HS21]
SHA3-224	224	6 / 24	$2^{97.75}/\sqrt{S}$	$S(\leq 2^{28.5})$	[GLST22]
SHA3-256	256	6 / 24	$2^{104.25}/\sqrt{S}$	$S(\leq 2^{47.5})$	[GLST22]

があると判定されるようになる可能性がある．

今までの議論は Case 0 におけるものであったが，他の設定でも同様である．例えば Case 2 における汎用衝突攻撃は CNS のアルゴリズム（4.4 節）で，計算量は $2^{2n/5}$ である．古典的な誕生日攻撃の計算量 $2^{n/2}$ からの下げ幅は BHT よりもさらに小さく，Case 2 においても攻撃が容易になると考えられる．

Case 1a に至っては，下げ幅がほぼ無いに等しくなる．使える計算機のサイズ（量子ビットの数のみでなく，古典的計算機のプロセッサの数やメモリの大きさをも全て含んだサイズ）を S とすると，Case 1a での汎用攻撃は parallel rho 法を用いたものであり [Ber09]，計算量は $2^{n/2}/S$ である．S が小さい場合は誕生日攻撃の計算量 $2^{n/2}$ とほぼ変わらない．よって，衝突攻撃が一層容易になる．

■具体例：SHA-2 と SHA-3　SHA-256（段数は全部で 64 段）と SHA-512（80 段）への衝突攻撃については，古典で破れているのは本原稿執筆時点で両方とも 31 段までである [LLW24]．一方量子計算機がある場合，Case 1a ではそれぞれ 38 段・39 段まで破れることが示されている [HS21]．また SHA-3 について，SHA3-224 と SHA3-256 は古典で 24 段中 5 段までしか破れていないが [GLL+20, SLG17]，量子計算の Case 1a だと 6 段まで破れることが示されている [GLST22]．計算量を表 4 にまとめる．

8.2　原像攻撃

ハッシュ関数について衝突攻撃と共に重要な攻撃は原像攻撃である．原像攻撃については，汎用攻撃の計算量は古典の 2^n に対し量子で $2^{n/2}$ であり（5.1 節），平方根程度の高速化が得られてしまう．よって衝突攻撃とは違い，量子計算機が使える設定になったからといって，特定のハッシュ関数に対する原像攻撃の攻撃可能段数が容易に伸びるとは言えない．実際，量子計算機を用いて特定のハッシュ関数に原像攻撃を行う研究は [SS22] 等で行われているが，攻撃可能段数は今のところ古典を上回るものではない．

9　考察とまとめ

量子コンピュータが共通鍵暗号技術の安全性に及ぼす影響の調査および評価を報告した．既存文献について調査を行い，量子コンピュータを用いた攻撃のモデル，特にハッシュ関数以外の（秘密鍵を用いる）共通鍵暗号技術への攻撃のモデルには Q1 モデルと Q2 モデルの二種類のモデルが存在することを確認した．Q1 モデルにおいては鍵の埋め込まれたオラクルは古典的な攻撃モデルと同じ古典オラクルだが，Q2 モデルにおいては鍵の埋め込まれたオラクルが量子オラクルとなる．それぞれのモデルにおけるハッシュ関数以外の暗号技術への攻撃と（モデルの区別がない）ハッシュ関数への攻撃について、汎用攻撃（暗号技術の内部構造を利用せず任意の方式に適用できる攻撃）と特定の方式の内部構造を利用する攻撃に分け、それぞれ既存研究を調査した．また主にハッシュ関数への汎用衝突探索攻撃について攻撃コストの評価に関する既存の議論の調査を行った．

以下，Q1 モデルの攻撃，Q2 モデルの攻撃，ハッシュ関数への攻撃に分けてそれぞれ簡潔にまとめを述べる．また種々の重要な方式の安全性への影響をまとめる．より具体的には，CRYPTREC の電子政府推奨暗号リストの方式，および CRYPTREC 暗号技術ガイドライン（軽量暗号）2023 年度版 [CRY23] で触れらているものの中で特に最近 NIST 標準に選ばれた Ascon に焦点を当てる．なお電子政府推奨暗号リストは本稿執筆時点で最新の令和 6 年 5 月 16 日版（CRYPTREC LS-0001-2022R1 [デ 24]）を参照する．また Ascon のアルゴリズムについては，NIST SP 800-232 の initial public draft [TMC+24] に定められているもの（Ascon-AEAD128, Ascon-Hash256, Ascon-(C)XOF128），および NIST 投稿版 [DEMS21] に含まれ耐量子性を考慮している Ascon-80pq を取り上げるものとする．

■Q2 モデルにおける攻撃　Q2 モデルにおいては，古典的に安全とされているいくつかの共通鍵暗号技術（CBC-MAC や GCM など）に多項式時間の攻撃が存在するが，このモデルでの攻撃を実行するためには攻撃対象の暗号技術が量子回路上に実装されている必要がある（鍵長が長いときは Q2 攻撃を Q1 攻撃に変換できることがあるが，便宜上これは Q1 攻撃とみなすことにする）．

ある関数を計算するための古典計算機向けのプログラムコードがあった場合，その関数を量子回路上に実装することが可能になる．ゆえに，Q2 モデルにおいて多項式時間の攻撃が可能な暗号技術については，例え難読化処理等を施してもその関数（例えば CBC-MAC でメッセージからタグを計算する関数）を実装して秘密鍵を埋め込んだコードを量子コンピュータを持った攻撃者に手渡すべきではない．しかし，攻撃対象となる暗号技術が量子回路上に実装されているような（あるいは量子回路上に移植可能となるような）非常に特殊な状況でない限り，既存の共通鍵暗号技術に Q2 モデルの攻撃の影響が及ぶことは現状では無いと考えられる．

特に，CRYPTREC の電子政府推奨暗号リストにある共通鍵暗号技術や Ascon の安全性を評価する上で Q2 モデルの攻撃を考慮する必要性は低い．

■Q1 モデルにおける攻撃　Q1 モデルでの攻撃については，Q2 モデルと異なり，古典的に安全とされている共通鍵暗号技術に多項式時間の攻撃は存在しないことを確認した．

しかし近年の攻撃研究の進展により，暗号技術の内部構造に依存した攻撃が Q1 モデルにおいて

も多数報告されている．ブロック暗号の攻撃可能段数が古典より伸びるという例は今のところ見つかっていないが，古典的に $2k$ ビット以上の安全性があっても Q1 モデルでの安全性が k ビットを下回る例が示されている（7.4 節）．Q1 モデルであっても，鍵を 2 倍にしたら古典と同じ安全性が保障されるとは限らないため，暗号技術ごとに確認が必要である．また Even-Mansour 暗号および類似の構造を持つ暗号技術（Chaskey など）については，使用される置換が n ビット置換であるとき，n の多項式個程度の量子ビットを計算に使用できる量子コンピュータがあれば時間 $\tilde{O}(2^{n/3})$ で鍵回復が可能になるため（7.2 節および 7.3 節），量子コンピュータに対して k ビット安全性を達成したい場合は $3k$ ビット以上の大きさの置換を使用する必要がある．

主にモードについて，ideal permutation model などプリミティブを理想化したモデルでなく反証可能な標準的仮定（ブロック暗号の PRP 安全性など）に依拠する古典的安全性証明は Q1 モデルにそのまま持ち上がる．つまり，古典的な安全性証明がついていれば，（ブロック暗号などのプリミティブに対する攻撃の影響を考慮する必要はあるが）データ量やクエリ回数などについて安全性が保障される範囲は古典的設定と Q1 モデルで変わらない．Ideal permutation model や ideal cipher model で証明された安全性は Q1 モデルに持ち上がるとは限らないので，方式ごとに安全性を再精査する必要がある．

幸い Q1 モデルにおいて，CRYPTREC の電子政府推奨暗号リストにある（ハッシュ関数以外の）共通鍵暗号技術や Ascon-AEAD128 と Ascon-80pq の安全性に量子コンピュータが与える影響は“Grover のアルゴリズムを用いると k ビット鍵の全数探索が時間 $O(2^{k/2})$ で実行できるため，長期的に保護したいデータには鍵長が 192 ビットや 256 ビットの暗号技術を使用した方が賢明である”という以上のものは現状では無いと考えられる．しかし，Even-Mansour 暗号への Q1 モデルにおける攻撃のように安全性へ現実的な影響を直接及ぼしうる攻撃が今後発見される可能性もあるため，研究の動向には今後も注意を払っておく必要がある．

電子政府推奨暗号リストにあるハッシュ関数以外の方式と Ascon-AEAD および Ascon-80pq について，Q1 モデルで安全性が現状期待できる範囲を表 5 に示す．

■ハッシュ関数への攻撃　多くのハッシュ関数について，量子計算機が使えるようになれば衝突攻撃の攻撃可能段数が伸びることが示されている（8.1 節）．攻撃可能段数が伸びるものには，CRYPTREC の電子政府推奨暗号リストにある SHA-256 と SHA-512 および SHA3-256 が含まれるが，破れているのはそれぞれ 64 段中 38 段，80 段中 39 段，および 24 段中 6 段で，まだ余裕がある．段数削減なしで衝突耐性が破れる心配は今の所無いと考えられるが，今後も研究の進展を注視する必要がある．特に，CRYPTREC の電子政府推奨暗号リストにあるハッシュ関数や Ascon（Ascon-Hash256, Ascon-(C)XOF128）の安全性に量子計算機が及ぼす影響については，汎用的な攻撃の影響のみ考慮すれば今のところは充分である．

汎用的な攻撃のうち主に考慮に入れるべきものは，（量子計算の有無に関わらず）原像探索と衝突探索である．原像探索については，Grover のアルゴリズムを用いれば n ビット出力ハッシュ関数の原像を発見するのに要する時間が古典の $O(2^n)$ から $O(2^{n/2})$ にまで高速化される（5.1 節）．また衝突探索については，BHT のアルゴリズムを用いれば衝突を発見するのに要する時間が古典の $O(2^{n/2})$ から量子の $O(2^{n/3})$ まで高速化される（4.2 節）．なおスポンジ構造，特に XOF について

表 5　電子政府推奨暗号リストにあるハッシュ関数以外の共通鍵暗号技術と Ascon-AEAD および Ascon-80pq について，Q1 モデルで安全性が期待できる範囲．古典計算機のみが使える典型的な安全性評価と整合性を取るため，単一鍵の安全性に焦点を当て，量子計算機のリソースの想定としては 4.5 節の Case 0（小さい多項式サイズの量子計算機が 1 つ使えて，QRAM は必要な分だけ大きなものを使える）を仮定する．この表にある暗号技術については，Q1 モデルにおいて（古典的に考慮すべき事項から追加して）現状考慮すべきと思われる事柄は Grover のアルゴリズムによる鍵回復のみである．

技術分類	暗号技術名	鍵長（ビット）	Q1 モデルで安全性が期待できる範囲
ブロック暗号	AES, Camellia	128	時間 $\leq 2^{64}$
		192	時間 $\leq 2^{96}$
		256	時間 $\leq 2^{128}$
ストリーム暗号	KCipher-2	128	時間 $\leq 2^{64}$
秘匿モード	CBC, CFB, CTR, OFB, XTS	k	時間 $\leq 2^{k/2}$ かつ古典的に安全性が保障される範囲
認証付き秘匿モード	CCM, GCM	k	時間 $\leq 2^{k/2}$ かつ古典的に安全性が保障される範囲
メッセージ認証コード	CMAC, HMAC	k	時間 $\leq 2^{k/2}$ かつ古典的に安全性が保障される範囲
認証暗号	ChaCha20-Poly1305	256	時間 $\leq 2^{128}$ かつ古典的に安全性が保障される範囲
認証暗号	Ascon-AEAD128	128	時間 $\leq 2^{64}$ かつ古典的に安全性が保障される範囲
認証暗号	Ascon-80pq	160	時間 $\leq 2^{80}$ かつ古典的に安全性が保障される範囲

は，内部状態のキャパシティ部分で衝突を見つけられれば出力の衝突を見つけられる．よって，出力長とキャパシティがそれぞれ ℓ ビットおよび c ビットのとき，$O\left(\min(2^{\ell/3}, 2^{c/3})\right)$ の計算時間と量子メモリで衝突を見つけられる．

BHT のアルゴリズムは非常に大きな量子メモリを必要とし，古典衝突探索アルゴリズムや他の単純な衝突探索アルゴリズムと比べて真に効率的か否かについては様々な議論がある（4 章）．しかし，SHA-256 や SHA-512，SHA3-256 を含むハッシュ関数の攻撃可能段数が古典より伸びることがここ数年で判明していることも考慮すると，重要な用途に供するハッシュ関数の出力長（スポンジ構造の場合は出力長に加えキャパシティ長）は BHT のアルゴリズムの計算量を基準にして 384 ビッ

表 6　電子政府推奨暗号リストのハッシュ関数と Ascon-Hash256 および Ascon-(C)XOF128 に対して BHT のアルゴリズム（4.2 節）を適用する際に必要な計算時間と量子メモリの概算値．計算時間の値はそのまま，安全性が期待できる時間の範囲の上限に対応する．なお出力長とキャパシティの単位はいずれもビットである．スポンジ構造のハッシュ関数，特に XOF（SHAKE128，SHAKE256，Ascon-(C)XOF128）については，内部状態のキャパシティ部分で衝突を見つける攻撃も考慮に入っていることに注意されたい．

暗号技術名	キャパシティ	出力長	計算時間	量子メモリ
SHA-256	-	256	$2^{85.3}$	$2^{85.3}$
SHA-512/256	-			
SHA3-256	512			
SHA-384	-	384	2^{128}	2^{128}
SHA3-384	768			
SHA-512	-	512	$2^{170.7}$	$2^{170.7}$
SHA3-512	1024			
SHAKE128	256	ℓ (≥ 256)	$\min(2^{85.3}, 2^{\ell/3})$	$\min(2^{85.3}, 2^{\ell/3})$
SHAKE256	512	ℓ (≥ 256)	$\min(2^{170.7}, 2^{\ell/3})$	$\min(2^{170.7}, 2^{\ell/3})$
Ascon-Hash256	256	256	$2^{85.3}$	$2^{85.3}$
Ascon-(C)XOF128	256	ℓ (> 0)	$\min(2^{85.3}, 2^{\ell/3})$	$\min(2^{85.3}, 2^{\ell/3})$

トや 512 ビットのものを用いた方が無難であると考えられる．

電子政府推奨暗号リストと Ascon-Hash256 および Ascon-(C)XOF128 について，BHT のアルゴリズムに必要な計算時間および量子メモリの概算値を表 6 にまとめる．

参考文献

[ABKM22] Gorjan Alagic, Chen Bai, Jonathan Katz, and Christian Majenz. Post-quantum security of the Even-Mansour cipher. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 458–487. Springer, 2022.

[AIK+00] Kazumaro Aoki, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, and Toshio Tokita. Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis. In Douglas R. Stinson and Stafford E. Tavares, editors, *SAC 2000, Proceedings*, volume 2012 of *Lecture Notes in Computer Science*, pages 39–56. Springer, 2000.

[AMG+16] Matthew Amy, Olivia Di Matteo, Vlad Gheorghiu, Michele Mosca, Alex Parent, and John M. Schanck. Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3. In Roberto Avanzi and Howard M. Heys, editors, *Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers*, volume 10532 of *Lecture Notes in Computer Science*, pages 317–337. Springer, 2016.

[AR17] Gorjan Alagic and Alexander Russell. Quantum-secure symmetric-key cryptography based on hidden shifts. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 65–93, 2017.

[ASAM18] Mishal Almazrooie, Azman Samsudin, Rosni Abdullah, and Kussay N. Mutter. Quantum reversible circuit of AES-128. *Quantum Information Processing*, 17(5):112, 2018.

[Bab95] S.H. Babbage. Improved "exhaustive search" attacks on stream ciphers. In *European Convention on Security and Detection, 1995.*, pages 161–166, 1995.

[BB17] Gustavo Banegas and Daniel J. Bernstein. Low-communication parallel quantum multi-target preimage search. In Carlisle Adams and Jan Camenisch, editors, *Selected Areas in Cryptography - SAC 2017 - 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers*, volume 10719 of *Lecture Notes in Computer Science*, pages 325–335. Springer, 2017.

[BBHT98] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505,

1998.

[BBS06] Elad Barkan, Eli Biham, and Adi Shamir. Rigorous bounds on cryptanalytic time/memory tradeoffs. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 1–21. Springer, 2006.

[BCSS23] Xavier Bonnetain, André Chailloux, André Schrottenloher, and Yixin Shen. Finding many collisions via reusable quantum walks - application to lattice sieving. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 221–251. Springer, 2023.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011.

[Ber05] Daniel J. Bernstein. The Poly1305-AES message-authentication code. In Henri Gilbert and Helena Handschuh, editors, *Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers*, volume 3557 of *Lecture Notes in Computer Science*, pages 32–49. Springer, 2005.

[Ber09] Daniel J. Bernstein. Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete? In *SHARCS*, 2009.

[BFH22] Barbara Jiabao Benedikt, Marc Fischlin, and Moritz Huppert. Nostradamus goes quantum. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part III*, volume 13793 of *Lecture Notes in Computer Science*, pages 583–613. Springer, 2022.

[BHN+19] Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. Quantum attacks without superposition queries: The offline Simon's algorithm. In *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*, pages 552–583, 2019.

[BHT97] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. *SIGACT News*, 28(2):14–19, 1997.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.

[BMS05] Alex Biryukov, Sourav Mukhopadhyay, and Palash Sarkar. Improved time-memory trade-offs with multiple data. In Bart Preneel and Stafford E. Tavares, editors, *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*, volume 3897 of *Lecture Notes in Computer Science*, pages 110–127. Springer, 2005.

[BN18] Xavier Bonnetain and María Naya-Plasencia. Hidden shift quantum cryptanalysis and implications. In *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, pages 560–592, 2018.

[BNS19a] Xavier Bonnetain, María Naya-Plasencia, and André Schrottenloher. On quantum slide attacks. In Kenneth G. Paterson and Douglas Stebila, editors, *Selected Areas in Cryptography - SAC 2019 - 26th International Conference, Waterloo, ON, Canada, August 12-16, 2019, Revised Selected Papers*, volume 11959 of *Lecture Notes in Computer Science*, pages 492–519. Springer, 2019.

[BNS19b] Xavier Bonnetain, María Naya-Plasencia, and André Schrottenloher. Quantum security analysis of AES. *IACR Trans. Symmetric Cryptol.*, 2019(2):55–93, 2019.

[BR00] John Black and Phillip Rogaway. CBC MACs for arbitrary-length messages: The three-key constructions. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, volume 1880 of *Lecture Notes in Computer Science*, pages 197–215. Springer, 2000.

[BR02] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 384–397. Springer, 2002.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.

[BS92] Eli Biham and Adi Shamir. Differential cryptanalysis of the full 16-round DES. In Ernest F. Brickell, editor, *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*, pages 487–496. Springer, 1992.

[BS00] Alex Biryukov and Adi Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2000.

[BSS+13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK families of lightweight block ciphers. IACR Cryptology ePrint Archive 2013/404, 2013.

[BSS22] Xavier Bonnetain, André Schrottenloher, and Ferdinand Sibleyras. Beyond quadratic speedups in quantum attacks on symmetric schemes. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 315–344. Springer, 2022.

[BSU12] Simon R. Blackburn, Douglas R. Stinson, and Jalaj Upadhyay. On the complexity of the herding attack and some related attacks on hash functions. *Des. Codes Cryptogr.*, 64(1-2):171–193, 2012.

[BW99] Alex Biryukov and David A. Wagner. Slide attacks. In Lars R. Knudsen, editor, *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, volume 1636 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 1999.

[BW00] Alex Biryukov and David A. Wagner. Advanced slide attacks. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May*

14-18, 2000, Proceeding, volume 1807 of *Lecture Notes in Computer Science*, pages 589–606. Springer, 2000.

[CCP24] Maya Chartouny, Benoit Cogliati, and Jacquess Patarin. Classical and quantum generic attacks on 6-round Feistel schemes. IACR Cryptology ePrint Archive 2024/458, 2024.

[CE05] Andrew M. Childs and Jason M. Eisenberg. Quantum algorithms for subset finding. *Quantum Info. Comput.*, 5(7):593–604, nov 2005.

[CHLS20] Carlos Cid, Akinori Hosoyamada, Yunwen Liu, and Siang Meng Sim. Quantum cryptanalysis on contracting Feistel structures and observation on related-key settings. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology - INDOCRYPT 2020 - 21st International Conference on Cryptology in India, Bangalore, India, December 13-16, 2020, Proceedings*, volume 12578 of *Lecture Notes in Computer Science*, pages 373–394. Springer, 2020.

[CKS21] Amit Kumar Chauhan, Abhishek Kumar, and Somitra Kumar Sanadhya. Quantum free-start collision attacks on double block length hashing with round-reduced AES-256. *IACR Trans. Symmetric Cryptol.*, 2021(1):316–336, 2021.

[CLF+24] Jingwen Chen, Qun Liu, Yanhong Fan, Lixuan Wu, Boyun Li, and Meiqin Wang. New SAT-based model for quantum circuit decision problem: Searching for low-cost quantum implementation. *IACR Commun. Cryptol.*, 1(1):31, 2024.

[CLS22] Federico Canale, Gregor Leander, and Lukas Stennes. Simon's algorithm and symmetric crypto: Generalizations and automatized applications. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 779–808. Springer, 2022.

[CNS17] André Chailloux, María Naya-Plasencia, and André Schrottenloher. An efficient quantum collision search algorithm and implications on symmetric cryptography. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 211–240. Springer, 2017.

[CPT23] Maya Chartouny, Jacques Patarin, and Ambre Toulemonde. Quantum cryptanalysis of 5 rounds Feistel schemes and Benes schemes. In Said El Hajji, Sihem Mesnager, and El Mamoun Souidi, editors, *Codes, Cryptology and Information Security - 4th International Conference, C2SI 2023, Rabat, Morocco, May 29-31, 2023, Proceedings*, volume 13874 of *Lecture Notes in Computer Science*, pages 196–203. Springer,

2023.

[CRY23] CRYPTREC 暗号技術評価委員会. Cryptrec 暗号技術ガイドライン（軽量暗号）2023年度版, 2023. 文書番号 CRYPTREC GL-2006-2023.

[DEMS21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2 Submission to NIST, 2021.

[DKRS24] Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir. Quantum time/memory/data tradeoff attacks. *Des. Codes Cryptogr.*, 92(1):159–177, 2024.

[DLW19] Xiaoyang Dong, Zheng Li, and Xiaoyun Wang. Quantum cryptanalysis on some generalized Feistel schemes. *SCIENCE CHINA Information Sciences*, 62(2):22501:1–22501:12, 2019.

[DSS+20] Xiaoyang Dong, Siwei Sun, Danping Shi, Fei Gao, Xiaoyun Wang, and Lei Hu. Quantum collision attacks on AES-like hashing with low quantum random access memories. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 727–757. Springer, 2020.

[DW18] Xiaoyang Dong and Xiaoyun Wang. Quantum key-recovery attack on Feistel structures. *SCIENCE CHINA Information Sciences*, 61(10):102501:1–102501:7, 2018.

[EM91] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. In Hideki Imai, Ronald L. Rivest, and Tsutomu Matsumoto, editors, *Advances in Cryptology - ASIACRYPT '91, International Conference on the Theory and Applications of Cryptology, Fujiyoshida, Japan, November 11-14, 1991, Proceedings*, volume 739 of *Lecture Notes in Computer Science*, pages 210–224. Springer, 1991.

[FLN+20] Antonio Flórez-Gutiérrez, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, André Schrottenloher, and Ferdinand Sibleyras. New results on Gimli: Full-permutation distinguishers and improved collisions. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 33–63. Springer, 2020.

[GLL+20] Jian Guo, Guohong Liao, Guozhen Liu, Meicheng Liu, Kexin Qiao, and Ling Song. Practical collision attacks against round-reduced SHA-3. *J. Cryptol.*, 33(1):228–270, 2020.

[GLM08] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, Apr 2008.

[GLRS16] Markus Grassl, Brandon Langenberg, Martin Roetteler, and Rainer Steinwandt. Applying Grover's algorithm to AES: quantum resource estimates. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings*, volume 9606 of *Lecture Notes in Computer Science*, pages 29–43. Springer, 2016.

[GLST22] Jian Guo, Guozhen Liu, Ling Song, and Yi Tu. Exploring SAT for cryptanalysis: (Quantum) collision attacks against 6-round SHA-3. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part III*, volume 13793 of *Lecture Notes in Computer Science*, pages 645–674. Springer, 2022.

[GNS18] Lorenzo Grassi, María Naya-Plasencia, and André Schrottenloher. Quantum algorithms for the k -xor problem. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 527–559. Springer, 2018.

[Gol97] Jovan Dj. Golic. Cryptanalysis of alleged A5 stream cipher. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 239–255. Springer, 1997.

[GR04] Lov K. Grover and Terry Rudolph. How significant are the known collision and element distinctness quantum algorithms? *Quantum Information & Computation*, 4(3):201–206, 2004.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996.

[GT12] Peter Gazi and Stefano Tessaro. Efficient and optimally secure key-length extension for block ciphers via randomized cascading. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 63–80. Springer, 2012.

[HA17] Akinori Hosoyamada and Kazumaro Aoki. On quantum related-key attacks on iterated Even-Mansour ciphers. In Satoshi Obana and Koji Chida, editors, *Advances*

in Information and Computer Security - 12th International Workshop on Security, IWSEC 2017, Hiroshima, Japan, August 30 - September 1, 2017, Proceedings, volume 10418 of *Lecture Notes in Computer Science*, pages 3–18. Springer, 2017.

[Hel80] Martin Hellman. A cryptanalytic time-memory trade-off. *IEEE transactions on Information Theory*, 26(4):401–406, 1980.

[HKK20] Samir Hodzic, Lars Ramkilde Knudsen, and Andreas Brasen Kidmose. On quantum distinguishers for type-3 generalized Feistel network based on separability. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings*, volume 12100 of *Lecture Notes in Computer Science*, pages 461–480. Springer, 2020.

[Hos23] Akinori Hosoyamada. Quantum speed-up for multidimensional (zero correlation) linear distinguishers. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part III*, volume 14440 of *Lecture Notes in Computer Science*, pages 311–345. Springer, 2023.

[Hos24] Akinori Hosoyamada. Quantum algorithms for fast correlation attacks on lfsr-based stream ciphers. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part VIII*, volume 15491 of *Lecture Notes in Computer Science*, pages 396–430. Springer, 2024.

[HS05] Jin Hong and Palash Sarkar. Rediscovery of time memory tradeoffs. *IACR Cryptology ePrint Archive*, page 90, 2005.

[HS18a] Akinori Hosoyamada and Yu Sasaki. Cryptanalysis against symmetric-key schemes with online classical queries and offline quantum computations. In Nigel P. Smart, editor, *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, volume 10808 of *Lecture Notes in Computer Science*, pages 198–218. Springer, 2018.

[HS18b] Akinori Hosoyamada and Yu Sasaki. Quantum Demiric-Selçuk meet-in-the-middle attacks: Applications to 6-round generic Feistel constructions. In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 386–403. Springer, 2018.

[HS20] Akinori Hosoyamada and Yu Sasaki. Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT*

2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II, volume 12106 of *Lecture Notes in Computer Science*, pages 249–279. Springer, 2020.

[HS21] Akinori Hosoyamada and Yu Sasaki. Quantum collision attacks on reduced SHA-256 and SHA-512. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 616–646. Springer, 2021.

[HS22] Zhenyu Huang and Siwei Sun. Synthesizing quantum circuits of AES with lower t-depth and less qubits. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part III*, volume 13793 of *Lecture Notes in Computer Science*, pages 614–644. Springer, 2022.

[HSTX19] Akinori Hosoyamada, Yu Sasaki, Seiichiro Tani, and Keita Xagawa. Improved quantum multicollision-finding algorithm. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers*, volume 11505 of *Lecture Notes in Computer Science*, pages 350–367. Springer, 2019.

[HSX17] Akinori Hosoyamada, Yu Sasaki, and Keita Xagawa. Quantum multicollision-finding algorithm. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 179–210. Springer, 2017.

[IHM+19] Gembu Ito, Akinori Hosoyamada, Ryutaroh Matsumoto, Yu Sasaki, and Tetsu Iwata. Quantum chosen-ciphertext attacks against Feistel ciphers. In Mitsuru Matsui, editor, *Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings*, volume 11405 of *Lecture Notes in Computer Science*, pages 391–411. Springer, 2019.

[IK03] Tetsu Iwata and Kaoru Kurosawa. OMAC: One-Key CBC MAC. In Thomas Johansson, editor, *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, volume 2887 of *Lecture Notes in Computer Science*, pages 129–153. Springer, 2003.

[IMPS17] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A fast tweakable block cipher mode for highly secure message authentication. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO*

2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III, volume 10403 of *Lecture Notes in Computer Science*, pages 34–65. Springer, 2017.

[ISO19] ISO/IEC. ISO/IEC 29192-6:2019 Information technology — Lightweight cryptography Part 6: Message authentication codes (MACs), 2019.

[JNRV20] Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. Implementing grover oracles for quantum key search on AES and LowMC. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 280–310. Springer, 2020.

[JS19] Samuel Jaques and John M. Schanck. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 32–61. Springer, 2019.

[Kap16] Marc Kaplan. Quantum attacks against iterated block ciphers. *Mat. Vopr. Kriptogr.*, 7:71–90, 2016.

[KHJ18] Panjin Kim, Daewan Han, and Kyung Chul Jeong. Time–space complexity of quantum search algorithms in symmetric cryptanalysis: Applying to AES and SHA-2. *Quantum Information Processing*, 17(12):339, 2018.

[KK06] John Kelsey and Tadayoshi Kohno. Herding hash functions and the Nostradamus attack. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 183–200. Springer, 2006.

[KLLN16a] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 207–237. Springer, 2016.

[KLLN16b] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Quantum differential and linear cryptanalysis. *IACR Trans. Symmetric Cryptol.*, 2016(1):71–94, 2016.

[KM10] Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In *IEEE International Sympo-*

sium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings, pages 2682–2685. IEEE, 2010.

[KM12] Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type Even-Mansour cipher. In *Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012*, pages 312–316. IEEE, 2012.

[KR96] Joe Kilian and Phillip Rogaway. How to protect DES against exhaustive key search. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 252–267. Springer, 1996.

[KR11] Ted Krovetz and Phillip Rogaway. The software performance of authenticated-encryption modes. In Antoine Joux, editor, *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, volume 6733 of *Lecture Notes in Computer Science*, pages 306–327. Springer, 2011.

[Kup05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.*, 35(1):170–188, 2005.

[LGQW23] Zhenqiang Li, Fei Gao, Sujuan Qin, and Qiaoyan Wen. New record in the number of qubits for a quantum implementation of AES. *Frontiers in Physics*, 11:1171753, 2023.

[LH24] Dongjae Lee and Seokhie Hong. Improved quantum rebound attacks on double block length hashing with round-reduced AES-256 and ARIA-256. *IACR Trans. Symmetric Cryptol.*, 2024(3):238–265, 2024.

[LKL+24] Jongheon Lee, Yousung Kang, You-Seok Lee, Boheung Chung, and Dooho Choi. Toffoli-depth reduction method preserving in-place quantum circuits and its application to SHA3-256. *Quantum Information Processing*, 23(4):153, 2024.

[LLLC23] Jongheon Lee, Sokjoon Lee, You-Seok Lee, and Dooho Choi. T-depth reduction method for efficient SHA-256 quantum circuit construction. *IET Information Security*, 17(1):46–65, 2023.

[LLW24] Yingxin Li, Fukang Liu, and Gaoli Wang. New records in collision attacks on SHA-2. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I*, volume 14651 of *Lecture Notes in Computer Science*, pages 158–186. Springer, 2024.

[LM17] Gregor Leander and Alexander May. Grover meets Simon - quantumly attacking

the FX-construction. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 161–178. Springer, 2017.

[LPS20] Brandon Langenberg, Hai Pham, and Rainer Steinwandt. Reducing the cost of implementing the advanced encryption standard as a quantum circuit. *IEEE Transactions on Quantum Engineering*, 1:1–12, 2020.

[LPTY16] Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda. A MAC mode for lightweight block ciphers. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 43–59. Springer, 2016.

[LPZW23] Qun Liu, Bart Preneel, Zheng Zhao, and Meiqin Wang. Improved quantum circuits for AES: Reducing the depth and the number of qubits. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part III*, volume 14440 of *Lecture Notes in Computer Science*, pages 67–98. Springer, 2023.

[LR85] Michael Luby and Charles Rackoff. How to construct pseudo-random permutations from pseudo-random functions (abstract). In Hugh C. Williams, editor, *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*, volume 218 of *Lecture Notes in Computer Science*, page 447. Springer, 1985.

[LRW02] Moses D. Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable block ciphers. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2002.

[LZ19] Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 189–218. Springer, 2019.

[MMH+14] Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: An efficient MAC algorithm for 32-bit microcontrollers. In Antoine Joux and Amr M. Youssef, editors, *Selected Areas in*

Cryptography - SAC 2014 - 21st International Conference, Montreal, QC, Canada, August 14-15, 2014, Revised Selected Papers, volume 8781 of *Lecture Notes in Computer Science*, pages 306–323. Springer, 2014.

[MV04] David A. McGrew and John Viega. The security and performance of the Galois/Counter Mode (GCM) of operation. In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.

[Nat77] National Bureau of Standards. Data encryption standard. FIPS 46, January 1977.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 2010.

[NIDI19] Boyu Ni, Gembu Ito, Xiaoyang Dong, and Tetsu Iwata. Quantum attacks against type-1 generalized Feistel ciphers and applications to CAST-256. In Feng Hao, Sushmita Ruj, and Sourav Sen Gupta, editors, *Progress in Cryptology - INDOCRYPT 2019 - 20th International Conference on Cryptology in India, Hyderabad, India, December 15-18, 2019, Proceedings*, volume 11898 of *Lecture Notes in Computer Science*, pages 433–455. Springer, 2019.

[NIS01] NIST. Advanced Encryption Standard (AES). NIST FIPS PUB 197, 2001.

[NIS05] NIST. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST SP 800-38B, 2005.

[NIS24a] NIST. Module-lattice-based digital signature standard. NIST FIPS PUB 204, 2024.

[NIS24b] NIST. Module-lattice-based key-encapsulation mechanism standard. NIST FIPS PUB 203, 2024.

[NIS24c] NIST. Stateless hash-based digital signature standard. NIST FIPS PUB 205, 2024.

[NS20] María Naya-Plasencia and André Schrottenloher. Optimal merging in quantum k-xor and k-xor-sum algorithms. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 311–340. Springer, 2020.

[Oec03] Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 617–630. Springer, 2003.

[Pol75] JM Pollard. A Monte Carlo method for factorization. *BIT Numerical Mathematics*,

15(3):331–334, 1975.

[Pre22] Richard H Preston. Applying Grover's algorithm to hash functions: a software perspective. *IEEE Transactions on Quantum Engineering*, 3:1–10, 2022.

[RBBK01] Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In Michael K. Reiter and Pierangela Samarati, editors, *CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6-8, 2001*, pages 196–205. ACM, 2001.

[Rog02] Phillip Rogaway. Authenticated-encryption with associated-data. In Vijayalakshmi Atluri, editor, *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, pages 98–107. ACM, 2002.

[Rog04] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2004.

[RS15] Martin Rötteler and Rainer Steinwandt. A note on quantum related-key attacks. *Inf. Process. Lett.*, 115(1):40–44, 2015.

[SBK+17] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full SHA-1. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 570–596. Springer, 2017.

[Sch21] André Schrottenloher. Improved quantum algorithms for the k-xor problem. In Riham AlTawy and Andreas Hülsing, editors, *Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers*, volume 13203 of *Lecture Notes in Computer Science*, pages 311–331. Springer, 2021.

[Sch23] André Schrottenloher. Quantum linear key-recovery attacks using the QFT. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Proceedings, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 258–291. Springer, 2023.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134. IEEE Computer Society,

1994.

[Sim94] Daniel R. Simon. On the power of quantum computation. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 116–123. IEEE Computer Society, 1994.

[SLG17] Ling Song, Guohong Liao, and Jian Guo. Non-full sbox linearization: Applications to collision attacks on round-reduced keccak. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 428–451. Springer, 2017.

[SS22] André Schrottenloher and Marc Stevens. Simplified MITM modeling for permutations: New (quantum) attacks. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 717–747. Springer, 2022.

[STKT08] Kazuhiro Suzuki, Dongvu Tonien, Kaoru Kurosawa, and Koji Toyota. Birthday paradox for multi-collisions. *IEICE Transactions*, 91-A(1):39–45, 2008.

[TMC+24] Meltem Sönmez Turan, Kerry McKay, Donghoon Chang, Jinkeon Kang, and John Kelsey. Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions. NIST SP 800-232 (Initial Public Draft), 2024.

[vOW94] Paul C. van Oorschot and Michael J. Wiener. Parallel collision search with application to hash functions and discrete logarithms. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, and Ravi S. Sandhu, editors, *CCS '94, Proceedings of the 2nd ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 2-4, 1994*, pages 210–218. ACM, 1994.

[WH87] Robert S. Winternitz and Martin E. Hellman. Chosen-key attacks on a block cipher. *Cryptologia*, 11(1):16–20, 1987.

[WYY05] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2005.

[WZ11] Wenling Wu and Lei Zhang. LBlock: A lightweight block cipher. In Javier López and Gene Tsudik, editors, *Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings*,

volume 6715 of *Lecture Notes in Computer Science*, pages 327–344, 2011.

[XWY+24] Zejun Xiang, Xiaoyu Wang, Bo Yu, Bing Sun, Shasha Zhang, Xiangyong Zeng, Xuan Shen, and Nian Li. Links between quantum distinguishers based on Simon's algorithm and truncated differentials. *IACR Trans. Symmetric Cryptol.*, 2024(2):296–321, 2024.

[Yas11] Kan Yasuda. A new variant of PMAC: Beyond the birthday bound. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 596–609. Springer, 2011.

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information & Computation*, 15(7&8):557–567, 2015.

[ZSWS24] Mengyuan Zhang, Tairong Shi, Wenling Wu, and Han Sui. Optimized quantum circuit of AES with interlacing-uncompute structure. *IEEE Transactions on Computers*, 73(11):2563–2575, 2024.

[ZWS+20] Jian Zou, Zihao Wei, Siwei Sun, Ximeng Liu, and Wenling Wu. Quantum circuit implementations of AES with fewer qubits. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 697–726. Springer, 2020.

[デ 24] デジタル庁，総務省，経済産業省. 電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）, 2024. 文書番号 CRYPTREC LS-0001-2022R1.

[細 20] 細山田光倫. 量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価, 2020. 報告書文書番号 CRYPTREC EX-2901-2019.

資料４－１

2024 年度　暗号技術活用委員会活動報告

1.　2024 年度の活動概要

1.1　活動目的

暗号技術活用委員会では、情報システム全般のセキュリティ確保に寄与することを目的として、暗号の取り扱いに関する観点から、運用ガイドライン／ガイダンスの作成を行う。

2024 年度は、2023 年度から引き続き暗号鍵管理ガイダンス WG を設置して、2022 年度の成果として発行した暗号鍵管理ガイダンスの拡充を行う。暗号鍵管理ガイダンスの拡充は今年度の完成を目標とする。また、新たなガイダンスとして、クラウドにおける鍵管理ガイダンスの作成作業を開始する。

1.2　活動概要

今年度の活動概要は以下の通りである。

(1) 暗号鍵管理ガイダンスの拡充

暗号鍵管理ガイドラインの拡充を目的として、2020 年度に発行した「暗号鍵管理システム設計指針（基本編）（以降「設計指針」と表記）」の副読本として、「暗号鍵管理ガイダンス」を作成する。本活動は暗号鍵管理ガイダンス WG にて実施する。具体的には、2021 から 2022 年度に作成した「暗号鍵管理ガイダンス Ver.1.0」ではカバーできていない項目について、同ガイダンスの拡充を行う。2024 年度の完成を目標とする。

(2) クラウドにおける鍵管理ガイダンスの検討

以下のテーマに関する新たなガイダンスの作成に着手する。おおむね 2 年程度での完成を想定して執筆作業を行う。

- クラウドにおける鍵管理ガイダンス（日本クラウドセキュリティアライアンス（CSA）と共同での作成も検討）

※クラウド利用者が留意すべき鍵管理を解説することを目的とする

1.3　暗号技術活用委員会の委員構成及び開催状況

暗号技術活用委員会の委員構成は表 1-1 のとおりである。また、2024 年度に開催された暗号技術活用委員会での議案は表 1-2 のとおりである。

表 1-1 暗号技術活用委員会　委員構成

委員長	松本　勉	国立研究開発法人産業技術総合研究所　フェロー 横浜国立大学　先端科学高等研究院　上席特別教授
委員	上原　哲太郎	立命館大学　情報理工学部　情報理工学科　教授
委員	垣内　由梨香	Microsoft Corporation　セキュリティレスポンスチーム セキュリティプログラムマネージャー
委員	菊池　浩明	明治大学　総合数理学部　先端メディアサイエンス学科　教授
委員	佐藤　直之	SCSK セキュリティ株式会社　コンサルティング本部 シニアプロフェッショナルコンサルタント
委員	佐藤　雅史	セコム株式会社　IS 研究所 デジタルプラットフォームディビジョン　主幹研究員
委員	須賀　祐治	株式会社インターネットイニシアティブ　セキュリティ本部 セキュリティ情報統括室　シニアエンジニア
委員	田村　裕子	日本銀行　金融研究所　情報技術研究センター　企画役
委員	手塚　悟	慶應義塾大学　グローバルリサーチインスティテュート　特任教授
委員	寺村　亮一	GMOサイバーセキュリティbyイエラエ株式会社 上席執行役員　サイバーセキュリティ事業本部長
委員	三澤　学	三菱電機株式会社　情報技術総合研究所 情報ネットワークシステム技術部　グループマネージャ
委員	満塩　尚史	順天堂大学　健康データサイエンス学部　健康データサイエンス学科 准教授
委員	山口　利恵	東京大学　大学院情報理工学系研究科 ソーシャル ICT 研究センター　准教授
委員	渡邊　創	国立研究開発法人産業技術総合研究所 サイバーフィジカルセキュリティ研究センター　副研究センター長

（2025 年 3 月 4 日現在）

表 1-2　暗号技術活用委員会　開催状況

回	開催日	議案
メール審議	2024 年 6 月	● 2024 年度暗号鍵管理ガイダンス WG 活動計画の審議
第一回	2024 年 10 月 28 日	● 2024 年度暗号技術活用委員会活動計画の確認 ● 2024 年度暗号鍵管理ガイダンス WG 活動計画の確認 ● 暗号鍵管理ガイダンス WG 進捗報告 ● クラウドにおける鍵管理ガイダンスについて

第二回	2025 年 3 月 4 日	● 2024 年度暗号鍵管理ガイダンス WG 活動報告及びガイダンス案の審議 ● クラウドにおける鍵管理ガイダンスについて ● 2024 年度暗号技術活用委員会活動報告案について

2. 成果概要

以下に成果概要の要約を記載する。詳細については、CRYPTREC Report 2024 暗号技術活用委員会報告[1]を参照されたい。

2.1 暗号鍵管理ガイダンスの拡充

本年度は、ガイダンス Ver.1.0 において記載を見送った部分について追補版のガイダンスを執筆した。追補版はガイダンス Ver.1.0 の分冊であるため、ガイダンス Ver.1.0 を「暗号鍵管理ガイダンス Part 1」、今年度執筆した追補版を「暗号鍵管理ガイダンス Part 2」と呼ぶこととした。

以下にガイダンス Part 2 の概要をまとめる。

暗号鍵管理ガイダンス Part 2 の位置づけ

- 暗号鍵管理機能を持つシステム設計者向けのガイダンスである。このガイダンスは 2020 年に発行した「設計指針」に記載された各検討項目（Framework Requirement：以降 FR）をより詳細に解説した副読本である
- 「設計指針」に記載された各検討項目の解釈に役立つ、検討項目の背景や補足事項、実システムでの設計において検討項目に基づいた要求事項を含めるかどうかの判断材料などを解説する
- 暗号鍵管理システムのシンプルなモデル（トイモデル）を例示し、トイモデルにおける各検討項目への対応例を説明することで、各検討項目の内容や思想の理解を促進する

想定読者

- 暗号鍵管理機能を持つシステムの設計者

暗号鍵管理ガイダンス Part 2 の章構成

暗号鍵管理ガイダンス Part 2 の構成は、「設計指針」の章構成に対応して表 2-1 のとおりである。

[1] CRYPTREC Report 2024 暗号技術活用委員会報告, https://www.cryptrec.go.jp/promo_cmte.html

表 2-1　暗号鍵管理ガイダンスの章構成

暗号鍵管理システム設計指針（基本編）	暗号鍵管理ガイダンス Part 1（2023 年 5 月発行）	暗号鍵管理ガイダンス Part 2（今年度執筆）
1. はじめに	1. はじめに	1. はじめに
2. 暗号鍵管理の在り方	（1 章に集約）	（1 章に集約）
3. 本設計指針の活用方法	（1 章に集約）	（1 章に集約）
4. 暗号鍵管理システム（CKMS）の設計原理と運用ポリシー		2. 暗号鍵管理システム（CKMS）の設計原理と運用ポリシー
5. 暗号アルゴリズム運用のための暗号鍵管理オペレーション対策	2. 暗号アルゴリズム運用のための暗号鍵管理オペレーション対策	
6. 暗号アルゴリズムの選択	3. 暗号アルゴリズムの選択	
7. 暗号アルゴリズム運用に必要な鍵情報の管理	4. 暗号アルゴリズム運用に必要な鍵情報の管理	
8. 暗号鍵管理デバイスへのセキュリティ対策		3. 暗号鍵管理デバイスへのセキュリティ対策
9. 暗号鍵管理システム（CKMS）のオペレーション対策		4. 暗号鍵管理システム（CKMS）のオペレーション対策

ガイダンス Part 1（2023 年 5 月発行）では、CKMS の利用環境に関わらず検討する必要がある項目のうち、「設計指針」の 5 章から 7 章に該当する項目に関して、項目の概説及びその記載例を提供している。これらの項目は「狭義」の意味での暗号鍵管理に相当するものである。これらは CKMS を設計する場合だけでなく、暗号アルゴリズムを使ったアプリケーション等を利用する場合なども含めて、全ての暗号鍵管理に対して検討が必要となる項目である。

ガイダンス Part 2（今年度執筆）では、「設計指針」の 4 章、8 章、9 章に該当する項目に関して、項目の概説及びその記載例を提供している。Part 2 の 2 章は CKMS の利用環境に関わらず検討する必要がある項目のうち、CKMS の全体方針を定める項目である。Part 2 の 3 章は CKMS に利用するデバイス管理を含む場合に検討すべき項目であり、Part 2 の 4 章は CKMS のシステム管理を含む場合に検討すべき項目である。Part 2 の 3 章や 4 章までを含む場合、「広義」の意味での暗号鍵管理に相当する内容となる。

ガイダンス Part 2 の各章の記載概要は以下のとおりである。

1 章は、イントロダクションとして、本ガイダンス Part 2 の位置づけについて、「設計指針」やガイダンス Part 1 との関係を含めて記載した。また、本ガイダンス Part 2 におけるトイモデルとして設定した「IoT 機器（家電製品を想定）向けに公開鍵証明書を発行するプライベート CA システム」の概要を説明した。

2 章は、「設計指針」での「暗号鍵管理システム(CKMS)の設計原理と運用ポリシー」におけ

る検討項目について解説・考慮点を記載した。本章は CKMS として実現すべき全体方針に関わる検討項目であり、ガイダンス Part 2 における以降の章やガイダンス Part 1 内の各章における検討項目は本章の検討結果と整合をとって定めることになる。CKMS のセキュリティポリシー、CKMS に関わるエンティティの定義、CKMS を構成するデバイスやコンポーネントの一覧、CKMS での実現目標、CKMS に関わる法規制や標準化技術、将来的な移行対策などの検討項目から構成される。トイモデルとしてプライベート CA のセキュリティポリシーや運用の想定例を設定して、各検討項目の対応例を説明した。

3 章は、「設計指針」での「暗号鍵管理デバイスへのセキュリティ対策」における検討項目について解説・考慮点を記載した。本章は、CKMS におけるセキュアな暗号鍵管理・保管の中核となる暗号鍵管理デバイスへのアクセスコントロールに対する検討事項、暗号鍵管理デバイス内の暗号モジュールに対する検討項目、暗号鍵管理デバイス及び CKMS のセキュリティ評価・試験に関する検討項目、暗号鍵管理デバイスにおける障害発生時の BCP 対策に関わる検討項目などで構成される。トイモデルとして、プライベート CA で用いるハードウェア・セキュリティモジュールに関わる具体例として各検討項目の対応例を説明した。

4 章は、「設計指針」での「暗号鍵管理システム(CKMS)のオペレーション対策」における検討項目について解説・考慮点を記載した。本章は、CKMS における物理的セキュリティコントロール及びコンピュータシステムやネットワークにおけるセキュリティコントロールとそれらが危殆化した場合の BCP 対策、システム及びデバイスの開発プロセスやセキュリティメンテナンスに関わる検討項目、セキュリティアセスメントに関わる検討項目、CKMS 全体に関わる災害・障害発生時の BCP 対策などの検討項目で構成される。トイモデルとして、プライベート CA の設置環境や入退室管理、プライベート CA を構成するサーバシステムでのセキュリティコントロール、セキュリティアセスメントにおける実施項目、プライベート CA の災害復旧対策を想定して各検討項目の対応例を説明した。

2.2　クラウドにおける鍵管理ガイダンスの検討

新たなガイダンスとして、「クラウドにおける鍵管理ガイダンス（仮称)」を設定し、作成方針を検討した。議論を通して整理した概要を以降に説明する。

① 目的・想定読者・スコープについて

本ガイダンス作成の目的については以下のように整理した。

クラウドサービスを活用して効率的に情報システムを構築することは政府機関、民間企業を問わず一般的になっている。一方で、クラウドサービスの活用には、クラウドサービスに預けた情報が漏洩すること、設定不備やクラウドサービスにおける障害波及のリスクがあること、等のオンプレミスとは異なる懸念事項も生じる。クラウドサービスにおける暗号鍵管理システムを適切に選択・構築・運用することによって、そうした懸念事項に対処できる部分がある。クラウドサービスにおける暗号鍵管理の仕組みや注意事項を解説したガイダンスを作成し、クラウド環境で安全に暗号を運用するための一つの

ガイダンスとする。
クラウドサービスプロバイダが提供する鍵管理サービスを体系化し、ユースケースに応じてどのような鍵管理サービスを利用すべきかを判断できる情報を提供する。
クラウドサービスを導入した場合でも鍵管理において検討すべきことの全体は変わらないが、クラウドサービスの活用によって利用者や調達者が重点的に検討すべきことは変わる。どういう項目に重点が置かれるかを整理する。

上記の目的でガイダンスを作成することの意義について、クラウドサービスプロバイダや政府系プラットフォームの構築者にヒアリングしたところ、そもそもクラウドサービスにおける鍵の管理にどのような事項があり、誰がそれを管理するのかといった内容について、利用者や SI 事業者の理解が十分でないとの意見をいただいた。そこで、上記の目的に加えて、まずは鍵管理における基本的な事項の解説に重点を置くこととした。

本ガイダンスの想定読者については以下とした。
クラウドサービスを利用した情報システムの構築者（SI 事業者）、運用者、利用者。

本ガイダンスのスコープについては以下のように整理した。
IaaS や PaaS のクラウドサービスを利用して、情報システムのプラットフォーム構築を行うケースを対象に、どのような鍵管理サービスを提供すべきかをターゲットとする。
暗号機能による保護の対象はクラウドサービスに預けたデータ及び鍵情報の機密性と完全性の確保、並びに暗号化消去とする。

上記のスコープについては、クラウドにおける鍵管理には多様な形態が含まれ、例えばクラウドサービスプロバイダの IaaS 上に SaaS 事業者が介在する場面もあるので、議論が発散しないようにスコープを定義すべきであるとの意見を元に設定した。設定の上では、クラウドサービスプロバイダ、SI 事業者、政府系プラットフォームの構築者にヒアリングした結果と活用委員会での委員の意見を踏まえた。ただし、最初からスコープを狭くしすぎるのはまずいのではないかとの意見もあり、スコープの妥当性については引き続き検討することとした。

② 記載内容のポイントについて
本ガイダンス記載内容のポイントとなる事項について、以下の点を設定した。

- クラウド鍵管理サービスの分類
  クラウドサービスプロバイダ(CSP)が提供する鍵管理サービスを体系化し、ユースケースに応じてどのような鍵管理サービスを利用すべきかを判断できる情報を提供する。

  上記に関して、クラウドサービスプロバイダが提供する鍵管理サービスとしては、「ク

ラウドネイティブ暗号化」、「BYOK（Bring Your Own Key）」、「HYOK（Hold Your Own Key）」、「BYOE（Bring Your Own Encryption）」と呼ばれる分類がある[2]が、必ずしも明確な定義は与えられていない。本ガイダンスではこれらの用語に定義を与える代わりに、鍵管理サービスの分類軸となる項目を明確にして鍵管理サービスを分類することとした。

- クラウド鍵管理サービスに関わる責任分界について
  クラウドサービス利用時の鍵管理システムに関わる CSP との責任分界について、クラウド鍵管理サービスの種類に応じて原則となる考え方を整理する。その際、クラウドサービスモデル（IaaS、PaaS、SaaS）に依存する部分があるかについても現状を整理する。

  上記のクラウドサービスモデルに依存する部分については、SI 事業者にヒアリングした中で次の見解をいただいた。すなわち、クラウドネイティブ暗号化と BYOK は IaaS、PaaS、SaaS の全てにおいて概ね利用できるが、HYOK や BYOE は適用できるクラウドサービスが限定されているとのことである。

- 暗号鍵管理ガイダンス（SP 800-130）の Framework Requirement との関係
  暗号鍵管理ガイダンスにおいて解説している検討項目（Framework Requirement）について、クラウドサービス利用時はどのように検討されるべきかを記載する。現在の暗号鍵管理ガイダンスでは、オンプレミスに構築した CKMS をトイモデルに設定して検討項目への対応例を記載しているため、クラウドサービスを利用した場合に重点的に検討すべき項目や対応例がどのように変わるかを説明する。

  上記に関しては SP 800-130 の検討項目の中で、暗号鍵のライフサイクル管理（鍵の生成、ローテーション、破棄など）、暗号鍵の保管方法、暗号鍵へのアクセス制御を重点的に検討すべき項目の候補と考えている。

③ 作成スケジュールおよび検討体制
本ガイダンスの作成にあたって、2025 年度より WG を新しく設置することとなった。WG 委員として CSP 事業者、SI 事業者・SaaS 事業者、クラウド HSM ベンダ、クラウドサービス利用者、大学や関連団体などの有識者にそれぞれ参画いただき、事務局を中心に委員の知見をまとめる形での作成を予定している。

[2] 例えば、「Cloud Data Protection、バージョン 1.0」、日本クラウドセキュリティアライアンス（2021 年 8 月）。

作成スケジュールについては、WG の任期である 2 年間で遅くとも本ガイダンスの作成を行う計画である。ただし、技術やサービスの進展が早い領域であり、計画は柔軟に捉えることとしたい。ガイダンスに含める内容を明確にして作成を進めると共に、本ガイダンスに続いて検討すべき内容も並行して議論していく。

3. 今後に向けて

2025 年度は、新たなガイダンスである「クラウドにおける鍵管理ガイダンス（仮称）」について WG を設置し、ガイダンス作成を本格的に開始する予定である。また、分冊構成とした暗号鍵管理ガイダンスについて合冊化の検討や、同ガイダンス作成中に「設計指針」記載内容に対して修正意見があった事項の対応については今後の課題である。

資料４－２

2024 年度　暗号鍵管理ガイダンス WG 活動報告

1.　2024 年度の活動概要

1.1.　活動目的と活動概要

CRYPTREC では、情報システム設計者やシステム調達者が暗号の利活用を適切に行うためのガイドライン作成を進めており、暗号鍵管理ガイダンス WG では暗号鍵管理が必要なシステムの設計者向けに、暗号鍵管理システム（CKMS：Cryptographic Key Management System）の設計において考慮すべき事項を解説したガイダンスの作成を行っている。本 WG では、2020 年度に発行した「暗号鍵管理システム設計指針（基本編）」の副読本として 2021 から 2022 年度で「暗号鍵管理ガイダンス Ver.1.0」を作成した。しかしながら、同ガイダンス Ver.1.0 は、あらゆる暗号鍵管理システムにおいて検討が必要となる共通項目に絞って解説を行ったものであり、「暗号鍵管理システム設計指針（基本編）」に記載された項目のすべてをカバーしたものではなかった。

本年度は、ガイダンス Ver.1.0 において記載を見送った部分について追補版のガイダンスを執筆した。分冊構成としたため、ガイダンス Ver.1.0 を「暗号鍵管理ガイダンス Part 1」、今年度執筆した追補版を「暗号鍵管理ガイダンス Part 2」と呼ぶこととした。

1.2.　暗号鍵管理ガイダンス WG の委員構成及び開催状況

暗号鍵管理ガイダンス WG の委員構成は表 1-1 のとおりである。また、2024 年度の開催状況は表 1-2 のとおりである。

表 1-1　暗号鍵管理ガイダンス WG 委員構成

主査	上原　哲太郎	立命館大学 情報理工学部 情報理工学科 教授
委員	泉　雅明	シスコシステムズ合同会社 東日本公共・法人システムズエンジニアリング ソリューションズエンジニアリング第 1 ソリューションズエンジニア
委員	漆嶌　賢二	GMO グローバルサイン株式会社 事業企画部 フェロー
委員	垣内　由梨香	Microsoft Corporation セキュリティレスポンスチーム セキュリティプログラムマネージャー
委員	菅野　哲	GMO サイバーセキュリティ by イエラエ株式会社 常務取締役 CTO of Development
委員	菊池　浩明	明治大学 総合数理学部 先端メディアサイエンス学科　教授
委員	小林　浩二	パナソニック オートモーティブシステムズ株式会社 開発本部プラットフォーム開発センター セキュリティ開発部 開発 2 課 2 係　係長
委員	須賀　祐治	株式会社インターネットイニシアティブ セキュリティ本部セキュリティ情報統括室　シニアエンジニア

委員	舟木　康浩	タレス DIS ジャパン株式会社 クラウドプロテクション＆ライセンシング データプロテクション事業本部 セールスエンジニアマネージャ
委員	程吉　英仁	株式会社 NTT データ　ソリューション事業本部 セキュリティ＆ネットワーク事業部 サイバーセキュリティ統括部 課長代理
委員	満塩　尚史	順天堂大学　健康データサイエンス学部 健康データサイエンス学科 准教授

（2025 年 1 月 22 日現在）

表 1-2　暗号鍵管理ガイダンス WG 開催状況

回	開催日	議事概要
第一回	2024 年 7 月 30 日	■ 2024 年度 WG 活動計画の確認 ■ 「暗号鍵管理システムのオペレーション対策」の記載概要に関する審議 ■ ガイダンス執筆方針及びレビュー計画に関わる審議
メール審議	2024 年 8 月 ～10 月	■ 「暗号鍵管理デバイスへのセキュリティ対策」ドラフトのレビュー（7/30～9/13） ■ 「暗号鍵管理システムの設計原理と運用ポリシー」ドラフトのレビュー（10/1～11/1）
第二回	2024 年 11 月 18 日	■ 「暗号鍵管理デバイスへのセキュリティ対策」ドラフトの審議 ■ 「暗号鍵管理システムの設計原理と運用ポリシー」ドラフトの審議
メール審議	2024 年 11 月 ～2025 年 1 月	■ 「暗号鍵管理システムのオペレーション対策」ドラフトのレビュー（11/20～1/17） ■ 「はじめに」ドラフトのレビュー（12/5～1/17）
第三回	2025 年 1 月 22 日	■ 「はじめに」ドラフトの審議 ■ 「暗号鍵管理システムのオペレーション対策」ドラフトの審議 ■ ガイダンス全体に関わる審議
メール審議	2025 年 1 月 ～2 月	■ ガイダンスの最終レビュー（1/27～2/14）

2.　成果概要

2.1.　暗号鍵管理ガイダンス Part 2 の位置づけと章構成

暗号鍵管理ガイダンス Part 2 の位置づけ及び想定読者は次のとおりである。これらは同ガイダンス Part 1 と同一である。

暗号鍵管理ガイダンス Part 2 の位置づけ

- 暗号鍵管理機能を持つシステム設計者向けのガイダンスである。このガイダンスは 2020 年に発行した「暗号鍵管理システム設計指針（基本編）」に記載された各検討項目（Framework Requirement：以降 FR）をより詳細に解説した副読本である
- 「暗号鍵管理システム設計指針（基本編）」に記載された各検討項目の解釈に役立つ、検討項目の背景や補足事項、実システムでの設計において検討項目に基づいた要求事項を含めるかどうかの判断材料などについて解説する
- 暗号鍵管理システムのシンプルなモデル（トイモデル）を例示し、トイモデルにおける各検討項目への対応例を説明することで、各検討項目の内容や思想の理解を促進する

想定読者

- 暗号鍵管理機能を持つシステムの設計者

暗号鍵管理ガイダンス Part 2 の章構成

暗号鍵管理ガイダンス Part 2 の構成は「暗号鍵管理システム設計指針（基本編）」の章構成に対応して表 2-1 のとおりである。

表 2-1　暗号鍵管理ガイダンスの章構成

暗号鍵管理システム設計指針（基本編）	暗号鍵管理ガイダンス Part 1（2023 年 5 月発行）	暗号鍵管理ガイダンス Part 2（今年度執筆）
1. はじめに	1. はじめに	1. はじめに
2. 暗号鍵管理の在り方	（1 章に集約）	（1 章に集約）
3. 本設計指針の活用方法	（1 章に集約）	（1 章に集約）
4. 暗号鍵管理システムの設計原理と運用ポリシー		2. 暗号鍵管理システムの設計原理と運用ポリシー
5. 暗号アルゴリズム運用のための暗号鍵管理オペレーション対策	2. 暗号アルゴリズム運用のための暗号鍵管理オペレーション対策	
6. 暗号アルゴリズムの選択	3. 暗号アルゴリズムの選択	
7. 暗号アルゴリズム運用に必要な鍵情報の管理	4. 暗号アルゴリズム運用に必要な鍵情報の管理	
8. 暗号鍵管理デバイスへのセキュリティ対策		3. 暗号鍵管理デバイスへのセキュリティ対策
9. 暗号鍵管理システムのオペレーション対策		4. 暗号鍵管理システムのオペレーション対策

2.2. 「はじめに」の章

本年度の WG で議論した事項の概要は以下のとおりである。

① トイモデルについて

本ガイダンスにおけるトイモデルは IoT 製品向けのプライベート CA システム（図 2-1）を設定した。本ガイダンスの各章において同一のトイモデルを扱うこととした。

トイモデルで扱う「プライベート CA システム」の概要

- CKMS の範囲を CA サーバ、HSM（Hardware Security Module）、ルータまでとする
- プライベート CA システムは IoT 製品（家電想定）向けに公開鍵証明書の発行及び証明書の失効管理に使われる CRL の発行を行う
- IoT 製品向けの ID 管理、プライベート鍵生成、発行された証明書の機器埋め込みは工場内で行う
- IoT 製品は運用時にインターネット接続され、スマートフォン内専用アプリから IoT 製品ハブ経由で状態の監視や制御が行われる。証明書は専用アプリと IoT 機器の接続（TLS での認証と秘匿通信確立）に利用される

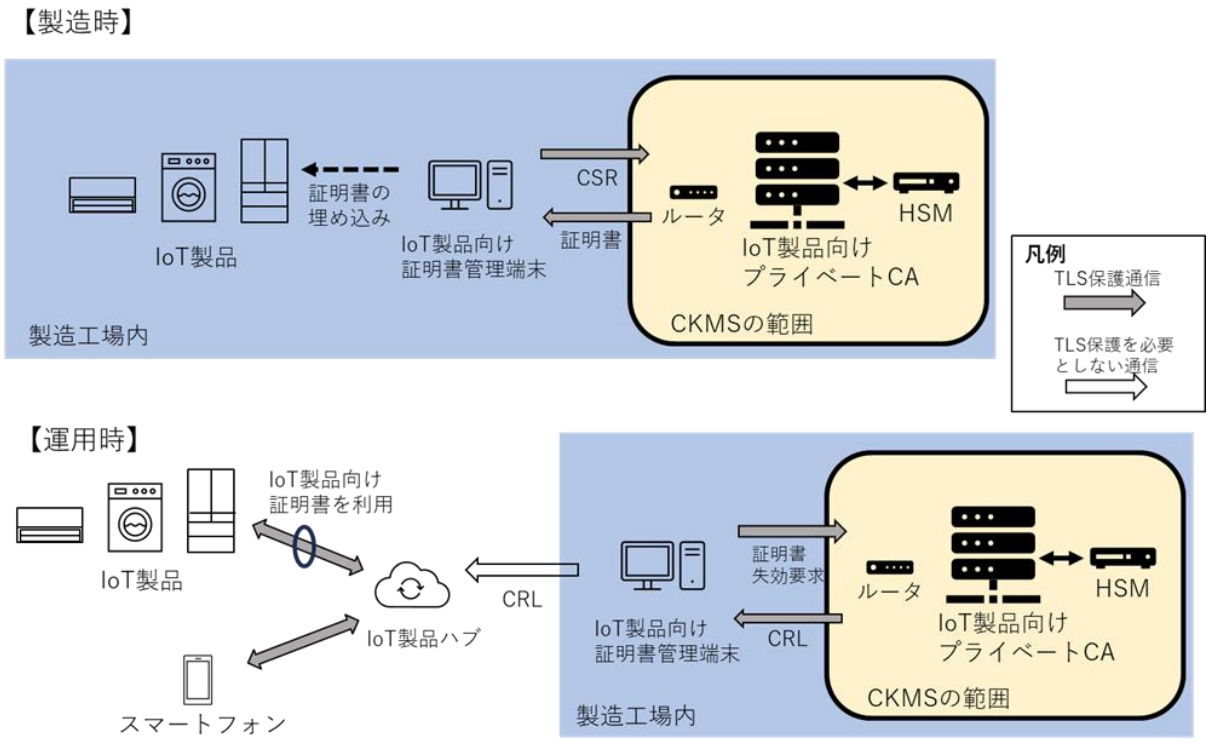


図 2-1　トイモデル（プライベート CA システム）の概要図

トイモデルに関して、今回は CA 秘密鍵の管理に HSM を利用しているが、同様のシステムを設計する際に HSM を導入しないといけないと読者に誤解を与えないかとの意見があった。これ

に対して、HSM 以外の選択肢もあるが、その場合は HSM に代わる秘密鍵の保護に関する様々な対応が必要になることを注記した。また、トイモデルはあくまで FR の説明のための参考例であり、この内容を推奨しているわけではないことの注意が本文に書かれていることを確認した。

2.3. 「暗号鍵管理システムの設計原理と運用ポリシー」の章

本年度の WG で議論した事項の概要は以下のとおりである。

① セキュリティポリシーと準拠する規格・標準の関係について

CKMS セキュリティポリシーに関して、CKMS を運用する組織に存在する様々なセキュリティに関わるポリシーとの依存関係を整理する FR がある。

CKMS より上位に位置づけられるポリシー及び CKMS より下位のポリシーについて例示を行った際に、上位ポリシーの例として、製品規格のセキュリティポリシー、NIST SP 800-57 Part 2（鍵管理セキュリティポリシー）、IoT 製品に対するセキュリティ適合性評価制度構築方針（経済産業省）を追加してはどうかとの意見が出された。このことを契機に、「セキュリティポリシー（2.1 節）」と「準拠すべき標準・規格（2.6 節）」の関係を整理した。

セキュリティポリシーは準拠すべき規格を元に各組織で定める性格のものとするのが適当ではないかと判断し、製品規格や IoT 製品に対するセキュリティ適合性評価制度構築方針は「準拠すべき標準・規格（2.6 節）」の具体例として記載した。また、NIST SP 800-57 Part 2 は組織が暗号鍵を安全に管理するためのベストプラクティスを提供し、鍵管理セキュリティポリシーの策定に役立つガイド文書に相当する内容と捉え、「セキュリティポリシー（2.1 節）」の解説・考慮点にその旨を記載した。

② 実現目標の記載例について

「CKMS の構築環境及び実現目標（2.5 節）」において、ネットワーク性能やパフォーマンス特性などの目標値を整理することが求められる。

トイモデルの想定事例として、IoT 製品に証明書を埋め込む生産ラインでの性能要求から CKMS（プライベート CA）での証明書の発行処理におけるネットワーク性能及びパフォーマンス性能の目標値を規定していたが、証明書は事前に発行しておき、製造工程で製品に証明書を埋め込む処理は発行処理とは分れるのが一般的ではないかとの指摘があった。指摘のような運用が一般的と考えられることから、IoT 製品の生産ラインと連動してオンラインで証明書を発行する必要はないが、年間の生産台数の計画から証明書発行に関わるパフォーマンスが規定されるものとして事例を修正した。

2.4. 「暗号鍵管理デバイスへのセキュリティ対策」の章

本年度の WG で議論した事項の概要は以下のとおりである。

① トイモデルにおけるアクセスコントロールシステム（ACS）について

「鍵情報へのアクセスコントロール（3.1 節)」において、鍵情報へのアクセスコントロールへの要求事項を整理する FR がある。

トイモデルにおけるアクセスコントロールシステムの想定事例では、CA サーバにおける ACS と HSM における ACS を連携させて構成するものとした。このうち、HSM の ACS に関わるエンティティ認証、及び権限認可について、製品事例として以下の情報提供があった。

- HSM 製品は一般に PKCS #11 ベースのユーザ管理となっており、セキュリティ・オフィサー （HSM の管理者ロール）とクリプト・オフィサー（HSM 内の鍵生成や鍵利用を行うアプリケーションにおいて利用するユーザのロール）で HSM にログオンする
- HSM 管理者権限の取得などに利用するインタフェースとして、汎用 PC のシリアル接続や SSH 接続をサポートしている
- HSM 内の署名鍵を利用して署名付与を実施する際の処理概略は以下のとおりとなる。
  1. CA アプリケーションから HSM に対してクリプト・オフィサーの PIN で認証
  2. CA アプリケーションは利用する鍵の名称を指定して、鍵の探索を HSM に対して依頼し、HSM から鍵のオブジェクト・ハンドルを取得。
  3. CA アプリケーションはオブジェクト・ハンドルを指定して、提供するデータへの署名処理を HSM に対して依頼

上記を踏まえて、ACS 関連のトポロジー（図 2-2）と ACS に関わる処理フロー（図 2-3）を下図のとおりにまとめた。

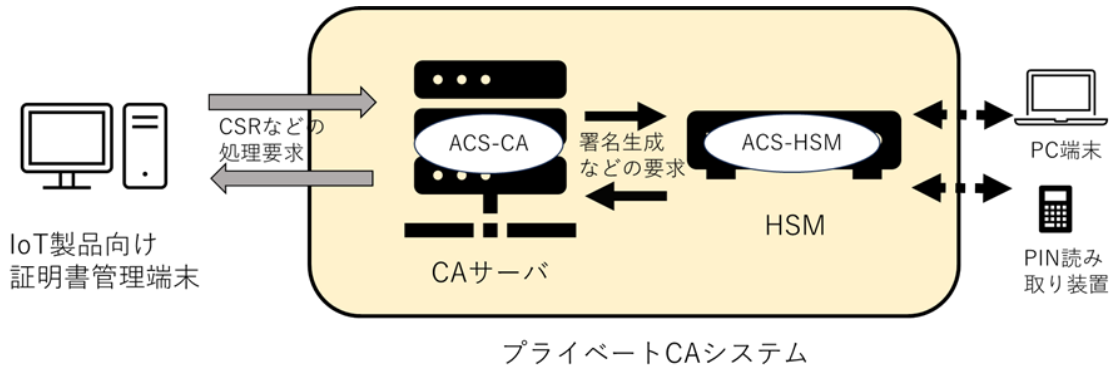


図 2-2　ACS の配置

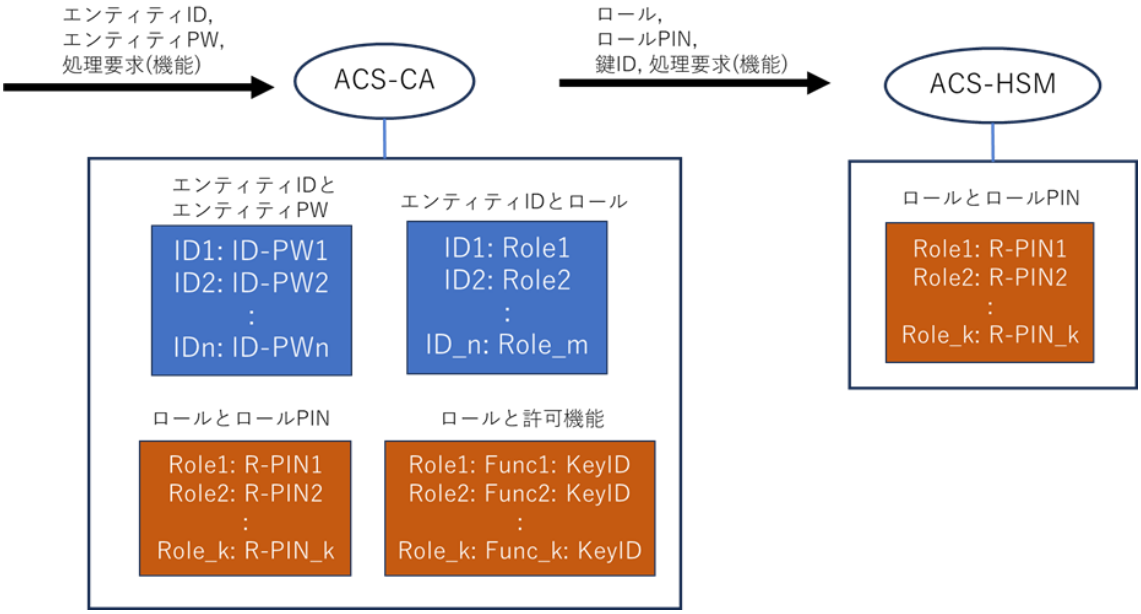


図 2-3　ACS に関わる処理フロー

CKMS のエンティティ、ACS、機能ロジックなどの接続配置については以下のとおりに記載した。

CA サーバ内の ACS-CA が要求を送出した利用者を識別・認証し、認証結果に応じて利用者に割り当てられた役割（ロール）の実行許可を与える。CA サーバは HSM に対して処理の依頼を利用者に代わって送出し、HSM から受け取った処理結果を利用者に返す。ここで、HSM 内の ACS-HSM は CA サーバのロールを認証して処理の実行を許可する。ACS に関わる処理フローは図 2-3 を参照。
一方、HSM 管理者の識別・認証は、CA サーバ内の ACS-CA を介さずに HSM 内の ACS-HSM によって行われる。その際、HSM に専用の PIN 読み取り装置を接続して認証処理を行う。また、HSM への管理用コマンドの送信には PC 端末から HSM がサポートしている方式（例えば、シリアル接続や SSH 等）で接続して処理を行う。

また、エンティティへの権限認可による署名付与の処理手順については以下のとおりに記載した。

CA サーバ及び HSM においてエンティティはロールベースで権限が付与される。ACS-CA にはエンティティの ID ごとに対応するロールを管理するデータベースがあり、さらにロールと許可された鍵管理機能および当該処理に利用する鍵 ID の対応を管理するデータベースがある。ACS-HSM では、PKCS #11 仕様に基づいてロールや鍵を管理しており、ACS-HSM で認証された利用者は、利用可能な鍵を検索して署名処理などを実行する。

② 相互運用性テストについて

「セキュリティ評価・試験（3.2 節)」の中に相互運用性を主張する際のテストの明確化に関わる FR がある。トイモデル（プライベート CA）が提供する機能は証明書の発行処理と証明書の失効処理における CA 署名の付与のみであり、通信相手は社内の証明書管理端末に限定される。さらに、プロトコルも単純であるため、トイモデルの記載例では相互運用性テストに相当するテストは不要で、API レベルの機能テストや結合テストで十分であると記載していた。

これに対して、相互運用性テストが不要であることはなく、ローカル環境やフィールドにおいて CKMS 提供機能の動作試験を行うものが該当するのではないかとの意見があった。委員の間でもどのようなテストが相互運用性テストに相当するかの見解が必ずしも一致していない状況を認識したので、本文書における相互運用性テストの解釈を記載することとした。具体的には、SP 800-130 該当節の記載から、CKMS レベルで他の CKMS との連携をテストするケースや保証ベースライン装置のようなゴールデンサンプルとの接続試験を行うものが相互運用性テストに該当する旨を追記した。

2.5. 「暗号鍵管理システムのオペレーション対策」の章

本年度の WG で議論した事項の概要は以下のとおりである。

① トイモデルにおける物理セキュリティコントロールについて

「CKMS へのアクセスコントロール（4.1 節)」において、CKMS デバイスやコンポーネントを保護するために設けられる物理的なセキュリティコントールを明確にする FR がある。

当該項目に関して、複数の物理アクセスコントロールの境界があり、敷地や建物、CKMS 収容エリアなどのレイヤによって求められる物理セキュリティの強度が異なること、さらにスイスチーズモデルのようにレイヤ構造の同じところに穴が開かないことを示せるとよいとの指摘があった。指摘を踏まえて、トイモデルにおける想定事例を図 2-4 のとおりにまとめた。各領域へのアクセスに必要となる認証情報がそれぞれ異なり、同一の認証情報によって複数の領域へのアクセスが可能とならないように設計していることを示した。

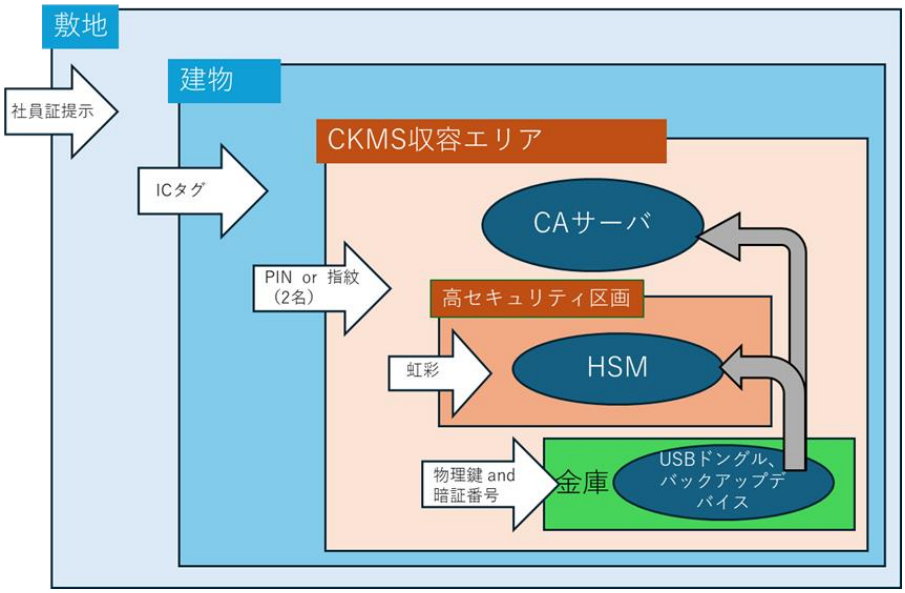


図 2-4　トイモデルにおける物理アクセスコントロール

②　トイモデルにおける災害復旧対策について

「CKMS 設備への障害・災害発生時の BCP 対策（4.5 節）」において、CKMS 設備における物理的損害が発生した場合の復旧対策を明確にする FR がある。地震及び水害などの自然災害や火災及び事故などの人為災害をも想定した内容であり、災害復旧対策と呼ばれる検討項目である。CKMS 運用の価値と機微度にふさわしいレベルでの対策を講じることが原則であり、トイモデルではどのような対策を想定するのが妥当かを議論した。

トイモデルは IoT 製品の製造拠点に設けられたプライベート CA を CKMS としており、災害復旧対策の基準は IoT 機器の製造設備と同等とするのが妥当であろうとなった。また、一般に IoT 製品の製造拠点は、現地生産等を目的に分散して設けられていることが多く、ある拠点がダウンしても他の拠点が製造をカバーすることができ、自ずと災害復旧対策にもなる。今回の CKMS のレベルで災害復旧の目的で複数サイトを用意することは現実的ではないとの意見があった。

議論を踏まえて、CKMS の災害復旧対策を目的としたプライベート CA の冗長化は行わず、障害対策として CA サーバのディスク冗長化、予備デバイスの準備、HSM 内の鍵情報を含むバックアップ取得と保管を実施する事例とした。ここで、バックアップ情報は災害復旧の目的で他の製造拠点にも保管するものとした。

3.　今後に向けて

「暗号鍵管理システム設計指針（基本編）」の副読本である「暗号鍵管理ガイダンス」を作成する活動は今年度で終了する。暗号鍵管理ガイダンス Part 1 と Part 2 を合体させるかどうかについては、今後のガイダンス読者の反応などに鑑みて判断する予定である。

執筆した暗号鍵管理ガイダンスにおいて設定したトイモデルは、オンプレミスに構築する CKMS を想定したが、近年はクラウドサービスを活用してシステム構築をするケースが増加している。クラウドサービスを活用した環境における CKMS 設計での暗号鍵管理のガイダンスを作成することは今後の課題である。

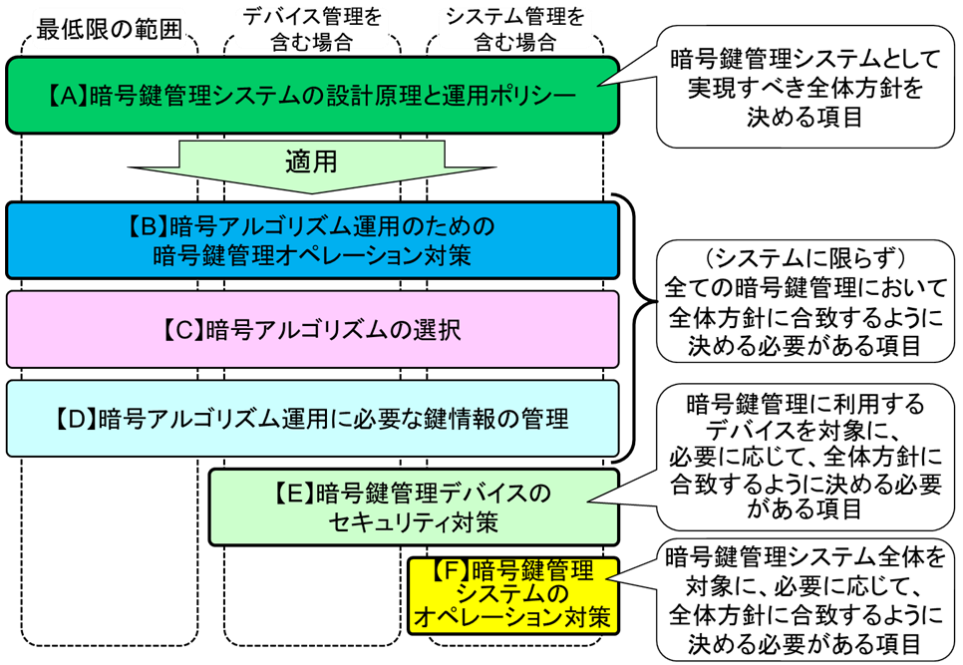
資料４－３

暗号鍵管理ガイダンスPart２
概要

1章 はじめに（資料4-4のP.4-P.8）

1.1 位置づけ

- 「暗号鍵管理システム設計指針（基本編）」は、あらゆるケースにおける暗号鍵管理システム（以下CKMS）を設計・構築・運用する際に考慮すべき事項一覧を提供する包括的な設計指針
- 「暗号鍵管理ガイダンス」は、「暗号鍵管理システム設計指針（基本編）」で記載が求められる項目について検討する際の有用な副読本となることを目的に作成
- ガイダンスはPart1（2023年5月発行）とPart2（本書）の2部構成
- Part2では【A】、【E】、【F】に該当する項目に関して、項目の概説及びその記載例を提供
- 【A】は全体方針であるが、【E】と【F】は広義の意味での検討項目



1.2 想定読者

- 主としてCKMS設計者（暗号鍵管理システム設計指針（基本編）での想定読者と同様）

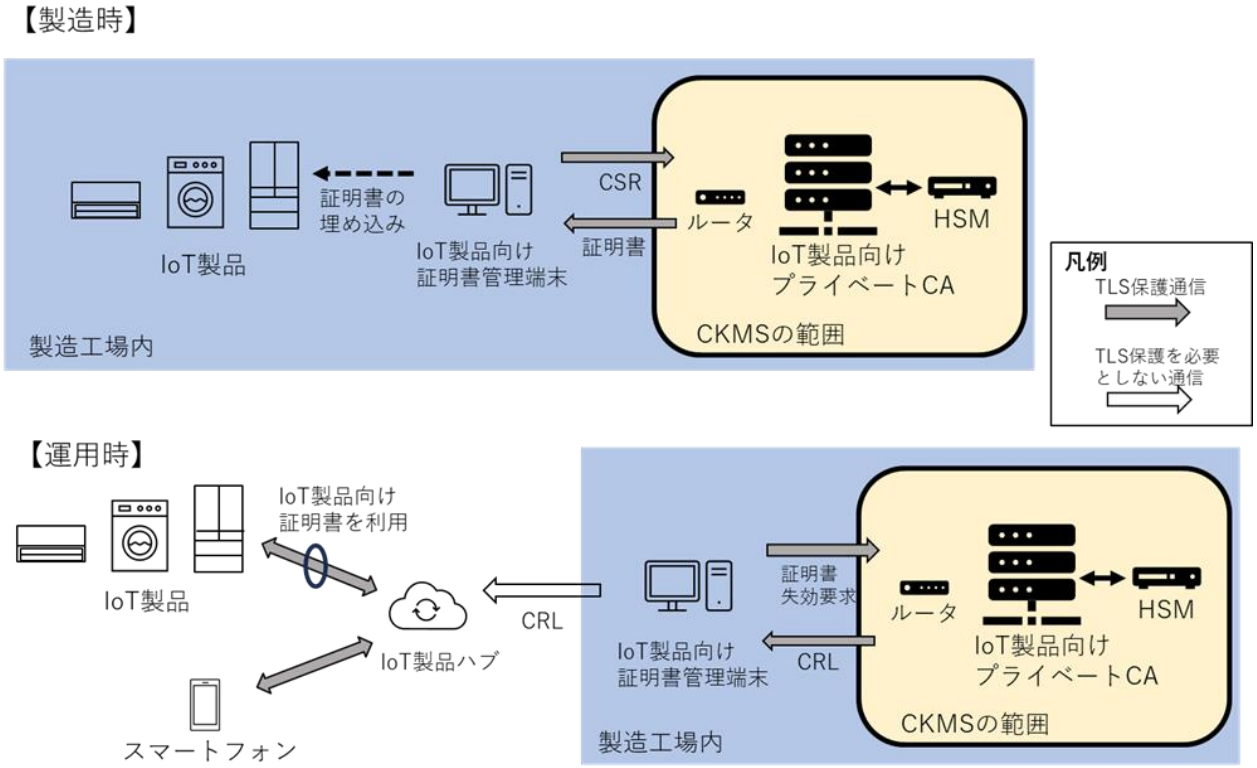
1.3 構成

- 各章での検討項目についての解説・考慮点を示す
- 理解を助けるため、簡単なシステム（トイモデル）を設定。トイモデルで設定された構成や運用条件などを踏まえて各章の検討項目における記載例を示す
- ここでのトイモデルの構成や運用条件は、これらの内容と各々の検討項目における記載例との対応関係が“理解しやすくなる”ように設けたものであり、これらの内容を“推奨しているわけではない”ことに十分に注意

1.4 トイモデル（資料4-4のP.8-P.10）

- トイモデルはIoT製品向けのプライベートCAシステム（下図）を設定。本ガイダンスPart2の各章で同一のトイモデルを扱う
- CKMSの範囲をCAサーバ、HSM（Hardware Security Module）（※）、ルータまでとする
- プライベートCAシステムはIoT製品（家電想定）向けに公開鍵証明書の発行及び証明書の失効管理に使われるCRLの発行を行う
- IoT製品は運用時にインターネット接続され、スマートフォン内専用アプリからIoT製品ハブ経由で状態の監視や制御が行われる。IoT製品は、製品向けに発行された証明書を利用してIoT製品ハブとの通信を確立し、スマートフォン内の専用アプリもIoT製品ハブと別途通信を確立する。これら2つの通信をIoT製品ハブが仲介することにより、IoT製品とスマートフォン間の保護された通信が実現される

※HSMを利用しないCAシステムを構築することも可能である。その場合は、HSMに代わる暗号鍵の保護に関する様々な対応が求められるため、本ガイダンスではHSMを利用するトイモデルとした



2章　暗号鍵管理システム（CKMS）の設計原理と運用ポリシー

CKMS設計において実現すべき全体方針を定める

- 2.1 CKMSセキュリティポリシー★
- 2.2 情報管理ポリシー等からの要求事項
- 2.3 ドメインのセキュリティポリシー

} CKMSのセキュリティポリシー（CKMSセキュリティポリシー、他の関連するセキュリティポリシー）

- 2.4 CKMSにおける役割と責任★
- 2.5 CKMSの構築目標及び実現目標
- 2.6 標準・規制に対する適合性

} CKMSの概要設計（エンティティと権限の定義、構築目標、関連する標準・規制）

- 2.7 将来的な移行対策の必要性★

} 将来の移行対策（暗号アルゴリズムや鍵管理デバイスの移行、技術の進歩に起因する課題評価）

★：後続のシートでピックアップ

2.1　CKMSセキュリティポリシー（資料4-4のP.12-P.18）

① CKMSの設計にあたって、CKMSセキュリティポリシーを作成しなければならない。
② CKMSセキュリティポリシーは、他のセキュリティポリシーや組織の様々なポリシーに依存することがあるので、それらを意識しなければならない。
③ CKMSセキュリティポリシーがCKMS内に電子的に保管され自動的に処理される場合には正しい処理が行われるように注意をしなければならない。

解説・考慮点

① セキュリティポリシーは、当該システムを運用する主体により、その主体の事情に合わせて策定されるものであり、どの程度の粒度で規定するかは運用主体が決定するものとなる。セキュリティポリシーの作成方法についての文書も少ないながら存在し、RFC 3647（Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework）はその1つとなる。また、NIST SP 800-57 Part 2には、組織が暗号鍵管理のポリシー（Key Management Policy）を作成する際に検討する内容とセキュリティポリシーを実現するための文書（Key Management Practice Statement）に書くべき内容のガイドが含まれている。

② CKMSを運用する組織にはCKMSセキュリティポリシー以外に様々なセキュリティに関わるポリシーが存在するのが一般的である。これらのポリシーは階層構造などの依存関係を持って、相互に矛盾なく規定される。CKMSセキュリティポリシーもこれらのポリシーとの関係を意識して定める必要がある。CKMSからみてより上位に位置づけられるポリシー（より汎用的なポリシー）もあれば、CKMSからみてより下位のポリシー（より具体的なポリシー）もある。上位のポリシーが変更された場合には、下位のポリシーの変更が必要となる可能性がある。

③ 例えば、X.509証明書においては「証明書ポリシー拡張」が存在し、拡張を閲覧することにより対象となる主体に対応するポリシーを確認することが可能となる。

トイモデルでの記載例

① 当該CKMSのセキュリティポリシーはRFC3647に基づくパブリックCAのCP（Certificate Policy）やCPS（Certification Practice Statement）を参考に作成した。

2.4　CKMSにおける役割と責任（資料4-4のP.27-P.31）

① CKMS参加者（責任者／管理者／運用者／ユーザ）には、それぞれの役割に応じて定義された特定の認可が必要であり、その役割の責任を果たすために、鍵情報を管理する一連の機能への必要なアクセスだけが提供されなければならない。

解説・考慮点

- 一般に役割の数が増えるほどより細やかなシステム運用が可能となる一方、オペレーションコストは増加する。CKMS設計者は、扱う情報の重要性とオペレーションコストのバランスを考慮しつつ、SP800-130に記載の11種の役割に代表される、システム運用に必要な役割を決定する。
- 役割を決める際の基本的な考え方は、その役割に求められる職務をもとに責任を規定した上で、その職務を遂行するために必要な最小限の権限を与えることである。

トイモデルでの記載例

- 責任者（システムオーソリティ）、管理者（システム管理者及び暗号責任者）、利用者（CKMSユーザ）、監査者（監査責任者）の役割が存在する。
- 管理者は、CA署名鍵の生成、更新、破棄が可能である。また、HSM内部の鍵情報のバックアップ・アンド・リストアが可能である。ただし、これらのHSM内部の鍵情報の操作にはHSM管理者としての権限が必要となる。当該権限の操作はマルチパーティコントロールがされているため、管理者2名以上の権限が必要である。管理者はこれに加えて、CAサーバやルータのアクセスコントロールの設定も可能である。
- 利用者は、CA署名鍵を使用した証明書発行、CA署名鍵を使用した証明書の失効処理の要求送出が可能である。
- 責任者および監査者は、操作ログ、運用ログの取得と閲覧が可能である。

2.7　将来的な移行対策の必要性（資料4-4のP.45-P.51）

① 使用中の暗号アルゴリズムは、必要なときに拡張又は置き換えができるように実装することを検討しておかなければならない。
② 技術の進歩に起因する潜在的な脅威についても考慮しておかなければならない。（暗号アルゴリズムに対する攻撃，鍵確立プロトコルに対する攻撃，デバイスに対する攻撃，量子コンピュータ）

解説・考慮点

① CKMS設計はそれによる保護の対象となるシステムやそのデータのライフタイムをカバーすべく、十分なセキュリティライフタイムを備えた暗号アルゴリズムを採用することとなる。採用した暗号アルゴリズムのセキュリティライフタイムを超えてCKMSサービスを提供する必要がある場合には、鍵長の変更や暗号アルゴリズムの変更が要求される。鍵長や暗号アルゴリズムの変更は、予め設定していた鍵長や暗号アルゴリズムのライフタイムを超える場合の他に、運用中の暗号アルゴリズムが予期せぬ危殆化をした結果、当初の設計よりも早く鍵長や暗号アルゴリズムの変更が必要となる場合もある。

② 長期の利用が想定されるCKMSでは、採用した暗号技術に対して新たな攻撃が発見される等の理由により、当初想定していた以上のペースでセキュリティ強度の低下が生じる可能性がある。そのため、長期の利用が想定されるCKMSでは、関連し得る技術の進歩を常に監視すると共に、潜在的な脅威への備えをしておく必要がある。具体的には、上記に挙げる4つの脅威を例とする潜在的な脅威が現実となった場合の影響評価等を予め実施することを推奨する。

トイモデルでの記載例

① 本CKMS設計において、CKMS及び各サブモジュールは極力シンプルに作られており、外部アプリケーションとの依存関係も極めて少ない。これは、暗号の置き換えが必要な場合において、CKMS全体またはHSMを含むサブモジュール単位での置き換えを想定しているためである。暗号解読可能な量子コンピュータ（CRQC：Cryptographically Relevant Quantum Computer）の実現可能性が高まった場合には、耐量子計算機暗号アルゴリズム（PQC：Post-Quantum Cryptography）をサポートしたHSM及びそれを含むCKMSへの置き換えを検討する。

② 当該CKMSの設計時のライフタイムは10年である。この期間で暗号解読可能な量子コンピュータが実現されることは、現在の技術水準、及び、これまでの量子コンピュータの発展状況に鑑みると、非常に困難と考えられる（本書を執筆した2025年時点の状況）。また、仮にそのような量子コンピュータの実現を仮定した場合でも、本IoT製品で保護する通信内容はリアルタイムで意味を持つ情報であり、過去の通信データを保存しておき後で解読する「ハーベスト攻撃」特有の脅威は小さい。

3章　暗号鍵管理デバイスへのセキュリティ対策

暗号鍵管理デバイス（暗号モジュール）に対する検討項目を定める

- 3.1 鍵情報へのアクセスコントロール
  - アクセスコントロールシステム★
  - 暗号モジュール
  - 人間による入力のコントロール
  - マルチパーティコントロール

} 鍵情報へのアクセスコントロール及び暗号モジュールによる保護

- 3.2 セキュリティ評価・試験
  - 機能テスト、セキュリティテスト、環境テスト、セルフチェックテスト、第三者テスト

} CKMSシステム及びデバイスに対するセキュリティ評価・試験

- 3.3 暗号モジュール障害時のBCP対策

} 障害発生に対する検知・回復

★：後続のシートでピックアップ

3.1　鍵情報へのアクセスコントロール（資料4-4のP.52-P.56）

• 3.1.1　　アクセスコントロールシステム

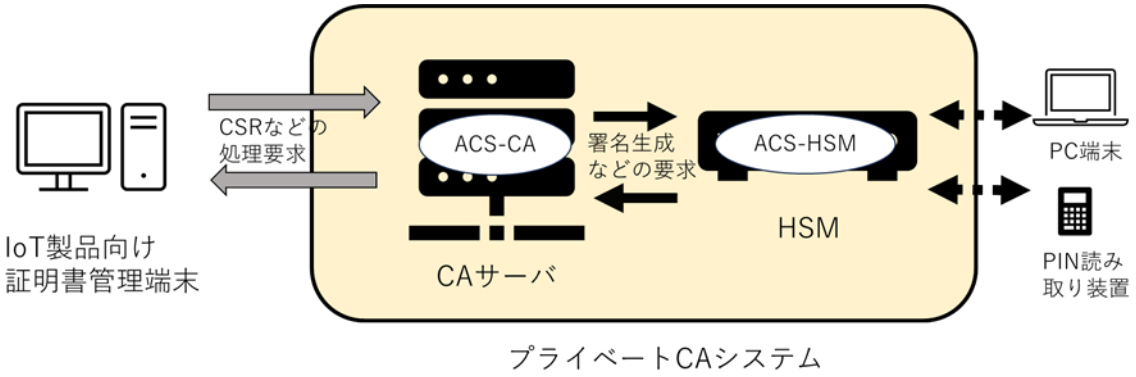
① アクセスコントロールへの要求事項を決めなければならない。

解説・考慮点

- CKMSのセキュリティは、鍵情報の管理機能が適切に設定され、認可されたエンティティからの要求のみに対応して鍵情報を利用した各種処理が実行されることによって担保される。アクセスコントロールシステム（ACS）はエンティティの認証や認可された処理の実行許可を与える機能モジュールであり、本節はCKMSが備えるACSを定義することを求めている。

トイモデルでの記載例

- 本プライベートCAシステムにおけるACSはCAサーバとHSMの2つ（ACS-CAとACS-HSM）がある。ACSのトポロジーは下図を参照。



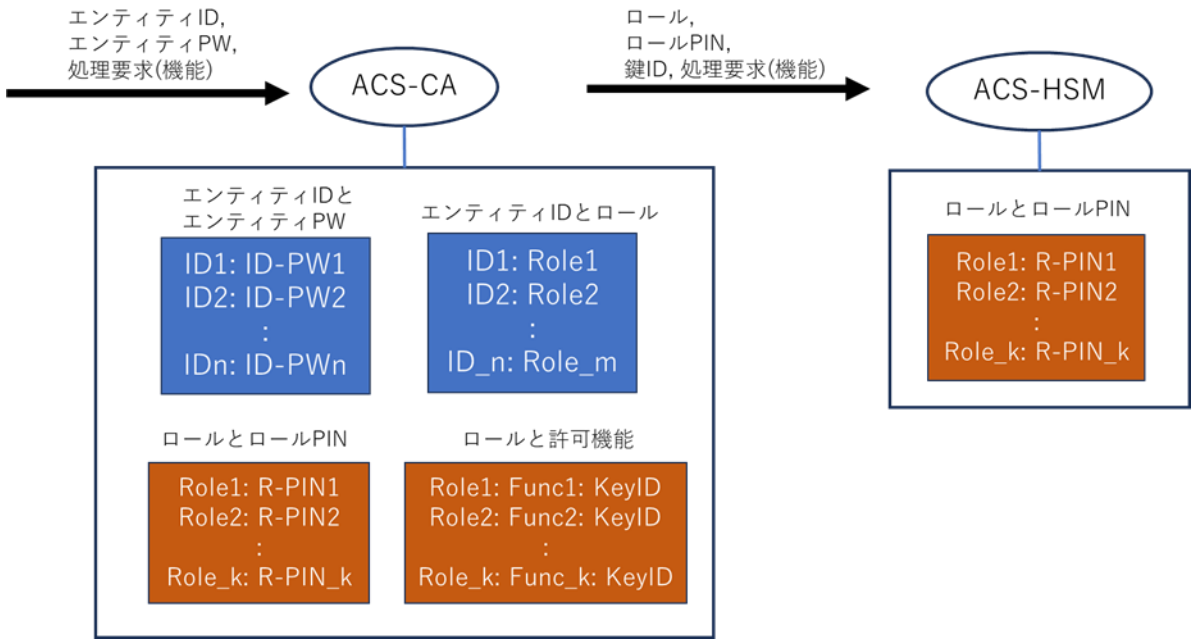
3.1　鍵情報へのアクセスコントロール（続き）（資料4-4のP.52-P.56）

• 3.1.1　　アクセスコントロールシステム

① アクセスコントロールへの要求事項を決めなければならない。

トイモデルでの記載例

- CAサーバ内のACS-CAが要求を送出した利用者を識別・認証し、認証結果に応じて利用者に割り当てられた役割（ロール）の実行許可を与える。CAサーバはHSMに対して処理の依頼を利用者に代わって送出し、HSMから受け取った処理結果を利用者に返す。ここで、HSM内のACS-HSMはCAサーバのロールを認証して処理の実行を許可する。ACSに関わる処理フローは下図を参照。
- 一方、HSM管理者の識別・認証は、CAサーバ内のACS-CAを介さずにHSM内のACS-HSMによって行われる。



4章　暗号鍵管理システムのオペレーション対策

暗号鍵管理システム全体に対する多層防御などの検討項目を定める

- 4.1 CKMSへのアクセスコントロール
  - 物理セキュリティコントロール★
  - コンピュータシステムセキュリティコントロール
  - ネットワークセキュリティコントロール
- 4.2 システム保証
  - 構成管理、配送、開発及びメンテナンス環境、欠陥修正能力

→ CKMS全体に対する包括的なセキュリティ対策

- 4.3 セキュリティアセスメント
  - 完全アセスメント、定期的レビュー、追加アセスメント、メンテナンス

→ CKMS全体のセキュリティアセスメント

- 4.4 CKMSへのアクセスコントール危殆化時のBCP対策
- 4.5 CKMS設備への障害・災害発生時のBCP対策

→ CKMS全体への危殆化・障害・災害発生に対する検知・回復

★：後続のシートでピックアップ

4.1　CKMSへのアクセスコントロール（資料4-4のP.77-P.80）

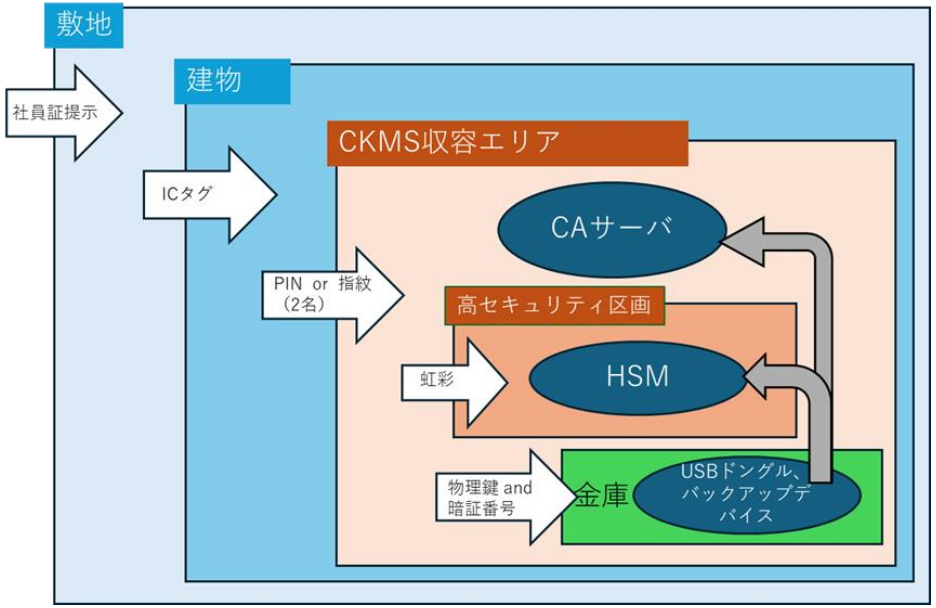
• 4.1.1　物理セキュリティコントロール

① CKMSコンポーネント及びデバイスに対する物理セキュリティの方法を決めなければならない。

解説・考慮点

- CKMSにおける物理的セキュリティ保護メカニズムを整理することを要求している。CKMSを構成するデバイスやコンポーネントの盗難やすり替え、物理的攻撃による改ざん、物理的攻撃による内部の機微な情報へのアクセスなどを防ぐために物理的保護が実施される。暗号モジュール自体が物理的保護メカニズムを備えている場合もあるが、多層防御としてCKMS全体を収容する施設（ファシリティ）としての物理的保護メカニズムも構築される場面が多い。
- 一般に施設レベルの物理的セキュリティ保護メカニズムとして、複数の段階の保護が設けられる。ある段階の物理的保護を解除するための認証情報や物理的鍵により、複数の段階の物理的保護を解除できることがないように保護メカニズムを構築することが重要である。

トイモデルでの記載例



Appendix. 参考資料一覧 （資料4-4のP.118-P.120）

■ **1章 はじめに**

- 「政府機関のサイバーセキュリティ対策のための統一基準（令和5年度版）」、NISC
- 「CRYPTREC暗号リスト（電子政府推奨暗号リスト）」、CRYPTREC
- 「暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準」、CRYPTREC
- 「暗号鍵設定ガイダンス」、CRYPTREC
- 「暗号鍵管理システム設計指針（基本編）」、CRYPTREC
- 「暗号鍵管理ガイダンスPart 1（2023年5月発行）」、CRYPTREC
- NIST SP 800-130（A Framework for Designing Cryptographic Key Management Systems）

■ **2章 暗号鍵管理システムの設計原理と運用ポリシー**

- RFC 3647（Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework）
- NIST SP 800-57 Part 2（Recommendation for Key Management: Part 2 – Best Practices for Key Management Organizations）
- 政府認証基盤GPKI（Government Public Key Infrastructure）
- PKCS #11（Cryptographic Token Interface Base Specification）
- PKCS #10（Certification Request Syntax Specification）
- RFC 5280（Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile）
- FIPS 140-2/-3
- ISO/IEC 15408
- CMVP認証
- 政府情報システムのためのセキュリティ評価制度（クラウドサービスの評価認証制度） ISMAP
- セキュリティ要件適合評価及びラベリング制度（JC-STAR）
- サイバーレジリエンス法、欧州
- サイバーセキュリティ法、中国
- GDPR（General Data Protection Regulation）、欧州
- 電子署名法
- 「暗号強度要件（アルゴリズム及び鍵長）設定基準」、CRYPTREC
- 「注意喚起情報」、CRYPTREC

Appendix. 参考資料一覧（続き） （資料4-4のP.118-P.120）

■ **3章 暗号鍵管理デバイスへのセキュリティ対策**
- FIPS 140-2/-3
- ISO/IEC 15408
- PKCS #11（Cryptographic Token Interface Base Specification）
- Shamirの秘密分散
- CMVP認証、CAVP認証

■ **4章 暗号鍵管理システムのオペレーション対策**
- 「データセンター セキュリティ ガイドブック」、日本データセンター協会
- 「金融機関等コンピュータシステムの安全対策基準・解説書」、金融情報システムセンター
- FIPS 140-2/-3
- CMVP認証
- 「セキュリティ設定共通化手順SCAP（Security Content Automation Protocol）概説」、IPA
- NIST SP 800-207（Zero Trust Architecture）
- 「脆弱性対処に向けた製品開発者向けガイド」、IPA
- 「ソフトウェア管理に向けたSBOMの導入に関する手引」、経済産業省
- CISSP (Certified Information Systems Security Professional)
- 情報処理安全確保支援士（Registered Information Security Specialist, RISS）
- CISM (Certified Information Security Manager)
- FIPS 199（Standards for Security Categorization of Federal Information and Information Systems）
- FIPS 200（Minimum Security Requirements for Federal Information and Information Systems）

https://www.cryptrec.go.jp/

資料４－４

暗号鍵管理ガイダンス

Part 2

（2025 年 3 月 10 日版）

独立行政法人情報処理推進機構

国立研究開発法人情報通信研究機構

目次

【修正履歴】

修正日	修正内容
2025.x.x	第 1 版公開

1 はじめに

1.1 位置づけ

企業や個人の管理する情報を保護するために暗号アルゴリズムが広く利用されている。各暗号アルゴリズムは、それぞれの情報が必要とする機密性、完全性、認証を提供する目的で利用される。

デジタル庁と総務省、経済産業省は、暗号技術に関する有識者で構成される CRYPTREC 活動を通して、電子政府で利用される暗号技術の評価を行っており、2023 年 3 月に「電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）」を改定した。CRYPTREC 暗号リストは、安全性、実装性能及び市場における利用実績を踏まえ、「電子政府推奨暗号リスト」、「推奨候補暗号リスト」及び「運用監視暗号リスト」で構成される。

CRYPTREC 暗号リスト（電子政府推奨暗号リスト）:
https://www.cryptrec.go.jp/list.html

実際、「政府機関のサイバーセキュリティ対策のための統一基準（令和 5 年度版）[1]」（令和 5 年 7 月 4 日、サイバーセキュリティ戦略本部。以下、「統一基準」という）では、政府機関における情報システムの調達及び利用において、図 1-1 のとおり、CRYPTREC 暗号リストのうち「電子政府推奨暗号リスト」に記載された暗号アルゴリズムを原則的に利用するように記載されている。このように、セキュアな暗号アルゴリズムの選択に関しては電子政府推奨暗号リストを活用する等により、比較的容易に満たすことができる。

しかしながら、実際のシステムがセキュアに動作し続けるためには暗号アルゴリズム自体がセキュアであるだけでは不十分である。統一基準でも暗号鍵の管理手順を定めることになっているように、データが保護される期間中、その暗号アルゴリズムが使用する暗号鍵もセキュアに管理されている必要がある。もし、暗号鍵がセキュアに管理されていなければ、管理が不十分な点を悪用した何らかの手段で暗号鍵が漏えいする可能性があり、その漏えいした暗号鍵を使ってシステムへの侵入、機密データの窃取や改ざん、なりすましなどが行われる。

一般に、暗号鍵管理の脆弱性を突く攻撃方法のほうが、セキュアな暗号アルゴリズム自体を解読するよりもはるかに容易な攻撃方法である。また、漏えいまでは至らなくても、暗号鍵にデータ不整合等が発生すればシステムエラーの原因となり、業務が停止するなどの悪影響が発生する場合もある。実際、セキュアな暗号アルゴリズムを利用していても、不十分な暗号鍵管理が原因となって、数多くのインシデントが発生している。

[1] 内閣サイバーセキュリティセンター（NISC）, https://www.nisc.go.jp/pdf/policy/general/kijyunr5.pdf

7.1.5 暗号・電子署名

遵守事項

(1) 暗号化機能・電子署名機能の導入

(a) 情報システムセキュリティ責任者は、情報システムで取り扱う情報の漏えいや改ざん等を防ぐため、以下の全ての措置を講ずること。

(ア) 要機密情報を取り扱う情報システムについては、暗号化を行う機能の必要性 の有無を検討し、必要があると認めたときは、当該機能を設けること。

(イ) 要保全情報を取り扱う情報システムについては、電子署名の付与及び検証を 行う機能を設ける必要性の有無を検討し、必要があると認めたときは、当該機 能を設けること。

(b) 情報システムセキュリティ責任者は、**暗号技術検討会及び関連委員会（CRYPTREC）により安全性及び実装性能が確認された「電子政府推奨暗号リスト」に基づき**、情報システムで使用する暗号及び電子署名のアルゴリズム及び鍵長並びにそれらを利用した安全なプロトコルを定めること。また、その運用方法について実施手順を定めること。

(c) 情報システムセキュリティ責任者は、機関等における暗号化及び電子署名のアルゴリズム、鍵長及び運用方法に、電子署名を行うに当たり、電子署名の目的に合致し、かつ適用可能な公的な公開鍵基盤が存在する場合はそれを使用するなど、目的に応じた適切な公開鍵基盤を使用するように定めること。

図 1-1　政府機関のサイバーセキュリティ対策のための統一基準（抜粋）

さらに、暗号鍵管理はうまく利用すると、大規模なデータ管理をセキュアに実現することも可能になる。例えば、クラウドサービスなど、外部の第三者にデータを預ける場合であっても、それらのデータを暗号化し、そのときの暗号鍵管理を利用者側が実施することで、クラウドサービス事業者に対しても機密性を維持できる。また、データセンターや大規模な記録メディアなどに保存されたデータで、物理的な破砕によるデータの完全削除を実現することが困難なケースでは、暗号鍵の破壊によって当該鍵で暗号化されたデータを事実上復号できなくすることでそれらのデータが完全に削除されたとみなす暗号化消去（Cryptographic Erase）といった方法を実現することもできる。

このような背景のもと、CRYPTREC では暗号鍵管理に関するガイドライン／ガイダンスを作成している。

- 暗号鍵管理システム設計指針（基本編）[2]
- 暗号鍵設定ガイダンス[3]

[2] https://www.cryptrec.go.jp/op_guidelines.html
[3] https://www.cryptrec.go.jp/op_guidelines.html

- 暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準[4]

このうち、「暗号鍵管理システム設計指針（基本編）」（以下、「設計指針」と呼ぶ）は、暗号鍵管理システム（以下、「CKMS (Cryptographic Key Management System)」という）を設計・構築・運用する際に参考すべきドキュメントとして作成されたものであり、「暗号鍵管理についての技術的内容」について解説している。具体的には、あらゆるユースケースにおける暗号鍵管理を安全に行うための構築・運用・役割・責任等に関する対応方針として考慮すべき事項一覧を提供し、CKMS 設計時に考慮すべきトピックス及び設計書等に明示的に記載する要求事項を示している。これは、CKMS の包括的な設計指針であり、CKMS 設計時に考慮すべきトピックス及び設計書等に明示的に記載する要求事項を列挙した NIST SP 800-130「A Framework for Designing Cryptographic Key Management Systems」をベースに作成されている。

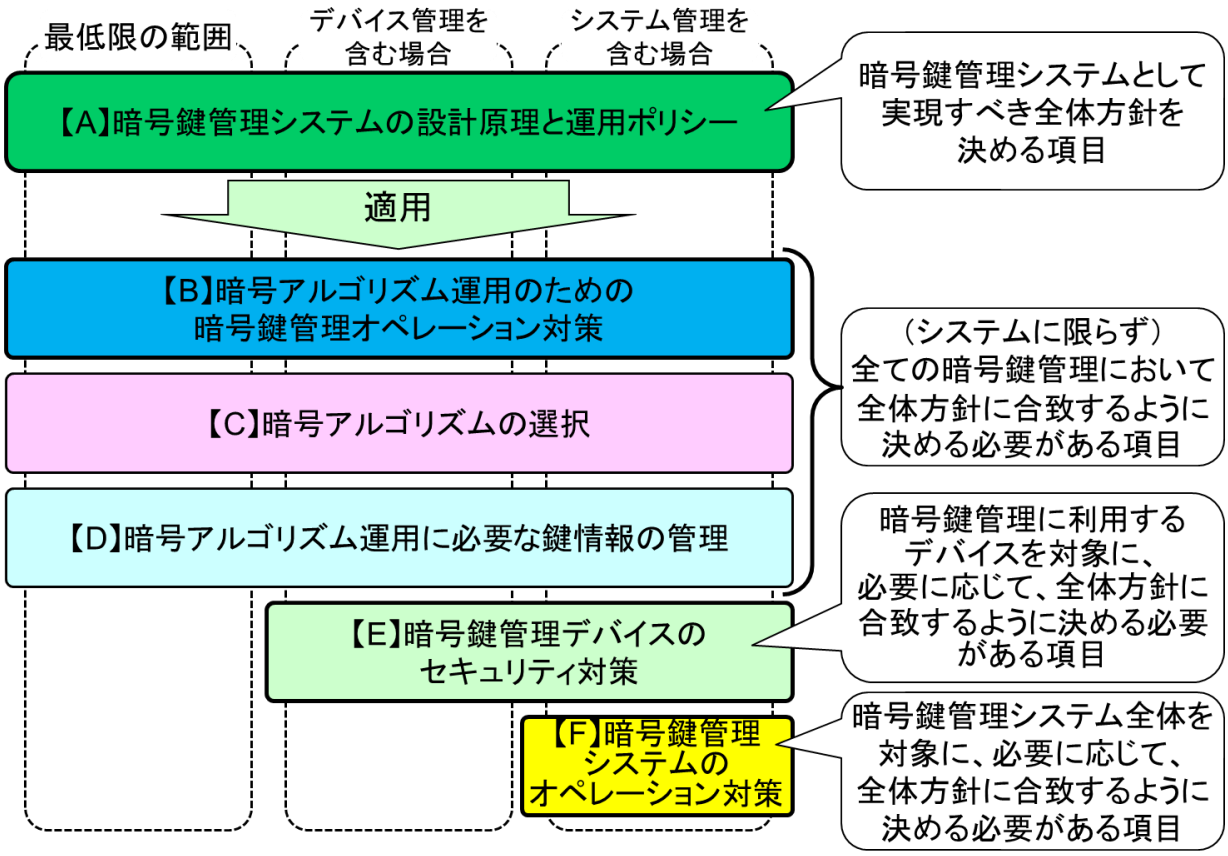


図 1-2　暗号鍵管理における目的別分類関係（「暗号鍵管理システム設計指針」より）

本書である「暗号鍵管理ガイダンス」は、「設計指針」で記載が求められる項目について検討する際の有用な副読本となることを目的として書かれたものである。本ガイダンスは Part 1 と Part 2 の 2 部構成となっている。

[4] https://www.cryptrec.go.jp/list.html

ガイダンス Part 1（2023 年 5 月発行）では、図 1-2 において、CKMS の利用環境に関わらず検討する必要がある項目のうちの【B】、【C】、【D】に該当する項目に関して、項目の概説及びその記載例を提供している。これらの項目は「狭義」の意味での暗号鍵管理に相当するものである。CKMS を設計する場合だけでなく、暗号アルゴリズムを使ったアプリケーション等を利用する場合なども含めて、全ての暗号鍵管理に対して検討が必要となる項目であることに留意されたい。

ガイダンス Part 2（本書）では、図 1-2 における【A】、【E】、【F】に該当する項目に関して、項目の概説及びその記載例を提供している。【A】は CKMS の利用環境に関わらず検討する必要がある項目のうち、CKMS の全体方針を定める項目である。【E】は CKMS に利用するデバイス管理を含む場合に検討すべき項目であり、【F】は CKMS のシステム管理を含む場合に検討すべき項目である。【E】や【F】までを含む場合、「広義」の意味での暗号鍵管理に相当する内容となる。

図 1-2 における【A】から【F】は、それぞれ順に「設計指針」の 4 章から 9 章に対応している。これらの「設計指針」の 4 章から 9 章についてはガイダンス Part 1 及び Part 2 においてより詳細を解説している[5]が、「設計指針」の 1 章から 3 章にイントロダクションとして記載されている内容については本ガイダンスに十分な記載をしていない。特に、「設計指針」の 2 章「暗号鍵管理の在り方」及び 3 章「本設計指針の活用方法」については、暗号鍵管理に責任を有するあらゆる担当者を想定読者とした内容であり、「設計指針」に基づいて CKMS を検討する際の基本的な事項を説明している。本ガイダンスと併せて「設計指針」の該当章を参照いただきたい。

上記のように、本ガイダンス Part 2 は、「設計指針」及びガイダンス Part 1 と併せて利用することを想定している。また、「暗号鍵管理システム設計指針（基本編）チェックリスト[6]」を利用する際には、以降に説明するトイモデルの記載例が参考となる。

1.2 想定読者

「設計指針」の 4 章以降に対する想定読者と同様であり、主として CKMS 設計者を想定読者としている。

1.3 構成

本ガイダンス Part 2 は、4 章で構成されており、章立ては以下のとおりである。

1 章は「はじめに」として、本ガイダンスの位置づけや想定読者を説明し、さらに本ガイダンスにおいて、各章の理解を助けるために設定した簡単なシステム（トイモデル）について説明する。トイモデルは、そこで設定された構成や運用条件などを踏まえて各章における項目に対する記載例を示すために導入したものである。

---

[5] 本書の 2 章以降において灰色枠内で囲った箇所は、「設計指針」該当部分の記載内容を転記したものである。

[6] IPA, https://www.ipa.go.jp/security/crypto/guideline/ckms.html

【トイモデルにおける注意】

ここでのトイモデルの構成や運用条件は、これらの内容と各々の項目における記載例との対応関係が“理解しやすくなる”ように設けたものであり、これらの内容を“推奨しているわけではない”ことに十分に注意されたい。

2 章は「暗号鍵管理システムの設計原理と運用ポリシー」における項目についての解説・考慮点を記載し、トイモデルでの記載例を示す。

同様に、3 章では「暗号鍵管理デバイスへのセキュリティ対策」における項目についての解説・考慮点を記載し、トイモデルでの記載例を示す。

最後に、4 章では「暗号鍵管理システムのオペレーション対策」における項目についての解説・考慮点を記載し、トイモデルでの記載例を示す。

なお、「設計指針」に述べているように、本ガイダンスにおいて各節のトピックスで対象とする Framework Requirements の目的を「①、②、…」として記載している。CKMS 設計者は、この目的及びそれに続く解説に照らし合わせて、個々のトピックスが今回設計する CKMS で検討する必要がある範囲であるかどうかの判断を行う。対象範囲と判断すれば Framework Requirement ごとにどのような対応をとるかを決定する。一方、対象範囲外と判断すればそのように判断した理由を明記したうえで当該 Framework Requirement は「対象外」として除外する。

さらに、対象範囲と判断した Framework Requirement であっても、ベンダから商用既製品として調達するデバイスやコンポーネントについては機能要件を定めることでよく、その機能の詳細に関わるものは、CKMS 設計者や調達者がその仕組みまでを確認するのではなく、ベンダに情報提供を求めることでよい。ベンダからの情報提供が得られないものは対象外としてもよい。

1.4 トイモデル

本書における CKMS のトイモデルは、図 1-3 に示す、IoT 製品向けに公開鍵証明書を発行するプライベート CA システムとする。ここでの IoT 製品としては家電製品を想定する。

このシステムの CKMS の設計範囲は、図 1-3 のとおり、CA サーバ、HSM（Hardware Security Module）[7]、及びルータまでとする。プライベート CA の主要機能は、IoT 製品向け証明書の発行及び証明書の失効処理である。IoT 製品利用者にとって、証明書のトラストアンカーは本プライベート CA となる。

CKMS の提供機能に関わる IoT 製品や製造工場内の機器と動作の概要を説明する。CKMS 外部の IoT 製品向け証明書管理端末は、プライベート CA に対して CSR（Certificate Signing Request）による IoT 製品向け証明書の発行の送出、及び証明書の失効要求の送出を行う。IoT 製

[7] HSM を利用しない CA システムを構築することも可能である。その場合は、HSM に代わる暗号鍵の保護に関する様々な対応が求められるため、本ガイダンスでは HSM を利用するトイモデルとした。

品向け証明書管理端末とプライベート CA は同じ IoT 製品の製造工場内に存在し、工場の計算機ネットワークに接続されている[8]。

IoT 製品の製造時に、プライベート CA は IoT 製品向け証明書管理端末からの要求を元に証明書を発行する。ここで、IoT 製品の ID 発番及びプライベート鍵生成を伴う ID 管理がプライベート CA 外部の IoT 製造環境で行われ、証明書の IoT 製品への埋め込みは製造工場内で行われる。

市場出荷後の IoT 製品は、当該製品の利用者の設定によってインターネット接続され、利用者のスマートフォン内の専用アプリから IoT 製品ハブを経由して当該製品の動作状態のセンシング及び設定や動作に関わる制御が行われる。IoT 製品は、製品向けに発行された証明書を利用して IoT 製品ハブとの通信を確立し、スマートフォン内の専用アプリも IoT 製品ハブと別途通信を確立する。これら 2 つの通信を IoT 製品ハブが仲介することにより、IoT 製品とスマートフォン間の保護された通信が実現される[9]。

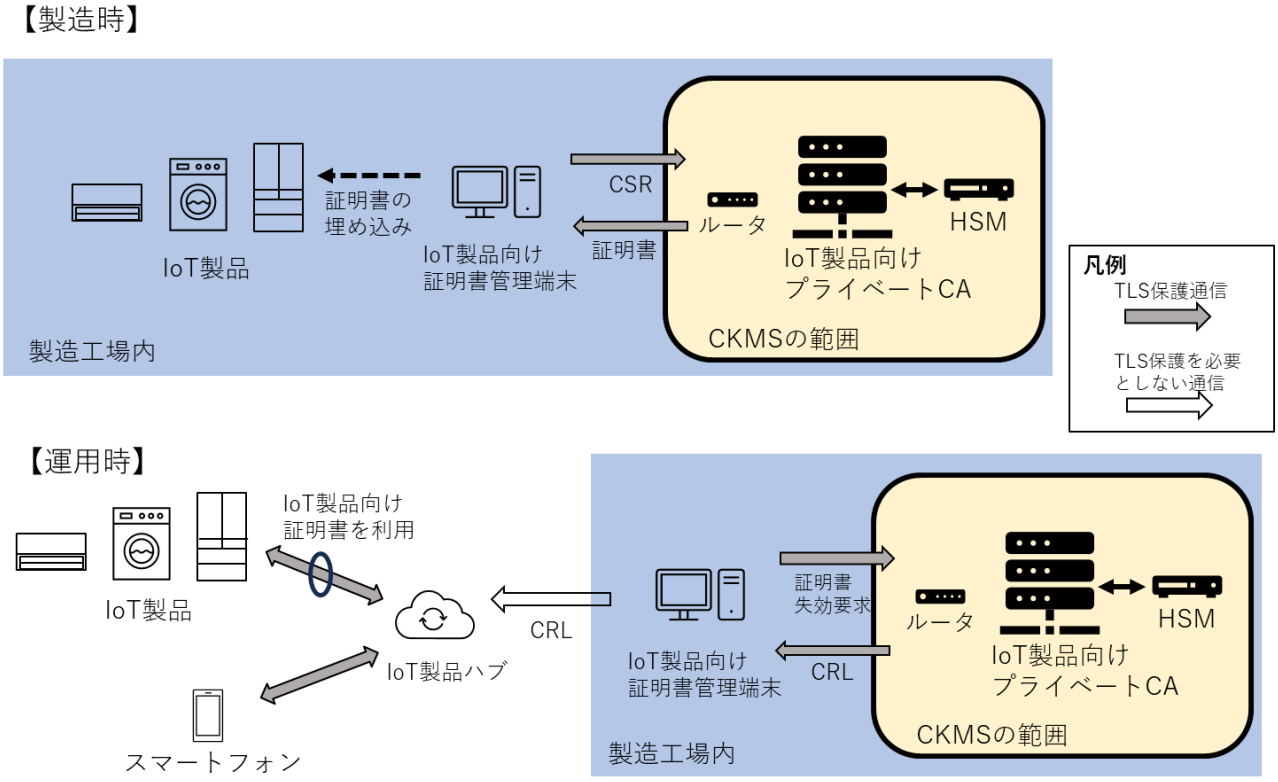


図 1-3 トイモデル（プライベート CA）の概要

出荷後の当該製品の運用中にセキュリティに関わる重大な事故が発生した場合や製品リコール時などに証明書の失効処理を行う場合がある。失効処理の対象となる IoT 製品の ID 管理も製造

[8] ASP（Application Service Provider）やクラウドサービス（SaaS）として CA 機能が提供される場合もあるが、本トイモデルではオンプレミスにプライベート CA を構築する想定であることに注意されたい。

[9] これらの通信の確立においては、IoT 製品及びスマートフォンアプリが IoT 製品ハブを認証するための情報や、IoT 製品ハブがスマートフォンの利用者を認証するための情報を用いることになるが、それらの情報を管理する機能は今回の CKMS 機能に含まれない。

工場内の環境で行われ、プライベート CA は IoT 製品向け証明書管理端末からの要求を元に CRL（Certificate Revocation List）を発行する。CRL は IoT 製品向け証明書管理端末から IoT 製品ハブに送られ、利用者は IoT 製品ハブにおいて証明書の失効状況を確認できる。

IoT 製品向け証明書管理端末とプライベート CA 間の証明書発行及び証明書失効に関わる通信は TLS を利用し、CSR や失効要求情報の改ざん防止及び送信元の認証が実施される。

以上から、本 CKMS が扱う鍵情報として、CA 証明書生成用の署名鍵・検証鍵、IoT 製品向けに生成された証明書が該当する。また、プライベート CA が IoT 製品向け証明書管理端末との通信を行う際の TLS 通信に用いられるサーバ証明書及びプライベート鍵も本 CKMS が扱う鍵情報に含まれる。ただし、本ガイダンスにおけるトイモデル記載例では、HSM で管理されている CA 証明書生成用の署名鍵を利用して発行される、IoT 製品向け証明書に関わる部分に注目している。プライベート CA と IoT 製品向け証明書管理端末との間の TLS 通信に必要となるサーバ証明書及びプライベート鍵については、当該プライベート鍵の保護に関わる事項の記載を省略している。

1.5 検討体制

本ガイダンス Part 2 は、2023 年度及び 2024 年度 CRYPTREC 暗号鍵管理ガイダンス WG において作成された。

表 1-1 暗号鍵管理ガイダンス WG の構成（2025 年 3 月時点）

主査	上原　哲太郎	立命館大学　情報理工学部　情報理工学科　教授
委員	泉　雅明	シスコシステムズ合同会社 東日本公共・法人システムズエンジニアリング ソリューションズエンジニアリング第 1 ソリューションズエンジニア
委員	漆嶌　賢二	GMO グローバルサイン株式会社　事業企画部　フェロー
委員	垣内　由梨香	Microsoft Corporation　セキュリティレスポンスチーム セキュリティプログラムマネージャー
委員	菅野　哲	GMO サイバーセキュリティ by イエラエ株式会社 常務取締役 CTO of Development
委員	菊池　浩明	明治大学　総合数理学部　先端メディアサイエンス学科　教授
委員	小林　浩二	パナソニックオートモーティブシステムズ株式会社 開発本部　プラットフォーム開発センター セキュリティ開発部　開発 2 課　2 係　係長
委員	須賀　祐治	株式会社インターネットイニシアティブ セキュリティ本部　セキュリティ情報統括室　シニアエンジニア

委員	舟木　康浩	タレス DIS CPL ジャパン株式会社 クラウドプロテクション&ライセンシング データプロテクション事業本部　セールスエンジニアマネージャ
委員	程吉　英仁	株式会社 NTT データ ソリューション事業本部　セキュリティ&ネットワーク事業部 サイバーセキュリティ統括部　課長代理
委員	満塩　尚史	順天堂大学　健康データサイエンス学部　健康データサイエンス学科 准教授

2 暗号鍵管理システム（CKMS）の設計原理と運用ポリシー

本章の目的・趣旨

本章は、「設計指針」の 4 章に記載されている要求事項（各節での灰色枠内で示している内容）について解説したものである。

CKMS として実現すべき全体方針を取り決める項目（項目 A.01～A.69）を集めている。ここには、主に以下のような項目を含んでいる。

- CKMS をどのような方針（ポリシー）で運用するのか。そのために、どういった機能を用意しなければならないのか
- CKMS 参加者（責任者／管理者／運用者／ユーザ）が誰でどういった権限を有しているのか
- CKMS の構築環境や実現目標はどういったものか
- 適合しなければならない法規制や標準化等があるのか。あるならどういったものか
- 将来的な移行対策を準備しておく必要があるか。あるならどういった準備をするか

ここでの項目の検討結果が、他の章における項目での具体的な技術的選択や精緻化などに当たっての条件として適用される。

2.1 CKMS セキュリティポリシー

解説・考慮点

本節は、SP 800-130 の 4.3 節、4.4 節、4.5 節に記載されている事項について解説したものである。

CKMS は、CKMS を使用しているそれぞれの組織の目標をサポートするやり方で設計されなければならず、またそれぞれの組織が有するポリシー群とも整合させる必要がある。そのうちのいくつかのポリシーは CKMS の設計及び使用に影響を及ぼすため、まず CKMS 設計における設計原理と運用ポリシーを整理する必要がある。これは、CKMS セキュリティポリシーとして定義される。

① CKMS の設計にあたって、CKMS セキュリティポリシーを作成しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.01	FR4.1	CKMS 設計は、実行するために設計した設定可能なオプションとサブポリシーを含む CKMS セキュリティポリシーを明記しなければならない。	4.3 節

A.02	FR4.2	CKMS 設計は、CKMS セキュリティポリシーが CKMS によってどのように実行されるのか（例えば、ポリシーが要求する保護を提供するために使用されるメカニズム）を明記しなければならない。	4.3 節

解説・考慮点

CKMS セキュリティポリシーは、情報管理ポリシー及び情報セキュリティポリシーに従ってデータを保護するために、CKMS がサポートしなければならないデータ、並びに鍵情報を保護するためのルールを規定するものである。
CKMS の設計にあたって、項目 A.01 は CKMS セキュリティポリシーを作成することを、A.02 はその CKMS セキュリティポリシーに明記すべき内容及びその実現・利用方法について明確化することを要求したものである。具体的には、以下のようなことが求められる。

- CKMS で使用される全ての鍵情報の機密性、完全性、可用性、及びソース認証（source authentication）を保護するためのルールを定める。
  - 鍵ライフサイクル全体にわたってカバーされなければならない。
  - CKMS が使用できる全ての暗号メカニズム及び暗号プロトコルの選択を含むこともある。
- 組織のより高位レベルのポリシー群（情報管理ポリシー及び情報セキュリティポリシー）と定めたルールが整合している必要がある。
- CKMS セキュリティポリシーに沿ってデータの保護を実行するために、いつどのようにセキュリティ機能が使用されるのかを文書化する。
- 教育・トレーニング等を通じて、各種ポリシーを役員・従業員が容易に理解して自らの役割及び責任を正しく実行できるように書かれるべきである。

なお、CKMS 設計において、CKMS セキュリティポリシーを適切にサポートしているか、又はサポートするように設定できるかどうかを保証・確認するのは、CKMS を使用する組織の責任である。

【参考】

- 情報管理ポリシーに明記すべき要件には、以下のようなものがある。
  a) 収集又は作成する情報、及び管理方法
  b) 情報を獲得及び利用するための高レベルの目標
  c) ポリシーに対する組織上の管理ルール及び責任
  d) 情報管理上の義務を実行するために要求される認可
  e) 認可されない開示（窃取）、改ざん、又は破壊に対して保護が必要な情報（保護対象の情報）のカテゴリ
  f) ポリシーを作成し、その実装と利用を管理するための権限を誰に与えるかのルール

- 情報セキュリティポリシーに明記すべき要件には、以下のようなものがある。

a) 機微と考えられる情報（保護対象の情報）のカテゴリ
b) 情報に関連するインパクトレベル
c) 情報に対する現時点で予測されている潜在的なリスク
d) 必要な保護を行うための方法
e) 情報を収集、保護及び配付するためのルール

- CKMS セキュリティポリシーに明記すべき要件には、以下のようなものがある。

a) ポリシーを適用する組織名称
b) ポリシーを承認／変更する権限を有する人（人物、役職、又は役割）
c) ポリシーに明記され、コントロールされる情報のインパクトレベル
d) 提供される主要なデータ及び暗号鍵／メタデータの保護処理（データ秘匿性、データ完全性、ソース認証）
e) サポートできるセキュリティ処理（例：個人の説明責任、個人のプライバシー、可用性、匿名性、連結不可能性、観測不可能性）
f) 暗号鍵及び関連付けられたメタデータに対する制限の影響及び取り扱い
g) 各々のインパクトレベル及び各々の保護サービスで利用されるアルゴリズム及び全ての関連パラメタ
h) 利用される各々の暗号アルゴリズムに対する鍵情報の期待される最大許容暗号鍵有効期間（この期間を超えて同一の鍵情報（暗号鍵やメタデータ）が利用され続けてはならない）
i) 暗号鍵及び関連付けられたメタデータによって保護される各々の情報インパクトレベルに対するユーザ／役割及びソース認証の受け入れ可能な方法
j) 各々の情報インパクトレベルに応じた鍵情報に対するバックアップ、アーカイブ及び復元要求
k) サポートされる役割
l) 各々のインパクトレベルに対する鍵情報に対するアクセスコントロール及び物理的セキュリティ要件
m) 鍵情報を復元する手段とルール
n) 機微データ、及び鍵情報を保護する際の利用される通信プロトコル

CKMS の設計にあたって、項目 A.01 は CKMS セキュリティポリシー（CKMS のセキュリティを確保するための指針）を明文化することを要求している。CKMS セキュリティポリシーは CKMS の設計前に定めておくことが好ましいが、仮に CKMS が暗黙的なセキュリティポリシーに基づいて設計されていた場合、そのセキュリティポリシーを後からでも明文化することを項目 A.01 は要求している。

セキュリティポリシーは、当該システムを運用する主体により、その主体の事情に合わせて策定されるものであり、どの程度の粒度で規定するかは運用主体が決定するものとなる。セキュリティポリシーの作成方法についての文書も少ないながら存在し、RFC 3647（Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework）はその１つとなる。もっとも、RFC 3647 は CA のセキュリティポリシーに関する一般的なフレームワ

ークを記載するものであり、必ずしもその記載に則ってセキュリティポリシーを作成する必要はない。例えば、自社向けの小規模な CKMS に対して、簡略化したセキュリティポリシーを作成することは許容される。また、NIST SP 800-57 Part 2 には、組織が暗号鍵管理のポリシー（Key Management Policy）を作成する際に検討する内容とセキュリティポリシーを実現するための文書（Key Management Practice Statement）に書くべき内容のガイドが含まれている。

項目 A.02 はその CKMS セキュリティポリシーに明記された内容が、どのように実現されるかを記載したものとなる。例えば、セキュリティポリシーに「FIPS 140-2 レベル 3 を満たす HSM を FIPS モードで利用すること」と記載されていた場合に、具体的にどのメカニズム（暗号アルゴリズムや認証手法等）を、どのように利用するかを記載する。例えば、暗号アルゴリズムの標準名や、処理内容を記した仕様書等が存在する場合はその仕様書を指定する。ただし、項目 A.02 は詳細な仕様までを求めるものではなく、セキュリティポリシーを実現するためのメカニズムを概要レベルで記載することで良い。

《トイモデルと記載例》

本節のトイモデルは 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.01	当該 CKMS のセキュリティポリシーは RFC3647 に基づくパブリック CA の CP（Certificate Policy）や CPS（Certification Practice Statement）を参考に作成した。 セキュリティポリシーは<URI>に保存されている。
A.02	本プライベート CA において証明書や CRL に付与する署名の鍵生成や署名処理は、管理区域内に設置された FIPS 140-2/-3 の認証を取得した HSM によって実行する。管理区域への入室は CKMS 管理者権限のあるメンバに制限する。証明書の発行処理や失効処理の要求送出は CKMS 利用者権限のあるメンバに制限する。（以下略） CKMS メカニズムの仕様書は<URI>に保存されている。

② CKMS セキュリティポリシーは、他のセキュリティポリシーや組織の様々なポリシーに依存することがあるので、それらを意識しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.03	FR4.4	CKMS 設計は、CKMS セキュリティポリシーをサポートする他の関連するセキュリティポリシーを明記しなければならない。	4.4 節
A.04	FR4.5	CKMS 設計は、CKMS 設計によってサポートされるポリシーと、その設計によってどのようにサポートされるのかの要約を明記しなければならない。	4.5 節

解説・考慮点

高位レベルのポリシー群（情報管理ポリシー及び情報セキュリティポリシー）以外にも、CKMS セキュリティポリシーをサポートする別のセキュリティポリシー（例：コンピュータセキュリティポリシー）があったり、CKMS セキュリティポリシー以外のセキュリティポリシー（例：CKMS モジュールセキュリティポリシー）が存在したりする可能性がある。

CKMS の設計にあたって、項目 A.03 は CKMS セキュリティポリシーをサポートする別のセキュリティポリシー（もしあれば）の情報について、A.04 は CKMS の適切でセキュアな運用を実行するために要求される CKMS セキュリティポリシー以外のセキュリティポリシー（もしあれば）の情報について明確化することを要求したものである。

上記①の解説・考慮点にも記載されているように、CKMS を運用する組織には CKMS セキュリティポリシー以外に様々なセキュリティに関わるポリシーが存在するのが一般的である。例えば、組織が取り扱う情報全般に対する情報管理ポリシー、組織内の情報セキュリティの維持管理に関わる情報セキュリティポリシー、さらに組織内のコンピュータシステムにおける全般的なセキュリティ対策などを定めたコンピュータセキュリティポリシーなどが該当する。これらのポリシーは階層構造などの依存関係を持って、相互に矛盾なく規定される。

CKMS セキュリティポリシーもこれらのポリシーとの関係を意識して定める必要がある。CKMS からみてより上位に位置づけられるポリシー（より汎用的なポリシー）もあれば、CKMS からみてより下位のポリシー（より具体的なポリシー）もある。上位のポリシーが変更された場合には、下位のポリシーの変更が必要となる可能性がある。

項目 A.03 及び A.04 はこうした CKMS 以外の関連するセキュリティポリシーを CKMS から見た依存関係を含めて一通り整理することを要求するものであり、項目 A.03 は CKMS セキュリティポリシーに依存する下位のセキュリティポリシーのリストを、項目 A.04 は関連する上位のセキュリティポリシーのリストを明らかにすることを要求している。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。本モデルでは、IoT 製品のセキュアな製造及びセキュアな運用などの管理手順を定めたポリシーが「IoT 製品製造管理及び運用管理ポリシー」として明文化されていることを想定している。同ポリシーにおける IoT 製品向け証明書の発行及び失効管理に関わるセキュリティポリシーを詳細化したものが当該 CKMS セキュリティポリシーとして位置づけられる。

一方、IoT 製品の製造時や運用時のセキュリティポリシーとプライベート CA のセキュリティポリシーとの間には、次のような依存関係があることに注意が必要である。プライベート CA のセキュリティポリシーに依存して IoT 製品のセキュアな製造や運用が担保されるが、逆に IoT 製品の製造や運用におけるセキュリティはプライベート CA のセキュリティには影響を及ぼさない。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.03	当該 CKMS セキュリティポリシーの下位に位置づけられるポリシーには以下のものがある。 ● HSM セキュリティポリシー
A.04	当該 CKMS セキュリティポリシーの上位に位置づけられるポリシーには以下のものがある。 ● 情報管理ポリシー ● 情報セキュリティポリシー ● コンピュータ及びネットワークセキュリティポリシー ● IoT 製品製造管理及び運用管理ポリシー

③ CKMS セキュリティポリシーが CKMS 内に電子的に保管され自動的に処理される場合には正しい処理が行われるように注意をしなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.05	FR4.3	CKMS 設計は、CKMS セキュリティポリシーのあらゆる自動化部分についてどのように曖昧さのない表形式又は形式言語（例えば XML、ASN.1）で表現されているのかを明記しなければならない。CKMS の自動化されたセキュリティシステム（例えば table driven 又は syntax-directed software mechanisms）がそれらを実行できるようにするためである。	4.3 節

解説・考慮点

CKMS セキュリティポリシーが自動処理される場合、その内容が正しく処理されるように正確に表現されていなければならない。

項目 A.05 は、CKMS の設計にあたって、CKMS セキュリティポリシーが自動処理される場合の CKMS セキュリティポリシーの表現手法について明確化することを要求したものである。なお、自動処理される部分がなければ対象外の項目である。

例えば、X.509 証明書においては「証明書ポリシー拡張」が存在し、拡張を閲覧することにより対象となる主体に対応するポリシーを確認することが可能となる。その拡張を参照して、所定のポリシーを満たすか否かの判断が自動的に行われるような場合は、その旨を記載することとなる。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.05	当該 CKMS のセキュリティポリシーは、RFC 5280 に規定された X.509 v3 証明書の証明書ポリシー拡張にて、ポリシー識別子が ASN.1 DER 形式で記載される。証明書の検証処理において、当該拡張部のポリシー識別子の確認処理が実行される。

2.2 情報管理ポリシー等からの要求事項

解説・考慮点

本節は、SP 800-130 の 4.6 節、4.7 節に記載されている事項について解説したものである。

CKMS の設計にあたっては、CKMS セキュリティポリシーよりも上位のポリシー群から要求される事項が存在する場合がある。本節では、そのような上位のポリシー群から要求されることが多い事項を取り上げる。

① 機微な情報を管理するために、「個人の説明責任（Personal accountability）」について情報管理ポリシー等の要求事項に記載される場合には、どのように対応するかを決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.06	FR4.6	CKMS 設計は、個人の説明責任（personal accountability）が CKMS でサポートされるかどうか、及びどのようにサポートされるかを明記しなければならない。	4.6 節

解説・考慮点

「個人の説明責任」とは、ユーザが機微な情報にアクセスした行為が正当なものであることを保証することである。そのために、認可された範囲でのみ機微な情報にアクセスできることや、認可されないアクセスを検知・防御・管理者に通報することなどの機能を実現することが求められる。

項目 A.06 は、CKMS の設計にあたって、そのような機能を備えるかどうか、また備えるとすればどのように実現するのか明確化することを要求したものである。

個人の説明責任が要求される代表的な場面としては、監査及びリスクマネジメントがあげられる。例えば、複数の個人が行った一連のプロセスで事故や障害等が発生し、それらのインシデントが何かの個人のミス又は不適切な操作に起因する可能性が疑われるような状況において、各個人が自身の行った行動について説明を求められることがありうる。CKMS がその説明責任に応えることが可能となるように設計されているのであれば、その点について記載するべきというのが項目 A.06 の趣旨となる。また、より上位のポリシー等で説明責任が要求された場合も、項目 A.06 を参照することで、CKMS において説明責任を実現しているか否かの判断をすることができる。

個人の説明責任を果たす機能の具体例には、適切なアクセスコントロールの実施を可能とする設定ファイルやログファイル等が含まれる。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.06	当該 CKMS 設計は個人の説明責任をサポートする機能を備える。 CA サーバ及び HSM に対する各種処理の実行は、それぞれのアクセスコントロールによって権限を持つエンティティに制限されている。また、それらの各種処理やアクセスコントロールの設定及び変更の処理は、改ざん困難な実行ログに、実行時のエンティティ ID と共に記録される。

② エンティティに対するプライバシーの提供、関連法令の遵守、又はセキュリティ強化のために、「匿名性」「連結不可能性」「観測不可能性」（のいずれか）の保証について情報管理ポリシー等に記載される場合には、どのように対応するかを決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.07	FR4.7	CKMS 設計は、CKMS でサポートできる匿名性、連結不可能性（unlinkability）、及び観測不可能性（unobservability）に関するポリシーを明記しなければならない。	4.7 節
A.08	FR4.8	CKMS 設計は、どの CKMS トランザクションが匿名性保護を提供している、又は提供可能であるのかを明記しなければならない。	4.7.1 節
A.09	FR4.9	CKMS 設計は、匿名性の保証を提供する場合、CKMS トランザクションの匿名性保証をどのように達成するのかを明記しなければならない。	4.7.1 節
A.10	FR4.10	CKMS 設計は、どの CKMS トランザクションが連結不可能性（unlinkability）保護を提供している、又は提供可能であるのかを明記しなければならない。	4.7.2 節
A.11	FR4.11	CKMS 設計は、CKMS トランザクションの連結不可能性（unlinkability）をどのように達成するのかを明記しなければならない。	4.7.2 節
A.12	FR4.12	CKMS 設計は、どの CKMS トランザクションが観測不可能性（unobservability）保護を提供している、又は提供可能であるのかを明記しなければならない。	4.7.3 節

A.13	FR4.13	CKMS 設計は、CKMS トランザクションの観測不可能性（unobservability）をどのように達成するのかを明記しなければならない。	4.7.3 節

解説・考慮点

以下のセキュリティ特性はいずれもプライバシー保護に効果があるものである。

- 匿名性：パブリックなデータを所有者と関係付けることができないことを保証
- 連結不可能性：情報処理システムにおいて 2 つ以上の関連する事象を互いに関係付けることができないことを保証
- 観測不可能性：観測者がトランザクションに関係する当事者の識別子（ID）を特定又は推定することができないことを保証

項目 A.07～A.13 は、CKMS の設計にあたって、匿名性、連結不可能性、観測不可能性といったセキュリティ特性を実現するプライバシー保護機能を備えるかどうか、また備えるとすればどのように実現するのか明確化することを要求したものである。

なお、システムが扱う情報の種類によってはプライバシーを提供することが適切ではない場合もあり得る。そのようなシステムに対しては、「プライバシー保護機能を提供してはならない」という選択を行うことも容認される。

これらの要件は、典型的には個人情報保護法や GDPR 等の法令準拠の観点で CKMS に求められ得る。匿名性はトランザクションに関わる主体を特定することが困難な性質であり、連結不可能性は複数のトランザクションが同一の主体に関わるものであることを当人以外が判定することが困難な性質であり、観測不可能性はトランザクション自体を当人以外が観測することが困難な性質を指す。

CKMS がこれらの性質を持つことが可能であれば、その旨を明示し（項目 A.07）、その性質がどのように達成されるかを明記しなければいけない(項目 A.08-A.13)。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。当該 CKMS が発行する証明書には対象となる IoT 製品の ID が記載されるが、この ID がどのように割り振られるか、及び ID や証明書をどのように利用するかは、本 CKMS のスコープ外となる。そのため、IoT 製品の ID 自体を保護する機能（匿名性、連結不可能性、観測不可能性）についても当該 CKMS のスコープ外である。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.07	当該 CKMS 設計は、匿名性、連結不可能性、及び観測不可能性をサポートしない。
A.08	当該 CKMS 設計は、トランザクションの匿名性保護を提供しない。
A.09	対象外である。
A.10	当該 CKMS 設計は、連結不可能性保護を提供しない。

A.11	対象外である。
A.12	当該 CKMS 設計は、観測不可能性保護を提供しない。
A.13	対象外である。

2.3 ドメインのセキュリティポリシー

2.3.1 セキュリティドメイン

解説・考慮点

「設計指針」4.3.1 節では、セキュリティドメイン及びそのポリシー（セキュリティドメインポリシー）を以下のように説明している。

> セキュリティドメインとは、同じドメインのセキュリティポリシー下で運用されるエンティティ／CKMS の集合のことである。互いに信頼するエンティティが同じセキュリティドメインに属しているとき、両者はドメインのセキュリティポリシーが要求する保護を提供しながら鍵情報を交換できる。
> ドメインのセキュリティポリシーが要求する保護の保証には、以下を含む。
> - 鍵情報（暗号鍵やメタデータ）を認可されない開示（窃取）から保護すること
> - 鍵情報（暗号鍵やメタデータ）の認可されない改変（改ざん）から保護すること
> - アプリケーションに要求された際の鍵情報（暗号鍵やメタデータ）のソース（送信者）及びディスティネーション（受信者）を確認できること
>
> このようなセキュリティドメインの例には、公開鍵証明書を発行する PKI がある。

2.3.2 異なるセキュリティドメイン間での鍵情報の交換

① 異なるセキュリティドメイン間で鍵情報の交換が必要な場合には、それができるためのルールを決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.14	FR4.15	CKMS 設計は、同等だが異なるセキュリティ保護を提供するとみなせる他のセキュリティドメインに属するエンティティ間での鍵情報（暗号鍵及びメタデータ）の交換を許可する設計仕様を明記しなければならない。	4.9.1 節
A.15	FR4.16	CKMS 設計は、鍵情報（暗号鍵やメタデータ）を異なるセキュリティドメインに属するエンティティ間で共有するときに実施されるソース認証ポリシー（source authentication policy）とデスティネ	4.9.2 節

		ーション認証ポリシー（destination authentication policy）を明記しなければならない。	
A.16	FR4.17	CKMS 設計は、鍵情報（暗号鍵やメタデータ）を異なるセキュリティドメインに属するエンティティ間で共有するときに実施される機密性と完全性のポリシーを明記しなければならない。	4.9.2 節
A.17	FR4.18	CKMS 設計は、他のセキュリティドメインのエンティティと通信するときに要求される保証要件を明記しなければならない。	4.9.2 節
A.18	FR4.19	CKMS 設計は、ドメイン間通信が許可される前に他のドメインのセキュリティポリシーのレビューと検証をサポートするかどうか、またどのようにサポートするのかを明記しなければならない。	4.9.3 節
A.19	FR4.20	CKMS 設計は、弱いポリシーを持つセキュリティドメインのエンティティとの通信がもたらす潜在的なセキュリティに関する影響をどのように検知、防止、又はエンティティに警告するのかを明記しなければならない。	4.9.3 節

解説・考慮点

2 つのエンティティが異なるセキュリティドメインに属しているとき、それらのエンティティは異なるドメインのセキュリティポリシーの下で運用されているため、交換した鍵情報に対して同等の保護を提供することができない可能性がある。

そのため、提供されるセキュリティ保護に関して、それぞれのセキュリティドメインに責任を持つオーソリティ（authority）が他方のセキュリティポリシーを自分自身のポリシーと同等であるかどうかを判断し、互いのエンティティが同等（ただし、同一ではなく異なる場合もある）のセキュリティポリシーであると承認した場合、他方のドメインに属するエンティティともデータ共有が可能となる。もし弱いセキュリティポリシーを持っているセキュリティドメインであると判断した場合には、あらゆる潜在的な危殆化の影響を軽減するために、鍵情報の交換を制限又は拒否することもある。

なお、共有した鍵情報は、他の同等のセキュリティドメインとも共有され得る（第三者共有）と認識しておく必要がある。

項目 A.14～A.19 は、CKMS の設計にあたって、異なるセキュリティドメイン間での鍵情報の交換が必要な場合に、互いのセキュリティポリシーの検証方法や、鍵情報を交換するための手順等を明確化することを要求したものである。なお、異なるセキュリティドメイン間での鍵情報の交換がなければ対象外の項目である。

異なるセキュリティドメイン間で鍵情報を交換する場合は、本節で述べるような様々な配慮が必要となる。そのため、その必要性が特にない場合においては、異なるセキュリティドメイン間での鍵交換を行わない、又は禁止することが一般的なアプローチとなる。

一方で、多数のステークホルダが関与するセキュリティシステムを、多様な用途に利用する場合においては、異なるセキュリティドメイン間での鍵情報の交換が必要となることもある。その場合における CKMS 設計では、本節に記載するような検討が必要となる。

なお、一般に、異なるセキュリティドメイン間で情報を交換する場合は、以下の順番で処理を行う。ここで、ベースラインとなるセキュリティ要件の簡単な例としては、各業界団体におけるセキュリティ規範等が挙げられる。

1） 情報交換の対象となるセキュリティドメイン間で、ベースラインとなるセキュリティ要件を規定し、その規定を守ることに合意する。
2） 上記対象となる各セキュリティドメインが、ベースラインとなるセキュリティ要件を満たすことを確認する。
3） 上記セキュリティ要件を満たしたドメイン間で情報の交換を行う。

日本の政府認証基盤 GPKI（Government Public Key Infrastructure）は本節に該当する事例であり、GPKI ではセキュリティドメインが異なる様々な CA に対して相互接続を行うためのブリッジ CA を設けている。GPKI ではブリッジ CA が行政機関の CA と民間の CA 等の信頼関係を仲介し、ブリッジ CA と個々の CA 間で相互認証の証明書を発行する。異なるドメインのエンティティ間で情報交換を行う際は、自ドメインの CA を起点に、相互認証されたブリッジ CA 及び相手方 CA を含む証明書チェーンによって相手方証明書の正当性を確認し、対向するエンティティの署名を検証して署名者の認証を行うことができる。ブリッジ CA は相互認証先の CA が満たすべき要件を相互認証基準として定めており、相互認証先の CA がこの基準に準拠することが要求されている。

項目 A.14 には、異なるセキュリティドメイン間で情報の交換を許可しない場合はその旨を記載し、許可する場合は情報交換のための設計仕様を記載する。項目 A.14 で設計仕様を記載した場合は、項目 A.15 から A.21 により詳細な情報を記載する。項目 A.15 では情報の送信元と受信先をどのようなポリシーで認証するかを記載し、項目 A.16 では交換する時に実施される（交換情報の）機密性と完全性保護のためのポリシーを記載し、項目 A.17 では要求される保証要件、項目 A.18 ではいかに相手のポリシーのレビューを行うか、項目 A.19 では通信相手のポリシーが弱い場合にどうするかを記載する。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。当該 CKMS は、単一のセキュリティポリシーのもとで、単一の企業により運用されている。そのため、項目 A.14 に記載のとおり、他のセキュリティドメインとの情報共有をサポートしないことを念頭に設計されているものとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.14	当該 CKMS 設計は他のセキュリティドメインとの情報共有をサポートしない。
A.15	対象外である。
A.16	対象外である。
A.17	対象外である。

A.18	対象外である。
A.19	対象外である。

② ドメインのセキュリティポリシーの変更が設定可能なシステムであり、その変更が機能の範囲内であっても、あらゆるポリシーの変更は実行前にドメイン管理者が必ず承認するなど、予め変更ルールを決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.20	FR4.26	CKMS 設計は、異なるドメインのセキュリティポリシー及び異なるアプリケーションをサポートするように、鍵情報（暗号鍵やメタデータ）の管理機能を設定することができるかどうか、及びどのように設定するのかを明記しなければならない。	4.9.7 節
A.21	FR4.27	CKMS 設計は、異なるセキュリティドメイン間のエンティティ同士との通信に適応するために、再設定によるドメインのセキュリティポリシーの変更をサポートしているかどうか、及びどのようにサポートできるかを明記しなければならない。	4.9.7 節

解説・考慮点

ドメインのセキュリティポリシーは、時々、改訂・更新されることが望ましい。

しかし、異なるセキュリティドメイン間での鍵情報の交換が認められている場合、別のセキュリティドメインが改訂・更新したドメインのセキュリティポリシーが、承認されている元のセキュリティポリシーと整合的ではない可能性があり得る。そのため、ドメインのセキュリティポリシーの変更が設定可能であっても自由に変更できるようにすべきではなく、変更前にドメイン管理者の承認を必要とするなど、予め決められた変更ルールに従って変更すべきである。

CKMS の設計にあたって、項目 A.20 は異なるセキュリティドメイン間でのセキュリティポリシーをサポートするように鍵管理機能が設定可能であるか、可能であるならばどのように設定するのか明確化することを、また A.21 ではセキュリティポリシーの変更に伴う再設定が可能であるか、可能であるならばどのように再設定するのか明確化することを要求したものである。これらも、異なるセキュリティドメイン間での鍵情報の交換がなければ対象外の項目である。

上記のように、本節は単なるドメインのセキュリティポリシーの変更に関わる項目ではなく、異なるセキュリティドメイン間での鍵情報の交換を行うために、互いのセキュリティポリシーを（動的に）変更することを可能とするかどうかに関わる項目であることに注意すべきである。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。当該 CKMS は、単一のセキュリティポリシーのもとで、単一の企業により運用されている。

そのため、A.14 にも記載のとおり、他のセキュリティドメインとの情報共有をサポートしないことを念頭に設計されているものとする。したがって、本節は対象外である。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.20	対象外である。
A.21	対象外である。

2.3.3 マルチレベルのセキュリティドメインポリシーを持つセキュリティドメインとの鍵情報の交換

① マルチレベルのセキュリティドメインをサポートする場合には、それができるためのルールを決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.22	FR4.21	CKMS 設計は、マルチレベルのセキュリティドメインをサポートするかどうかを明記しなければならない。	4.9.5 節
A.23	FR4.22	CKMS 設計は、サポートするセキュリティドメインのそれぞれのレベルを明記しなければならない。	4.9.5 節
A.24	FR4.23	マルチレベルのセキュリティドメインをサポートしている場合、CKMS 設計は、それぞれのセキュリティレベルに属する鍵情報（暗号鍵及びメタデータ）の分離をどのように保持しているのかを明記しなければならない。	4.9.5 節
A.25	FR4.24	CKMS 設計は、鍵情報（暗号鍵及びメタデータ）のアップグレード又はダウングレードをサポートするかどうか、及びどのようにサポートするのかを明記しなければならない。	4.9.6 節
A.26	FR4.25	CKMS 設計は、アップグレード又はダウングレード機能をどのようにドメインオーソリティ（domain authority）に制限しているかを明記しなければならない。	4.9.6 節

解説・考慮点

マルチレベルのセキュリティドメインとは、2 つの分離された保護レベルを有しているセキュリティドメインのことである。マルチレベルのセキュリティドメインに属するエンティティは、異なったセキュリティレベルで運用しているドメインに属するエンティティからの鍵情報（暗号鍵やメタデータ）を処理できるようになる。

マルチレベルのセキュリティドメインに属するエンティティは、2 つ（以上）の保護レベルを区別し、異なる保護レベルの鍵情報（暗号鍵やメタデータ）が互いに混同されないことを保証しなければならない。

また、保護レベルを変更するアップグレード（低セキュリティの鍵情報を高セキュリティの鍵情報として扱う）／ダウングレード（高セキュリティの鍵情報を低セキュリティの鍵情報として扱う）ともに、提供される保護レベルが異なることからセキュリティ上何らかのリスクが発生する。例えば、アップグレードは低レベルドメインからの鍵情報（暗号鍵やメタデータ）を高レベルドメイン側が受け入れるということであるから、当該鍵情報（暗号鍵やメタデータ）のソース及び信頼性に確信を持っている場合にのみ行うべきである。一方、ダウングレードは低レベルのセキュリティしか提供されないことから、送付する鍵情報（暗号鍵やメタデータ）に対して低レベルの保護でもよいと判断された場合に限り実行すべきである。

CKMS の設計にあたって、項目 A.22～A.24 はマルチレベルのセキュリティドメインをサポートするか、サポートするならばどのように運用するのか明確化することを、また A.25 と A.26 はアップグレード／ダウングレードが可能であるか、可能であるならばどのようなルールの下で行うのか明確化することを要求したものである。これらは、マルチレベルのセキュリティドメインを設置しなければ対象外の項目である。

図 2-1 のセキュリティドメイン D（ピンク色部分）内のドメインセキュリティポリシーのように、単一のセキュリティドメインポリシーが、異なるセキュリティレベルのドメインと情報交換を行うために、複数のレベルのセキュリティをサポートする場合がある。そのようなケースの例として、扱う情報を機密レベル１、機密レベル２、機密レベル３、等に分けるようなポリシーが該当する。このように複数のポリシーで管理された情報を分離し、異なるセキュリティレベルを持つドメインと情報を交換する場合、マルチレベルのセキュリティドメインに関する記載（項目 A.22 から A.26）を行うこととなる。

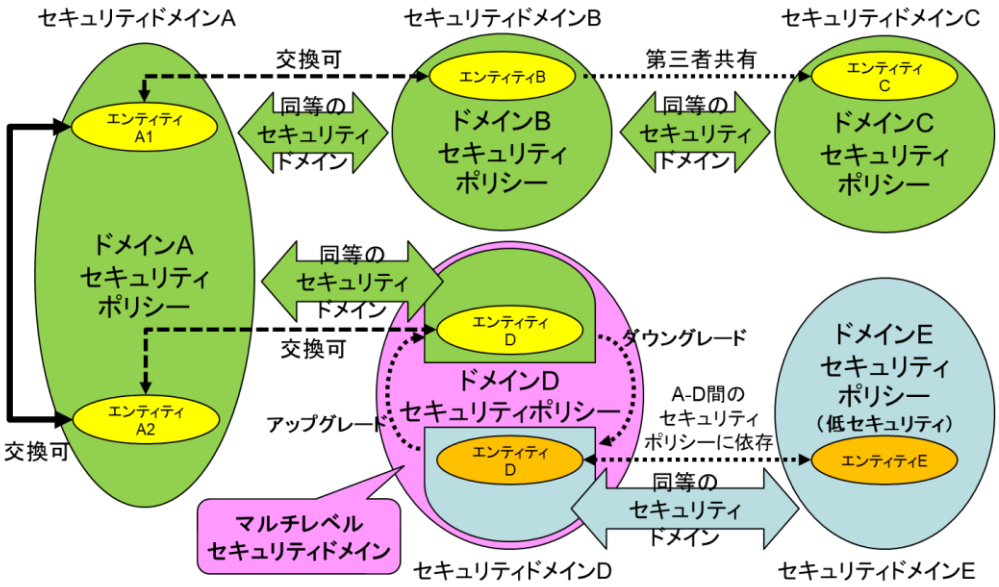


図 2-1　セキュリティドメインとセキュリティポリシーの関係

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。当該 CKMS は、単一のセキュリティポリシーのもとで、単一の企業により運用されている。そのため、項目 A.14 にも記載のとおり、他のセキュリティドメインとの情報共有をサポートしないことを念頭に設計されているものとする。したがって、マルチレベルのセキュリティドメインはサポートしておらず、本節の項目は対象外となる。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.22	対象外である。
A.23	対象外である。
A.24	対象外である。
A.25	対象外である。
A.26	対象外である。

2.4　CKMS における役割と責任

① CKMS 参加者（責任者／管理者／運用者／ユーザ）には、それぞれの役割に応じて定義された特定の認可が必要であり、その役割の責任を果たすために、鍵情報を管理する一連の機能への必要なアクセスだけが提供されなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.27	FR5.1	CKMS 設計は、CKMS に用いられているそれぞれの役割と責任、及びそれぞれの役割にどのようにエンティティが割り当てられるのかを明記しなければならない。	5 章
A.28	FR5.2	CKMS 設計は、CKMS に用いられているそれぞれの役割を満たしているエンティティが使用できる鍵情報（暗号鍵及びメタデータ）の管理機能（SP 800-130、6.4 節を参照）を明記しなければならない。	5 章
A.29	FR5.3	CKMS 設計は、どの役割が役割分離を必要とするのかを明記しなければならない。	5 章
A.30	FR5.4	CKMS 設計は、役割分離を必要とする役割に対してその分離がどのように保持されるのかを明記しなければならない。	5 章
A.31	FR5.5	CKMS 設計は、セキュリティ違反が認可された役割を実行する個人（内部者）によるのか、認可された役割がない人（外部者）によるのかを特定するための全ての自動化された対策を明記しなければならない。	5 章

解説・考慮点

本節は、CKMS 参加者（責任者／管理者／運用者／ユーザ）への権限付与の在り方について取り扱う。

CKMS の運用に関与するのは典型的には人間であるが、個々人に割り当てられる役割は異なり、そのために必要となる権限も異なる。CKMS 参加者に不必要な権限を与えることはインシデント発生時の原因究明の妨げになったり、場合によっては内部犯行を誘発する原因となったりする等、CKMS のセキュリティを低下させる方向に作用する。

責任とは、与えられた権限を適正に利用することであり、そのために付随する行為を含む。例えば、説明責任を果たすための操作ログの取得やセキュリティ維持・向上のためのセキュリティ教育の受講などがある。

CKMS における役割には以下のようなものがある。ただし、これらは例であり、CKMS によってはこれら全ての役割が必要となるわけではなく、またこれら以外の役割が定義されても構わない。最低限、CKMS 全体の最終責任を負う「システムオーソリティ」、CKMS の現場責任者に位置付けられる「システム管理者」及び「暗号責任者」、CKMS 運用から独立して監査を行う「監査責任者」、並びに「CKMS ユーザ」の役割定義が必要である。

なお、個人と役割は必ずしも一対一対応するものではない。ある役割が複数の個人に割り当てられることもあるし、ある一個人に対して複数の役割が割り当てられることもある。

a) システムオーソリティ
b) システム管理者
c) 暗号責任者
d) ドメインオーソリティ
e) 鍵管理者
f) 鍵所有者
g) CKMS ユーザ
h) 監査責任者
i) 登録エージェント
j) 鍵復元エージェント
k) CKMS オペレータ

CKMS での不正を防止・検知するために、システム的には、それぞれの役割を実行するために必要な範囲内での適切なアクセスコントロールを定める必要がある。

加えて、例えば監査と運用責任といった利益相反するような複数の役割に関しては、同時に両方の役割が割り当てられる個人がいないように、役割分離を行うべきである。また、長期の不正使用の可能性を最小化するために、役割を交代で割り当てることが望ましい。

CKMS の設計にあたって、項目 A.27 は CKMS でサポートする全ての役割及びそれぞれの役割にどのエンティティを割り当てるのか明確化し、A.28 でそれらの役割を実行するために採用するアクセスコントロールの手段を明確化することを要求したものである。

A.29 及び A.30 は、一個人に複数の役割を割り当てる場合にどのようにそれらの役割を混同せずに実行するのか明確化することを要求したものである。

A.31 は、不正が発覚した際の監査のための対策、特に有権限者の不正か否かを判定するための対策について明確化することを要求したものである。

「設計指針」には、役割として上記の 11 種が例示されている。これらの役割が担う責任と権限については SP 800-130 の 5 章を参照されたい。

一般に役割の数が増えるほどより細やかなシステム運用が可能となる一方、オペレーションコストは増加する。CKMS 設計者は、扱う情報の重要性とオペレーションコストのバランスを考慮しつつ、上記 11 種の役割に代表される、システム運用に必要な役割を決定する。限られた影響しか持たない暗号鍵の管理におけるオペレーションコストを下げるために 2、3 種程度の役割のみを利用することもあれば、資産価値の高い情報を扱うためにより多くの役割に分離することもある。

役割を決める際の基本的な考え方は、その役割に求められる職務をもとに責任を規定した上で、その職務を遂行するために必要な最小限の権限を与えることである。

項目 A.27 では、規定した役割とその責任、及び役割を割り当てるエンティティの条件や割り当て方法を明記する。項目 A.28 には、所定の役割が割り当てられたエンティティが使用可能な、鍵情報の管理方法を明記する。例えば、暗号化して鍵情報を管理する場合は暗号化方法を明記し、

署名により鍵情報の完全性を確保する場合は署名方法を明記する。

一人に複数の役割を割り当てることもあるが、その場合でも分離すべき役割については、同一人物に割り当てることがないようにすべきである。例えば、日常的な操作を行う役割と管理や監査をする役割は、異なる人物に割り当てることが望ましい。これにより、それらの役割を持つ人員のグループ同士で、内部不正を相互に牽制することが可能となり、内部不正のリスクが低下する。項目 A.29 は、このような分離すべき役割を明記する。項目 A.30 では、その分離方法を明記する。

項目 A.31 では、情報漏洩などのセキュリティ違反が、どの役割を割り当てられた個人により発生したのか、または外部者により発生したのかを特定する方法を明記する。これにより、情報漏洩が発生した場合においても、役割における個人の説明責任（項目 A.06）が明確になる。このことは、操作ミスや悪意のある挙動から情報漏洩を防ぐことにもなる。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.27	責任者（システムオーソリティ）、管理者（システム管理者及び暗号責任者）、利用者（CKMS ユーザ）、監査者（監査責任者）の役割が存在する。 ● 責任者は、当該 CKMS の日々の運用が的確なのか確認を実施する。責任者は CA システムの責任者としての権限を持ち、その権限により操作ログの取得が可能である。本システムの運用を担当している部署の管理責任者が担当する。 ● 管理者は、当該 CKMS の正常運用に関わるデバイス（CA サーバ、HSM、ルータ）の設定、運用、管理、メンテナンスに責任を持つ。本システムの運用を担当している部署に所属する責任者以外のメンバが担当する。 ● 利用者は、当該 CKMS のサービスを利用して IoT 製品向けの証明書発行手続き及び証明書の失効手続きを行う。IoT 製品の製造や顧客対応を担当する部署に所属するメンバが担当する。 ● 監査者は、当該 CKMS の監査を実施する。本システムの操作ログ、運用ログの閲覧が可能である。社内の内部監査メンバが担当する。外部監査時には内部監査メンバが事前に取得したログ情報を利用して対応する。
A.28	● 管理者は、CA 署名鍵の生成、更新、破棄が可能である。また、HSM 内部の鍵情報のバックアップ・アンド・リストアが可能である。ただし、これらの HSM 内部の鍵情報の操作には HSM 管理者としての権限が必要となる。当該権限の操作はマルチパーティコントロールがされているため、管理者 2 名以上の権限が必要である。管理者はこれに加えて、CA サーバやルータのアクセスコントロールの設定も可能である。 ● 利用者は、CA 署名鍵を使用した証明書発行、CA 署名鍵を使用した証明書の失効処理の要求送出が可能である。 ● 責任者及び監査者は、操作ログ、運用ログの取得と閲覧が可能である。

A.29	A.27 に挙げたそれぞれの役割は分離する。
A.30	A.27 のそれぞれの役割は、CA サーバの OS である Linux のユーザ管理機構により分離する。
A.31	CA サーバにおける証明書発行及び証明書失効アプリケーションのログ、Linux の操作ログ及び HSM の操作ログによりセキュリティ違反者の特定が可能である。

2.5 CKMS の構築環境及び実現目標

解説・考慮点

本節は、SP 800-130 の 2.10 節、3.1 節、3.2 節、3.4 節、3.5 節、6.2 節、7 章に記載されている事項について解説したものであり、CKMS をどのような実現目標を踏まえてどのように構築するのかといった全体像を取り扱う。

2.5.1 構築環境

① 鍵情報を保護、管理及び確立するために利用するデバイス及びコンポーネントの一式を明確化しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.32	FR2.5	CKMS 設計は、CKMS の全ての主要なデバイス（例えば、メーカ、モデル、バージョン）を明記しなければならない。	2.10 節

解説・考慮点

項目 A.32 は、CKMS の設計にあたって、どのようなデバイスやコンポーネントで鍵情報の保護や管理等が行われるか明確化することを要求したものである。例えば、認証された暗号モジュールを利用するなどがある。
なお、コンポーネントとは CKMS を構成するために必要とするハードウェアやソフトウェア、あるいはファームウェアという意味であり、デバイスとは特定の目的を供するコンポーネントの組み合わせを意味する（プロセッサ、通信メディア、ストレージユニットなど全てが該当）。

項目 A.32 が要求するような、CMKS が使用するデバイスを明記することは、例えば特定のデバイスに脆弱性が発見された場合に有用な情報となる。CKMS の管理者はその情報を利用することで、CKMS が当該脆弱性の対象となるデバイスを利用しているか否かを確認することができる。デバイスやソフトウェアに脆弱性が発見された場合には、迅速な対応が求められることも多く、デバイス名のみでなく、使用するソフトウェアのバージョンなども予め明記しておくことが望ましい。

これらの情報は、インベントリ管理、ベンダのサポート契約管理、リプレイス計画の策定、第三者監査などにおいても有用である。さらに、CKMS の機能拡張や増強、将来的な移行対策の検

討においてもこれらの情報は参照される。

当該 CKMS で利用されるソフトウェアやハードウェアの管理台帳が存在し、それを利用することで必要な情報を閲覧可能であれば、その管理台帳へのリンクを記載することで良い。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。本モデルでは、プライベート CA が担当するのは CA の機能のうち、IA（Issuing Authority）と VA（Validation Authority）の機能である。RA（Registration Authority）の機能は CKMS 外部の IoT 製品向け証明書管理端末が担っている。以下の項目 A.32 では上記を想定している。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.32	本プライベート CA システムの構成要素となるデバイスには、CA サーバ、HSM、ルータがある。また、CA サーバ上で動作するコンポーネントには CA ソフトウェアがあり、このソフトウェアによって IA（Issuing Authority）及び VA（Validation Authority）の機能を実現する。 これらのデバイスとソフトウェアのシステムインベントリは<URI>に保存されている。

② 様々な CKMS トランザクションや鍵情報で使用される日時について、正確でかつ Network Time Protocol（NTP）サーバのように権威ある情報源を元にすることが要求される場合のルールを決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.33	FR6.9	CKMS 設計は、システムで使用される日時に要求される正確さと精度を明記しなければならない。	6.2.1 節
A.34	FR6.10	CKMS 設計は、要求される正確さを達成するためにどの権威時刻ソース（authoritative time source）を使用するかを明記しなければならない。	6.2.1 節
A.35	FR6.11	CKMS 設計は、要求される正確さを達成するためにどのように権威時刻ソース（authoritative time source）を使用するかを明記しなければならない。	6.2.1 節
A.36	FR6.12	CKMS 設計は、どの日付、時刻、及び機能が信頼される第三者タイムスタンプ（trusted third-party time stamp）を要求するかを明記しなければならない。	6.2.1 節

解説・考慮点

トランザクションや鍵情報で使用される日時は重要な意味を持つため、正確である必要がある。また、場合によっては、信頼される第三者機関が提供するメカニズムによって日時の正確性を客観的に担保することが必要なこともある。

CKMS の設計にあたって、項目 A.33 は CKMS で使用される日時にどの程度の正確性が要求されるのか明確化することを要求したものである。また、具体的な達成手段として、A.34 及び A.35 は日時の正確性をどのような手段で達成するのか、A.36 は第三者機関によるタイムスタンプをどのように使うのか明確化することを要求したものである。なお、タイムスタンプを利用しなければ A.36 は検討対象外である。

項目 A.33～A.36 は、CKMS で使用する日時に関する項目である。暗号鍵のメタデータには一般に日時が含まれ（例えば、鍵生成日、活性化日、有効期限や失効日、更新予定日など）、日時は CKMS において暗号鍵のライフサイクル管理に利用される重要な要素である。要求される日時の精度はシステムによって異なるため、CKMS で要求される日時の正確さと精度、その実現手段について、権威時刻ソースを含めて明確にすることを求めている。

適切に時刻が管理されておらず、例えば、新しいポリシーに準拠できないとの理由で CKMS の内部時刻を巻き戻して過去のポリシーと時刻で処理を行うことが可能であれば、ポリシー変更の実効性は失われることとなる。このように日時の精度以前に内部時刻の巻き戻しを可能としないことを必要とする場面は多い。

システムによっては、信頼できる第三者機関によるタイムスタンプサービスを利用して、対象となるデータが確かにその日時に存在したことの客観的なエビデンスを必要とすることも考えられる。項目 A.36 は第三者タイムスタンプを必要とする場合の項目である。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。要求する時刻の精度や権威時刻ソースの使用例については、あくまで仮想的な設定であり、ここに記載した設定は必ずしも一般的なものではないことに注意されたい。また、項目 A.34 での「信頼できる NTP サーバ」について、実際には具体的な NTP サーバを記載することになるが、本書では具体的な記載を省略した。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.33	当該 CKMS において証明書の発行や証明書の失効に利用する日時情報は、A.34 記載の権威時刻ソースとの誤差が 10 秒以内であることを必要とする。
A.34	国内の「信頼できる NTP サーバ」を権威時刻ソースとして使用する。
A.35	当該 CKMS の CA サーバ及び HSM は、NTP によって上記権威時刻ソースとの時刻合わせを実施する。
A.36	当該 CKMS では、信頼される第三者タイムスタンプを要求する機能は無い。

2.5.2　実現目標

解説・考慮点

CKMS は、特定の目標を達成するために設計されるべきである。望ましいレベルのセキュリティを提供してアプリケーションと使用する組織のニーズを満たし、手頃なコストで、運用への負の影響が最小限になることを同時に満たすように機能するセキュリティメカニズム一式を規定する。そのためには、使用するセキュリティプロトコル標準（例：TLS、IKE、SSH、CMS）における鍵情報の安全な生成、配付、保管及び保護といった“セキュリティ”視点での実現目標だけでなく、本節で示すような視点での実現目標についても CKMS 設計で考慮する必要がある。

CKMS は利用する上での目的があり、その目的のために CKMS 設計は、暗号処理や認証処理に加えてアプリケーションやネットワークの性能も考慮して設計する必要がある。それにより、CKMS の処理がアプリケーションやネットワークに与える影響を最小限にすることができ、またアプリケーションやネットワークの特性を考慮して暗号アルゴリズムや暗号利用モードを選択することができる。

① CKMS を運用するネットワーク視点での実現目標を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.37	FR3.1	CKMS 設計は、それが機能する通信ネットワークに関しての目標を明記しなければならない。	3.1 節

解説・考慮点

項目 A.37 は、CKMS の設計にあたって、通信バックボーンを形成するネットワークへの影響がどの程度までなら許容できるのか明確化することを要求したものである。それには以下のような観点がある。

- ネットワークの効率性及び信頼性
- ネットワークサイズ及びスケーラビリティ
- ネットワークの特性

ネットワークの信頼性は、例えば有線 LAN を想定するのか、モバイル環境を想定するかで大きく異なる。ネットワークサイズやスケーラビリティは CKMS を利用するエンティティの規模や利用頻度に影響を及ぼす。

例えば、ネットワークの特性の１つの指標に誤り率がある。暗号技術は、秘匿と認証に関わる機能を提供するが、ネットワークの誤り率は、それぞれの暗号技術に対して次のような影響を与えることとなる。秘匿に関わる処理では、特定のケース（CTR や OFB などの暗号利用モードとストリーム暗号）を除いて通信路でのビット誤りが、復号結果においてブロックサイズにまで増大する性質がある。また、認証に関わる処理であるメッセージ認証（HMAC や CMAC など）、認

証暗号（GCM や CCM など）及びデジタル署名では、通信路で 1 ビットでも誤りがあれば検証時に認証エラーが発生する。従って、特に認証に関わる処理を伴う場合は、通信路誤りの影響を低減できる誤り制御を行うことが必要となる。

これらを考慮して CKMS の設計において、運用するネットワークにどのような想定をおくかを定めることを要求している。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。以下に記載した証明書の生成に関わる遅延時間の要求値は、あくまで仮想的なものであることに注意されたい。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.37	証明書の生成は、IoT 製品向け証明書管理端末とプライベート CA 間で工場内ネットワークでの通信が利用される。IoT 製品の生産ラインと連動してオンラインで証明書を生成する必要はないが、年間の IoT 製品の生産台数の計画から、IoT 製品向け証明書管理端末が CSR を送信後に対応する証明書を受信するまでの処理が 5 秒以内に実施される必要がある。HSM における署名付与の処理は十分に高速であるので、この処理時間がネットワークに関わる要求値となる。 なお、証明書の失効に関わる CRL の伝達に関しては 1 時間以内の遅延は許容される。

② アプリケーションでの CKMS 視点での実現目標を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.38	FR3.2	CKMS 設計は、それがサポートすることを意図しているアプリケーションを明記しなければならない。	3.1 節

解説・考慮点

サポートするアプリケーションを踏まえ、単一のアプリケーションに特化して暗号鍵管理機能と緊密統合する CKMS にするのか、多くのアプリケーションを包含して暗号鍵管理機能をできるだけ共有化する汎用的な CKMS にするのかを選択して設計するのが一般的である。
項目 A.38 は、CKMS の設計にあたってどちらの方法の CKMS が有利であるのかを判断するために、どれだけのアプリショーンをサポートするのか明確化することを要求したものである。

項目 A.38 では、当該 CKMS がどのようなアプリケーションをサポートすることを意図しているか明記する。ここで、サポートするアプリケーションは単一とは限らず、目的の異なる複数のアプリケーションをサポートする場合もある。CKMS 設計において、サービス対象とするアプリケーションを定義することを求めている。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.38	本プライベート CA のアプリケーションには次の 2 つがある。これらのアプリケーションは専用のプログラムによって IoT 製品管理端末によって実行される。 ● 証明書発行アプリケーション：IoT 製品の製造時の処理 ● 証明書失効アプリケーション：IoT 製品の出荷後、IoT 製品運用中の処理

③ CKMS に対するユーザニーズの視点での実現目標を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.39	FR3.3	CKMS 設計は、意図するユーザ数とそれらのユーザに課する責任を一覧にしなければならない。	3.1 節

解説・考慮点

項目 A.39 は、CKMS の設計にあたって、CKMS をどのようなユーザが利用するのか明確化することを要求したものである。なお、それらの事項はニーズとして顕在化しているとは限らないので、潜在的なニーズについても検討することが必要である。それには以下のような観点がある。

- 初期及び将来のユーザ数
- 利用目的
- 利用環境（場所、時間等）
- ユーザの能力・前提条件（ユーザに課す知識・責任等）

上記の項目は、システムの前提条件に関わる事項であり、CKMS 設計において明確にすべきである。サービスの継続と共に大きく変化する可能性がある事項については、特に注意が必要となる。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。以下に記載した担当者の人数については、仮想的な数字である。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.39	証明書発行及び署名書の失効の各処理要求は、専用のプログラムによって CKMS 利用担当（項目 A.27 の利用者の役割に該当）が送出する。CKMS 利用担当はプログラムの操作に習熟している必要がある。担当者は、当初は 10 名程度であり、今後 10 年間で最大 100 名程度である。

④ CKMS に対する将来的なスケーラビリティの視点での実現目標を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.40	FR3.14	CKMS 設計は、CKMS のパフォーマンス特性を明記しなければならない。それには、実装された機能とトランザクションのタイプによる処理可能な平均及びピーク時の負荷と、その負荷がかかったときの機能とトランザクションのタイプごとの応答時間を含む。	3.5 節
A.41	FR3.15	CKMS 設計は、増大する負荷要求に応じてシステムを拡張するために、サポートされ使うことができる技術を明記しなければならない。	3.5 節
A.42	FR3.16	CKMS 設計は、増大する負荷要求に対応して CKMS を拡張できる範囲を明記しなければならない。これは、追加される負荷、負荷に対する応答時間、及びコストの観点で表現しなければならない。	3.5 節

解説・考慮点

CKMS の設計にあたって、項目 A.40～A.42 は、将来的なニーズ増大の負荷に CKMS がどの程度まで耐えられるか明確化することを要求したものである。特に、A.40 はパフォーマンス観点で、A.41 及び A.42 はスケーラビリティ観点での項目である。
なお、それらの事項はニーズとして顕在化しているとは限らないので、潜在的なニーズについても検討することが必要である。

CKMS 設計において、パフォーマンスや将来的なスケーラビリティに関わる要求を定めることを求めている。項目 A.42 は CKMS の拡張時に採用できる方法及び拡張の限界やコストについて検討するものである。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。以下に記載した生産台数や証明書の生成に関わる処理時間は、あくまで仮想的なものであることに注意されたい。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.40	本プライベート CA システムは、単体の処理性能としては 50msec 以内に証明書発行及び証明書失効の処理が可能である。
A.41	IoT 製品の生産台数が増大し、証明書発行要求が増大した場合は、CA サーバ及び HSM の増設による並行処理によって対応可能である。
A.42	本プライベート CA による証明書の発行を必要とする IoT 製品の初期生産台数は 100 万台／年、今後の 10 年間で最終的に 1,000 万台／年まで増加する可能性がある。また、失

	効処理を必要とする証明書は、当初は最大 1,000 件／年、今後の 10 年間で 1 万件／年まで増加する可能性がある。 このように、今後 10 年間で 10 倍程度の生産台数増加を想定しているため、10 台の並行処理を可能とするようシステム設計する。1 台の CA システムでは、1 秒間に 20 件の証明書発行の処理が可能であり、10 台の並行処理を行った場合は、1 秒間に 200 件の証明書発行の処理が可能となる。

2.5.3 システム間の相互運用の必要性

① 複数のシステム間で相互運用しようとする場合のルールを決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.43	FR7.1	CKMS 設計は、デバイスのインタフェース間の相互運用性の要求事項がどのように満たされるかを明記しなければならない。	7 章
A.44	FR7.2	CKMS 設計は、サポートすることを意図しているアプリケーションとの相互運用に必要な標準、プロトコル、インタフェース、サポートする処理（service）、コマンド、及びデータフォーマットを明記しなければならない。	7 章
A.45	FR7.3	CKMS 設計は、相互運用性を意図している他の CKMS との相互運用に必要な標準、プロトコル、インタフェース、サポートする処理（service）、コマンド、及びデータフォーマットを明記しなければならない。	7 章
A.46	FR7.4	CKMS 設計は、アプリケーションと他の CKMS に対する全ての外部インタフェースを明記しなければならない。	7 章

解説・考慮点

複数のシステム間で相互運用しようとする場合には、インタフェースの詳細な仕様を有することでのみ達成可能である。 CKMS の設計にあたって、項目 A.43～A.46 は、相互運用しようとする場合の条件やインタフェース等について明確化することを要求したものである。

CKMS がサポートするアプリケーションの拡張やデバイスの置き換え、他の CKMS との相互運用を可能とするために、外部インタフェースやプロトコル、コマンドの仕様を明確にする必要がある。デバイスのインタフェース、アプリケーションのプロトコル・コマンド、CKMS 間の標準プロトコルなどが本節の項目の対象となる。

項目 A.43 は CKMS 内のデバイスの交換や増設、より高機能もしくはより高性能なデバイスへの置き換えなどを念頭に、インタフェース仕様を明確にしておくことを求めている。

項目 A.44 は CKMS が提供するサービスの対象となるアプリケーションとの相互運用に関わる内容であり、項目 A.45 は他の CKMS との相互運用に関わる内容である。これらの項目 A.44 から A.46 はいずれも CKMS がその外部と連携して動作する上で定義されるインタフェース、通信プロトコル、API などを一通り整理することを求めるものであり、対象とするアプリケーションや外部 CKMS の変更や拡張などを検討する際に必要な情報である。

一般にこれらのインタフェースは、対象となる CKMS が将来においてどの程度の拡張性が見込めるかを想定した上で定義される。それらの検討の成果物としてネットワーク図、データフロー図、アーキテクチャ図等が存在する場合は、それらのドキュメントをこれらの項目で参照することが望ましい。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。本プライベート CA のアプリケーションは、IoT 製品向けに証明書を発行する IoT 製品製造時のサービスと、IoT 製品向けに発行した証明書の失効処理を行う IoT 製品運用時のサービスの 2 つであり、項目 A.44 と A.46 はそれらを前提に記載している。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.43	当該 CKMS を構成するデバイスには CA サーバ、HSM、ルータがある。CA サーバとルータ間は TCP/IP で接続する。CA サーバと HSM は LAN ケーブルで接続され、HSM の API は PKCS #11 に基づいている。 デバイス間の通信様式はネットワーク図に記載されている。ネットワーク図は<URI>に保存されている。
A.44	証明書発行要求である CSR は PKCS #10、証明書及び CRL のフォーマットは RFC 5280 の各仕様及びプロトコルに従う。 データフォーマット、コマンドなどのアプリケーションとの相互運用に関わる要件はアーキテクチャ図に記載されている。アーキテクチャ図は<URI>に保存されている。
A.45	当該 CKMS は他の CKMS との相互運用を想定していないので、対象外である。
A.46	当該 CKMS は、同一社内に存在する IoT 製品向け証明書管理端末より、項目 A.44 に記載の証明書の発行及び証明書の失効要求を受け、その処理を実施する。これ以外に外部の機器や CKMS との通信は行わない。

2.5.4 ユーザインタフェースの重要性

① ユーザインタフェース（特に習熟していないユーザに対しての）を検討しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.47	FR3.10	CKMS 設計は、システムへの全てのユーザインタフェースを明記しなければならない。	3.4.1 節

A.48	FR3.12	CKMS 設計は、ユーザインタフェースの設計原理を明記しなければならない。	3.4.2 節
A.49	FR3.13	CKMS 設計は、システムに設計された全てのヒューマンエラー防止又はフェールセーフ機能を明記しなければならない。	3.4.2 節
A.50	FR3.11	CKMS 設計は、提案されたユーザインタフェースの使いやすさに関する､あらゆるユーザ受け入れテストの結果を明記しなければならない。	3.4.1 節

解説・考慮点

CKMS の利用にあたって最も重要な条件は、習熟していないユーザにとって分かりやすくかつ誤りなく安全にシステムを使わせることである。その際、ほとんどのユーザは暗号セキュリティのエキスパートではなく、かつセキュリティは一般に最優先の目的ではないので、使用しているセキュリティ機能の目的を十分に理解していない可能性が高いことに留意しておくべきである。
このため、習熟していないユーザに対するユーザインタフェースほど精錬されたものを用意すべきである。透過的なセキュリティを実現する一方、以下のような確立された使いやすいユーザインタフェースの設計原理を踏まえるべきである。

- 正しい操作を行うことが直感的で容易である
- 誤った操作を行うことが困難である
- 誤った操作を実行したときの回復が直感的で容易である

また、ユーザの技量に適応したユーザインタフェースは、習熟していないユーザをガイドすることができる一方、エキスパートには効率的なショートカットを使い、ステップバイステップのガイダンスを迂回できる。

CKMS の設計にあたって、項目 A.47 は、どのようなユーザインタフェースをサポートするのか明確化することを要求したものである。A.48 及び A.49 は、具体的なユーザインタフェースの設計指針・要求事項の明確化であり、どのように設計するのか明確化することを要求したものである。
A.50 はユーザインタフェースの使いやすさについての評価に関するものであり､評価を実施した際にはその結果を付けるように要求したものである。評価を実施していなければ対象外である。

項目 A.47 では CKMS で使用される全てのユーザインタフェースを把握できるよう、論理・物理問わずに全てを記述する。CKMS では CUI や GUI 以外の物理インタフェース（トークン、ピン・エントリー・デバイスなど）を使用していることも多いが、それらも余さずに記録することが重要となる。

項目 A.48 及び A.49 に記載する事項は、商用既製品であればマニュアルなどに記載されていることもあり、その場合はマニュアル類を参照することで良い。自社開発によるソフトウェアが存

在する場合には、ユーザインタフェースを整理し、それらの情報を利用者向けの教育資料やユーザマニュアルに記載する必要がある。それらの文書を参照すると共に、ユーザインタフェースの設計原理などを整理する。

項目 A.50 については、ユーザのベータテストや QA リストなどの各種テストの結果にユーザインタフェースに関わる内容があれば記載する。これらのテスト結果は、以降に変更を加える際に変更してよい箇所と変更が望ましくない箇所の把握にも利用できる。

一般に、社内に UI・UX を担当する人員が存在する場合は、それらの人員が一元的にユーザインタフェースの設計や管理を行うことで、効率良く統一性のある設計を達成することができる。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。ここでは項目 A.48 から A.50 について、社内で開発した、IoT 製品向け証明書管理端末から本プライベート CA に処理要求を送出するプログラムのユーザインタフェースに絞って検討を行ったことを想定して記載例を作成した。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.47	当該 CKMS のサービスを利用するプログラムのユーザインタフェースは、社内担当者向けのユーザマニュアルに記載している。また、CA ソフトウェアや HSM のセットアップ及び設定変更に関わるユーザインタフェースはそれぞれの製品に固有のマニュアルに記載されている。
A.48	当該 CKMS のサービスを利用するプログラムのユーザインタフェースの設計原理は、社内の UI 設計ガイドに基づいている。
A.49	当該 CKMS のサービスを利用するプログラムでは、使用できるコマンドは最小限に抑えられており、誤操作の可能性は低い。
A.50	当該 CKMS のサービスを利用するプログラムは、操作に習熟した社内の担当者が利用することを想定しているため、ユーザインタフェースに関わる評価を行っていない。

2.5.5 商用既製品の活用

① 商用既製品を活用する場合は、どのように CKMS の目標を満たすのかを検討しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.51	FR3.4	CKMS 設計は、CKMS で使用される商用既製品を明記しなければならない。	3.2 節
A.52	FR3.5	CKMS 設計は、商用既製品によってどのセキュリティ機能が実行されるのかを明記しなければならない。	3.2 節

A.53	FR3.6	CKMS 設計は、CKMS の目標を満たすために商用既製品をどのように設定し拡張するかを明記しなければならない。	3.2 節

解説・考慮点

商用既製品は、入手、運用及び保守のためのコストが特定顧客用にカスタム設計、製造された製品より安いことが多い。その一方、多数の顧客の“最小公倍数”的な要求を満たすように設計し製造されているので、セキュリティ要求を完全には満たさない可能性もある。したがって、拡張性と拡充性を許容しサポートしている商用既製品が望ましい。
CKMS の設計にあたって、項目 A.51～A.53 は、CKMS のセキュリティ機能部分に商用既製品を採用する場合に、どのような商用既製品を使い、その商用既製品でどのセキュリティ機能を実行し、セキュリティ要求を満たすためにどのような設定をするのか明確化することを要求したものである。

項目 A.51 及び A.52 は、当該 CKMS において、どのような商用既製品が何をするために用いられているのかを把握するために記載するものである。既にシステムインベントリ、ネットワーク図、データフロー図等が存在するのであれば、それらに記録されていることも考えられ、それらを引用することもできる。これらの情報は、商用既製品のアップグレードを行う際のマイグレーション準備、他社製品への置き換え、同等製品の追加配備による処理能力の増強等を行う際に非常に有用な情報となる。

商用既製品の中には、自社開発では取得が困難なセキュリティ認証を既に取得しているものもある。そうした製品を利用することで、セキュリティポリシーが定める鍵管理機能の具現化や監査要件などを満たす手続きが簡素化されることもある。例えば、FIPS 140-2/-3 レベル 3 もしくは ISO/IEC 15408 EAL4+などのセキュリティ認証を取得した製品の採用により、鍵情報の管理に関する大半の要求を満たせる場合がある。加えて、商用既成品の中には有償のサポート等を提供しているものも多く、技術支援や最新動向の把握をワンストップで行えることは大きな利点となる。

本節の項目 A.51 から A.53 は CKMS において商用既製品を採用しない場合は対象外となる。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.51	次に挙げる商用既製品を利用している。ハードウェア製品としては、サーバ、HSM、ルータであり、ソフトウェア製品としては CA ソフトウェア、Linux OS である。 上記の詳細は、システムインベントリに、メーカと型番、バージョン番号を含めて記載されている。システムインベントリは<URI>に保存されている。

A.52	商用既製品によって実行されるセキュリティ機能は次のとおりである。HSM によって、CA 鍵による証明書への署名付与、CA 鍵による CRL への署名付与、CA 署名鍵の管理が行われる。CA ソフトウェアによって、鍵のメタデータ管理を含むプライベート CA 機能の全体制御が行われる。 当該 CKMS に利用されている商用既製品を含むすべての機器の主な役割、それら機器の配置は、ネットワーク図及びデータフロー図に記載されている。ネットワーク図及びデータフロー図は<URI>に保存されている。
A.53	HSM 及び CA ソフトウェアの適切な設定については、それぞれの製品のセットアップマニュアルに記載されている。当該 CKMS を設定する手順は、これら製品のセットアップマニュアルに基づいた社内の設定管理プロセスによって管理されている。また、設定内容の変更を行う手順はセットアップマニュアルに基づいた変更管理プロセスによって管理されている。 設定変更には、CA ソフトウェアや HSM における設定メニュー内の設定項目の変更からソフトウェアアップデートによる機能修正、機能変更や機能拡張、さらには CA 署名における鍵長の変更など様々なものがある。これらの全てのケースについて変更管理プロセスに一連の実施手順が承認フローを含めて示されている。

2.6 標準／規制に対する適合性

解説・考慮点

本節は、SP 800-130 の 3.3 節、4.8 節に記載されている事項について解説したものであり、CKMS 設計における外的な制約条件となりうるルール等について取り扱う。

① CKMS が使用される地域・国家の法律、ルール及び規制に従わなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.54	FR3.7	CKMS 設計は、CKMS に使用される連邦政府標準（注：米国の場合）、国内標準、及び国際標準を明記しなければならない。	3.3 節
A.55	FR3.8	CKMS に使用されるそれぞれの標準に対して、CKMS 設計は、どの CKMS デバイスが標準を実装しているのかを明記しなければならない。	3.3 節
A.56	FR3.9	CKMS に使用されるそれぞれの標準に対して、CKMS 設計は、標準への適合がどのように検証されるか（例えば、第三者試験プログラムによって）を明記しなければならない。	3.3 節

A.57	FR4.14	CKMS 設計は、CKMS が使用されることを意図する国名や地域名、及び CKMS が実行することを意図する際のあらゆる法的規制を明記しなければならない。	4.8 節

解説・考慮点

標準を使用することは、相互運用性と競争の促進、及び製品又は実装における信頼性を高めることが多い。特に、適合性認証プログラムがある場合、CKMS が正しく実装されていることのさらなる信頼性が得られる。
一方、セキュリティに関して CKMS が使用される地域・国家によって適用される法律等が異なるため、国際的に使用できるように設計される CKMS の場合、各国の制限に従うことができる十分な柔軟性を持っているべきである。

CKMS の設計にあたって、項目 A.54 及び A.55 は CKMS がどのような標準に適合しているのか明確化することを要求したものであり、A.56 及び A.57 は CKMS が使用される地域・国家によって適用される各国の法律・ルール・規則等に準拠していることを明確化するものである。

項目 A.54～A.57 は CKMS で使用されている国内・国際標準を把握するためのものである。CKMS には多数のモジュールが存在し、標準の対象が大規模な複合モジュールとなる可能性もあれば、小規模なサブモジュールとなる可能性もある。また、それぞれのモジュールの標準の対象となる領域は、通信プロトコル、ファイルフォーマット、ハードウェア、ソフトウェア、適合性認証プログラム等、多岐に渡り得る。

項目 A.54～A.57 では、CKMS に含まれるモジュールが準拠する標準や仕様書、ポリシー等を記載することになる。システムインベントリ、ネットワーク図、データフロー図、アーキテクチャ図、システムの仕様書などにデバイスやモジュールごとに準拠する標準や第三者認証が記載されていれば、それを引用することも可能となる。監査要件がある場合は、ポリシーに記載されることもある。

項目 A.57 はポリシーや利用規約、契約書の類に記載されているのが一般的であり、それらとの整合性を意識して記載されるべきである。

CKMS に関連する暗号技術や暗号製品、暗号利用システムに関わる標準を以下に例示する。暗号アルゴリズム及び暗号鍵長に関する国内標準には、電子政府推奨暗号リスト（CRYPTREC)、暗号強度要件（アルゴリズム及び鍵長）設定基準（CRYPTREC）がある。また、暗号モジュールや暗号製品・システムに関わるセキュリティ認証の国内・国際標準の具体例として、ISO/IEC15408 に基づく IT セキュリティ評価制度（日本では JISEC、欧州では EUCC)、FIPS 140-2/-3 に基づく暗号モジュール認証（CMVP、日本では JCMVP)、日本での政府情報システムにおけるクラウドサービスの評価認証制度 ISMAP などがある。さらに、2024 年 8 月に経済産業省が公表した「IoT 製品に対するセキュリティ適合性評価制度構築方針」に基づいて構築されたセキュリティ要件適合評価及びラベリング制度（JC-STAR）がある。

法的規制についてはその具体例として、欧州のサイバーレジリエンス法、中国のサイバーセキ

ュリティ法などに加え、各国のデータ規制（欧州では GDPR 等）などがある。また、国内の電子署名法も暗号技術に関連した法令である。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.54	当該 CKMS は以下の暗号アルゴリズム及び暗号鍵長の基準に準拠する。 ● 電子政府推奨暗号リスト（CRYPTREC） ● 暗号強度要件（アルゴリズム及び鍵長）設定基準（CRYPTREC） また、当該 CKMS は、X.509 形式の証明書を発行する CA 機能を含み、その CA 機能は、ITU-T X.509 及び RFC 5280 及びそれらに関連する標準に準拠する。 HSM は FIPS 140-2/-3 レベル 3 の CMVP 認証を取得した製品を利用する。 当該 CKMS が準拠している標準の一覧はセキュリティポリシーに記載されている。
A.55	当該 CKMS では、HSM において、項目 A.54 に記載の暗号アルゴリズム及び暗号鍵長の基準に準拠した設定がなされており、暗号モジュール認証を取得している。また、CA サーバ上で動作する CA ソフトウェアにおいて CA 機能に関わる標準に準拠している。 上記のデバイスやコンポーネントごとに準拠している標準はシステムインベントリ及びアーキテクチャ図に記載されている。それらのファイルは<URI>に保存されている。
A.56	項目 A.54 に記載のように、当該 CKMS で利用する HSM は FIPS 140-2/-3 レベル 3 の CMVP 認証を取得している。本プライベート CA の新規システム構築時及びリプレイス時に有効な CMVP 認証を取得していることを要件とし、ベンダから提供された認証書により確認する。
A.57	当該 CKMS は当社国内の工場内のみでの運用が想定されているので、日本国の法律以外は適用されない。この旨はセキュリティポリシーに記載されている。

2.7　将来的な移行対策の必要性

解説・考慮点

本節は、SP 800-130 の 7 章、12 章に記載されている事項について解説したものである。
長期の利用が想定されている CKMS の場合には、システム的に長期にわたる CKMS のセキュリティライフタイムを持つように設計・実装されるべきであるため、移行戦略があることが望ましい。その際、円滑な移行には、少なくとも 2 つの暗号アルゴリズム（異なった鍵長であるかもしれない）の利用を同時にサポートする機能が要求されることが多い。なお、異なった暗号アルゴリズムによって保護されるデータのセキュリティは最も弱い暗号アルゴリズムを上回らないことにも留意すべきであり、可能な限り素早く移行することが最善である。

① 使用中の暗号アルゴリズムは、必要なときに拡張又は置き換えができるように実装することを検討しておかなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.58	FR7.5	CKMS 設計は、新規の、相互運用可能な、同等のデバイスへの移行のための全ての対策を明記しなければならない。	7 章
A.59	FR7.6	CKMS 設計は、暗号アルゴリズムのアップグレード又は置き換えのために提供されるあらゆる対策を明記しなければならない。	7 章
A.60	FR7.7	CKMS 設計は、暗号アルゴリズムの移行期間中に、どのように相互運用性をサポートするかを明記しなければならない。	7 章
A.61	FR7.8	CKMS 設計は、暗号アルゴリズムと鍵長の使用をネゴシエーションするプロトコルを明記しなければならない。	7 章

解説・考慮点

CKMS が保護する情報に見込まれるライフタイムと同じかそれ以上のセキュリティライフタイムを持つか、もしくはより強固なアルゴリズム及びより長い鍵長に将来移行するための移行戦略がある暗号アルゴリズムだけを利用しなければならない。
項目 A.58～A.61 は、CKMS の設計にあたって、CKMS の移行戦略を実行するためにどのような仕組みや機能を予めサポートしておくか明確化することを要求したものである。

上記のように、CKMS 設計はそれによる保護の対象となるシステムやそのデータのライフタイムをカバーすべく、十分なセキュリティライフタイムを備えた暗号アルゴリズムを採用することとなる。採用した暗号アルゴリズムのセキュリティライフタイムを超えて CKMS サービスを提供する必要がある場合には、鍵長の変更や暗号アルゴリズムの変更が要求される。鍵長や暗号アルゴリズムの変更は、予め設定していた鍵長や暗号アルゴリズムのライフタイムを超える場合の他に、運用中の暗号アルゴリズムが予期せぬ危殆化をした結果、当初の設計よりも早く鍵長や暗号アルゴリズムの変更が必要となる場合もある。

暗号アルゴリズム及びその鍵長のセキュリティライフタイムについては、「暗号強度要件（アルゴリズム及び鍵長）設定基準」（CRYPTREC）を参照するとよい。

鍵長や暗号アルゴリズムをより強度の高いものに移行するための具体的な方法としては、予め複数の鍵長や暗号アルゴリズムをサポートした製品を採用する、暗号製品のベンダが提供するソフトウェアアップデートによって移行する、及び暗号処理を担う製品自体を置き換えるなどが存在する。

通信データの保護や通信時の認証に関わる状況において暗号アルゴリズムを変更する場合は、全ての関連する機器やソフトウェアを一斉に変更するような場合を除いて、各機器やソフトウェアが通信相手と利用する暗号アルゴリズムや鍵長に関して合意することが可能であるようにその通信プロトコル設計がなされている必要がある。また、データ秘匿の目的で暗号化された上で保

存されているデータや署名が付与されて保存されているデータについては、汎用的な解決方法は存在しないものの、移行後の暗号アルゴリズムや鍵長による暗号化や署名付与を適用し直す、暗号学的タイムスタンプを実施する、暗号以外の手段（例えば物理的保護）によって秘匿データや署名データを安全に保管するなどの対策が存在する。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。当該 CKMS での暗号処理は HSM が担っているため、HSM における鍵長や暗号アルゴリズムの置き換え、及び HSM の移行が検討事項となる。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.58	当該 CKMS で利用する HSM について、鍵情報のバックアップ・アンド・リストア処理をサポートする製品への移行においては、鍵情報の転送を伴う移行が可能となる。鍵情報のバックアップ・アンド・リストア処理がサポートされない HSM 製品に移行する場合は、新たな鍵の生成を伴う新規セットアップを行うこととなる。
A.59	当該 CKMS で利用する HSM 選定においては、将来の移行を想定して、署名アルゴリズムと鍵長については、128 ビットセキュリティ（ECDSA P-256 及び SHA-256）及び 192 ビットセキュリティ（ECDSA P-384 及び SHA-384）を選択可能な製品を採用する。署名アルゴリズムの選択は CA ソフトウェアの設定によって行う。 運用中に署名アルゴリズムに危殆化が生じて署名アルゴリズムのアップグレードが必要になった場合は、次のいずれかの方法によって対応する。 192 ビットセキュリティの署名アルゴリズムへの切り替え、HSM ベンダにより提供される更新ファームウェアのサポート範囲内での署名アルゴリズムの更新や鍵長の変更、より強度の高い署名アルゴリズムをサポートした HSM 製品への置き換え、など。
A.60	当該 CKMS は、鍵長の更新以外の暗号アルゴリズムの移行をサポートする構成とはなっていない。暗号アルゴリズム自体の移行が必要となった場合は、CKMS 全体もしくは HSM を含むサブモジュールを再設計して、新たな CKMS として運用を行うこととなる。
A.61	当該 CKMS において、証明書生成や失効処理に利用可能な署名アルゴリズム及び鍵長は、CA サーバにおいて設定されている。また、TLS は利用する暗号アルゴリズム及び鍵長のネゴシエーションを行う機能を備えている。

② 技術の進歩に起因する潜在的な脅威についても考慮しておかなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
A.62	FR12.1	CKMS 設計は、システムに実装されたそれぞれの暗号アルゴリズムの想定されるセキュリティライフタイムを明記しなければならない。	12 章

A.63	FR12.2	CKMS 設計は、CKMS の運用に悪影響を与えることなしに、暗号アルゴリズムのどの副関数（例えば、HMAC の副関数として使うハッシュ関数）が、類似だが暗号学的に改良されている副関数にアップグレード又は置き換えを行うことができるかを明記しなければならない。	12 章
A.64	FR12.3	CKMS 設計は、どの鍵確立プロトコルがシステムによって実装されているかを明記しなければならない。	12 章
A.65	FR12.4	CKMS 設計は、システムに実装されているそれぞれの鍵確立プロトコルの想定されるセキュリティライフタイムを、採用されている暗号アルゴリズムの想定されるセキュリティライフタイムの観点から、明記しなければならない。	12 章
A.66	FR12.5	CKMS 設計は、CKMS デバイスへの外部からのアクセスが許容されている範囲を明記しなければならない。	12 章
A.67	FR12.6	CKMS 設計は、CKMS デバイスへの全ての許可された外部アクセスがどのようにコントロールされるかを明記しなければならない。	12 章
A.68	FR12.7	CKMS 設計は、CKMS の暗号アルゴリズムに対する量子コンピュータによる攻撃のような、新しい技術の発展の影響に抵抗又は軽減するために採用している機能を明記しなければならない。	12 章
A.69	FR12.8	CKMS 設計は、CKMS の暗号に対する量子コンピュータによる攻撃の、現在知られている影響を明記しなければならない。	12 章

解説・考慮点

長期の利用が想定されている CKMS の場合には、CKMS がセキュアでなくなるかもしれない技術の進歩に起因する潜在的な脅威についても考慮すべきである。

以下に 4 つの潜在的な脅威の例を挙げる。項目 A.62～A.68 は、CKMS の設計にあたって、それぞれの脅威に対する現時点で採用されている対策技術（及び対策の限界）について明確化することを要求したものである。なお、潜在的な脅威はこれら 4 つに限るものではない。

- 暗号アルゴリズムに対する新しい攻撃
  もともと暗号アルゴリズムには想定されるセキュリティライフタイムがある。また、時間が経過するにつれ、そのセキュリティライフタイムを短縮する新しい攻撃が発見される可能性もある。暗号アルゴリズムがセキュアでなくなった場合、最終的には、暗号アルゴリズムを完全にアップグレード又は置き換える必要がある。その場合、暗号アルゴリズムは、（当該アルゴリズム以外の）残りの実装への著しい影響なしで置き換え又はアップグレードができるような方法が望ましい。
- 鍵確立プロトコルに対する新しい攻撃
  CKMS のセキュリティは、暗号アルゴリズムの安全性のほか、鍵確立段階での対称鍵の安全性にも依存する。しかしながら、鍵確立プロトコルのセキュリティ評価は、暗号アルゴ

リズムに対して行われるのと同じ程度で評価されることはめったになく、数年間使用された後に弱点が発見されることが少なくない。
しかも、一旦広く使用されるようになると当該プロトコルをアップグレードすることは困難であることも多い。

- CKMS デバイス／アクセスコントロールに対する新しい攻撃
  認可されない当事者が CKMS の外部から CKMS デバイスへアクセスすることを、現実的な範囲で最大限防止しなければならない。CKMS のセキュリティが依存するアクセスコントロールメカニズムは、要求に応じて、最新の攻撃を実行したりアップグレードしたりして定期的にレビューされるべきである。
- 新しい計算機技術の発展
  現状の脅威で最も高い関心が払われているものは、暗号鍵を復元するのに十分な能力を持つ量子コンピュータの発展である。

上記のように、長期の利用が想定される CKMS では、採用した暗号技術に対して新たな攻撃が発見される等の理由により、当初想定していた以上のペースでセキュリティ強度の低下が生じる可能性がある。そのため、長期の利用が想定される CKMS では、関連し得る技術の進歩を常に監視すると共に、潜在的な脅威への備えをしておく必要がある。具体的には、上記に挙げる 4 つの脅威を例とする潜在的な脅威が現実となった場合の影響評価等を予め実施することを推奨する。

暗号アルゴリズムに対する新しい攻撃や新しい計算機技術の発展（特に暗号解読可能な量子コンピュータによる影響）については、CRYPTREC の発信する「注意喚起情報」が参考になる。また、CKMS デバイスに対する新しい攻撃については、デバイスベンダからの情報入手が原則となるため、有償のサポート契約を結ぶことを推奨する。

なお、暗号アルゴリズムのセキュリティライフタイムについては「暗号強度要件（アルゴリズム及び鍵長）設定基準」（CRYPTREC）を参照するとよい。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

A.62	本 CKMS で実装される暗号アルゴリズムは以下となる。これらのアルゴリズムは証明書や CRL への署名付与や IoT 製品向け証明書管理端末との間の TLS 通信において用いられる。 ● ECDSA（P-256, P-384） ● SHA-256、SHA-384 ● AES128、AES256 ● ECDH（P-256, P-384） 「暗号強度要件（アルゴリズム及び鍵長）設定基準（CRYPTREC）」によると、上記アルゴリズムのセキュリティライフタイムは以下のとおりである。

	● 128 ビットセキュリティ：ECDSA(P-256)、SHA-256、AES128、ECDH(P-256)。これら のセキュリティライフタイムは 2040 年まで ● 192 ビットセキュリティ：ECDSA(P-384)、SHA-384、ECDH(P-384)。これらのセキュリティライフタイムは少なくとも 2070 年までは有効 ● 256 ビットセキュリティ：AES256。このセキュリティライフタイムは少なくとも 2070 年までは有効
A.63	ECDSA に利用するハッシュ関数（副関数に相当）である SHA-256 もしくは SHA-384 に危殆化が生じた場合は ECDSA の標準文書に従って対応する。具体的には、同等のハッシュ長のダイジェストを生成する SHA-3 アルゴリズムなどに置き換えることが対応の候補となる。利用する HSM 製品において、代替となる安全なハッシュ関数への置き換えが可能であればそれを検討するが、置き換えができない場合は HSM 自体の置き換えを検討する。
A.64	当該 CKMS では TLS1.3 を IoT 製品向け証明書管理端末との通信に利用する。TLS1.3 における鍵確立には ECDH(P-256)を利用する。
A.65	「暗号強度要件（アルゴリズム及び鍵長）設定基準（CRYPTREC）」によると、TLS1.3 で利用する鍵確立アルゴリズム ECDH(P-256)のセキュリティライフタイムは 2040 年までである。
A.66	当該 CKMS では外部からの直接のアクセスは許可されていない。遠隔地から外部ネットワークを経由してアクセスを行う場合は、所定の VPN エンドポイントを経由し、社内ネットワークへの接続が許可された後に CKMS へ接続する必要がある。
A.67	当該 CKMS へ外部からアクセスを行うには、施設や管理区域への入退出管理、もしくはネットワークセキュリティコントロール及びコンピュータシステムのセキュリティコントロールを経由する必要がある。さらに、HSM 自体が備える HSM 内に保管される鍵に対するアクセスコントロールを介することになる。 なお、HSM において外部アクセスから鍵を保護する機能は、サイドチャネル攻撃や故障利用攻撃などの非侵襲攻撃の進化によって危殆化する可能性がある。そのため、リスク分析等を行った結果、HSM における鍵の保護機能を強化する必要が生じた場合には、より強固な耐性を備えた HSM への移行を検討するものとする。
A.68	本 CKMS 設計において、CKMS 及び各サブモジュールは極力シンプルに作られており、外部アプリケーションとの依存関係も極めて少ない。これは、暗号の置き換えが必要な場合において、CKMS 全体または HSM を含むサブモジュール単位での置き換えを想定しているためである。 さらに、モジュールの置き換えの容易性を確保するために、商用既製品の利用を想定し、CKMS の外部接続や HSM との接続には標準的な API を採用し、暗号アルゴリズムも CRYPTREC 暗号リストに記載される標準的なものから選定している。 上記のような構成のため、（量子コンピュータを含む）新たな技術の進展によって当該 CKMS が危殆化する場合においても、新たな技術に対抗可能な暗号技術をサポートした CKMS への置き換えを比較的容易に実現することが期待できる。 上記のような思想のもと、暗号解読可能な量子コンピュータ（CRQC：Cryptographically

	Relevant Quantum Computer）の実現可能性が高まった場合には、耐量子計算機暗号アルゴリズム（PQC：Post-Quantum Cryptography）をサポートした HSM 及びそれを含む CKMS への置き換えを検討する。
A.69	当該 CKMS の設計時のライフタイムは 10 年である。この期間で暗号解読可能な量子コンピュータが実現されることは、現在の技術水準、及び、これまでの量子コンピュータの発展状況に鑑みると、非常に困難と考えられる（本書を執筆した 2025 年時点の状況）。 また、仮にそのような量子コンピュータの実現を仮定した場合でも、本 IoT 製品で保護する通信内容はリアルタイムで意味を持つ情報であり、過去の通信データを保存しておき後で解読する「ハーベスト攻撃」特有の脅威は小さい。 さらに、項目 A.68 のように当該 CKMS は、PQC をサポートした CKMS への置き換えを比較的容易に実現可能と期待できる。 以上から、量子コンピュータによる暗号解読のリスクに関しては、そのような量子コンピュータの実現の兆候に気づいてからの対応で十分と考える。

3 暗号鍵管理デバイスへのセキュリティ対策

本章の目的・趣旨

本章は、「設計指針」の 8 章に記載されている要求事項（各節での灰色枠内で示している内容）について解説したものである。

本章の記載内容は、暗号鍵を管理するための個々のデバイスに対して、必要に応じて検討する項目（E.01～E.37）をまとめたものであり、主に以下のような検討項目を含んでいる。

- アクセスコントロールシステム／暗号モジュールを利用する際に、それらが有するべき機能や運用方法などはどういったものか
- デバイスのセキュリティ確認のためにどのようなセキュリティ評価試験を実施するか

本章の検討項目は、暗号鍵の管理・保管を実際に行う個々のデバイスを対象としており、「広義」の意味での暗号鍵管理に相当するものの一つである。また、アクセスコントロールシステムと暗号モジュールは暗号鍵のセキュアな管理を行うための主要なコンポーネントであることから、本章でまとめて解説する。

3.1 鍵情報へのアクセスコントロール

3.1.1 アクセスコントロールシステム

① アクセスコントロールへの要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.01	FR6.88	CKMS 設計は、エンティティ、ACS（アクセスコントロールシステム）、機能ロジック、及びそれらの間の接続の配置を示すことで CKMS のトポロジーを明記しなければならない。	6.7.1 節
E.02	FR6.89	CKMS 設計は、適切な操作を保証するために実装されている鍵管理機能に対する制限を明記しなければならない。	6.7.1 節
E.03	FR6.90	CKMS 設計は、鍵管理機能へのアクセスがどのように認可されたエンティティを制限しているかを明記しなければならない。	6.7.1 節
E.04	FR6.91	CKMS 設計は、鍵管理機能へのアクセスを制御するための ACS とそのポリシーを明記しなければならない。	6.7.1 節
E.05	FR6.92	CKMS 設計は、少なくとも以下を明記しなければならない： a) エンティティの粒度（例：人、デバイス、組織） b) エンティティが識別されているかどうか、及びその方法 c) エンティティが認証されているかどうか、及びその方法 d) エンティティの認可が検証されているか、及びその方法	6.7.1 節

		e) それぞれの鍵管理機能のアクセスコントロール	
E.06	FR6.93	CKMS 設計は、CKMS セキュリティポリシーを適応、実装、施行するための ACS の能力を明記しなければならない。	6.7.1 節

解説・考慮点

CKMS のセキュリティは、鍵情報の管理機能の適切なシーケンスと実行に依存する。そのため、鍵情報の管理機能が認可されたエンティティの要求（呼び出し）への応答としてのみ実行されること、及びその他の制限事項が全て満たされていることを保証することが必要である。
アクセスコントロールシステムは、暗号モジュールと連動して、鍵情報への適切なアクセスをコントロールするために動作する。
CKMS の設計にあたって、項目 E.01～E.06 は、アクセスコントロールへの要求事項を明確化することを求めたものである。E.01 及び E.06 はアクセスコントロールの構成や性能、E.02 は機能のコントロール、E.03 はエンティティの認証・認可、E.04 及び E.05 はアクセス条件を対象としている。

上記のように、CKMS のセキュリティは、鍵情報の管理機能が適切に設定され、認可されたエンティティからの要求のみに対応して鍵情報を利用した各種処理が実行されることによって担保される。アクセスコントロールシステム（ACS）はエンティティの認証や認可された処理の実行許可を与える機能モジュールであり、本節は CKMS が備える ACS を定義することを求めている。

ACS は暗号処理の実行主体である暗号モジュールと連動するため、暗号モジュール及び ACS を含む CKMS 全体のトポロジーと処理フローを定める必要がある(項目 E.01)。図 3-1 は SP 800-130 の 6.7.1 節に例示された CKMS のトポロジーである。

ここで暗号モジュールとは、FIPS 140-2/-3 で定義されているように、暗号境界内で暗号処理を実行するハードウェアもしくはソフトウェアの集合である。暗号モジュールは、暗号境界内で利用される暗号鍵の保護機能を備えている。

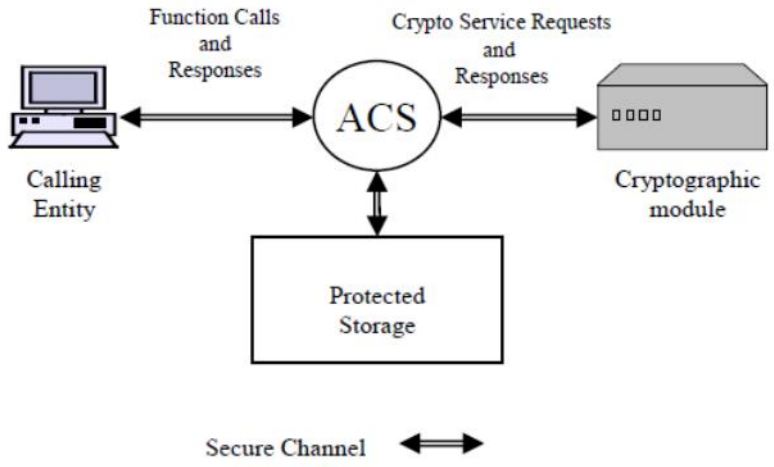


図 3-1　CKMS のトポロジー例（SP 800-130 から引用）

項目 E.02 は暗号モジュールなどの鍵管理機能の実行に対する制限を ACS と絡めて定義することを求めており、項目 E.03 は特にエンティティとの対応付けに関してエンティティ認証、認可の機能を定義することを求めている。また、項目 E.04 は暗号モジュールなどの鍵管理機能へのアクセスをコントロールする ACS が従うポリシーを定めることを求めている。

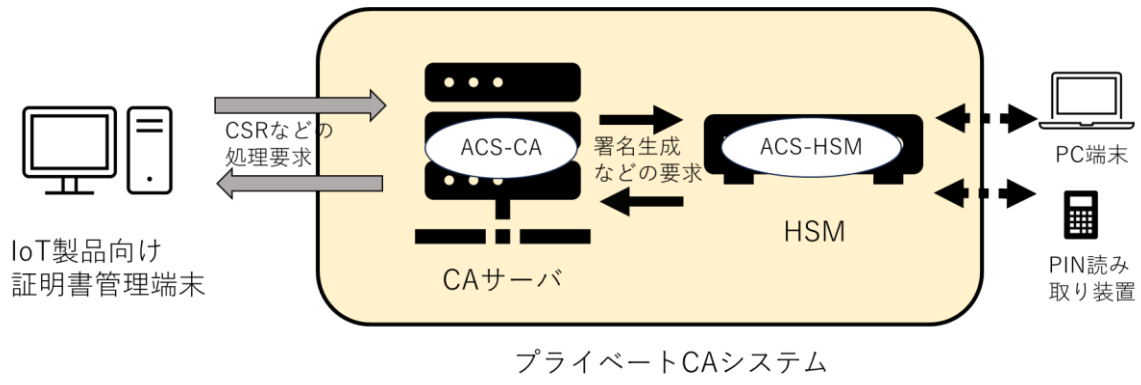
項目 E.05 は ACS におけるエンティティの粒度や識別方法、認証の方法、認可された鍵管理機能を明確にすることを定めており、ACS の本質的な部分である。

項目 E.06 は CKMS 全体のセキュリティポリシーの変更に際して、ACS ポリシーの変更がどの程度可能であるか、ACS ポリシーの更新に関わる手順を定めることを要求している。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.01	本プライベート CA システムにおける ACS は CA サーバと HSM の 2 つ（ACS-CA と ACS-HSM）がある。ACS のトポロジーは図 3-2 を参照。 CA サーバ内の ACS-CA が要求を送出した利用者を識別・認証し、認証結果に応じて利用者に割り当てられた役割（ロール）の実行許可を与える。CA サーバは HSM に対して処理の依頼を利用者に代わって送出し、HSM から受け取った処理結果を利用者に返す。ここで、HSM 内の ACS-HSM は CA サーバのロールを認証して処理の実行を許可する。ACS に関わる処理フローは図 3-3 を参照。 一方、HSM 管理者の識別・認証は、CA サーバ内の ACS-CA を介さずに HSM 内の ACS-HSM によって行われる。その際、HSM に専用の PIN 読み取り装置を接続して認証処理を行う。また、HSM への管理用コマンドの送信には PC 端末から HSM がサポートしている方式（例えば、シリアル接続や SSH 等）で接続して処理を行う。  **図 3-2　ACS の配置**

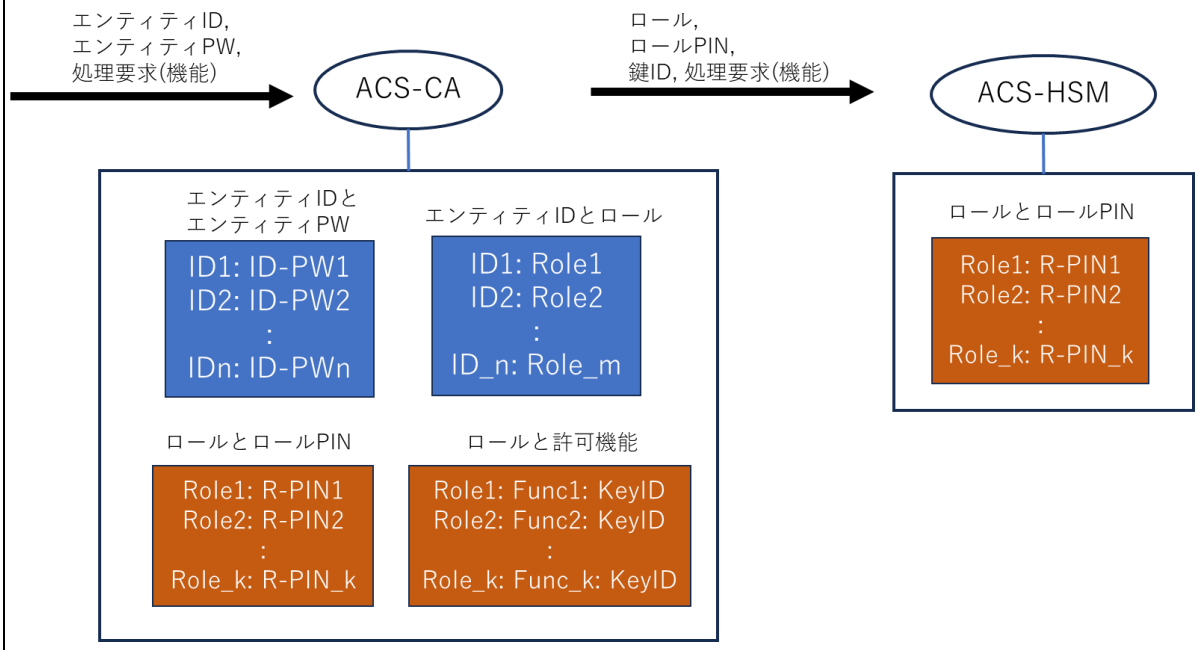


図 3-3　ACS に関わる処理フロー

E.02	HSM の機能である、署名鍵生成機能、署名生成機能、署名鍵の更新機能、署名鍵の破棄機能、署名鍵のバックアップ機能、HSM ログのアーカイブ機能のそれぞれの実行は権限を有するエンティティでなければ実施できない。
E.03	CKMS 利用担当(証明書の作成、失効処理の実行)、CKMS 管理担当(CA 鍵の生成、更新、破棄、バックアップ、HSM ログのアーカイブ)、監査者(HSM ログの閲覧)をそれぞれの鍵管理機能に対して認可されたエンティティとする。これら機能の実行権限の設定、認証や認可が ACS-CA 及び ACS-HSM によって管理される。
E.04	各エンティティは個人単位で識別・認証され、設定したロールに基づき認可された権限(E.03 に記載)を実行できる。ACS の設定や更新を行う場合、CKMS 管理担当が CA サーバ及び HSM の管理者権限を取得して設定を変更する。
E.05	a) CA サーバにおける ACS-CA でのエンティティの粒度は個人単位である。また、HSM における ACS-HSM でのエンティティの粒度はロール単位である。 b) CA サーバにおけるエンティティはエンティティ ID によって識別される。HSM ではロールによって識別される。 c) CA サーバにおけるエンティティの認証はエンティティパスワードによる。また、HSM におけるロールの認証はロール PIN による。ただし、HSM の管理者権限の認証では PIN に加えて複数の USB ドングルを専用の PIN 読み取り装置に接続する必要がある。HSM 管理者権限の取得に関わる認証処理は、マルチパーティコントロールが適用されている（E.23 記載例を参照）。 d) CA サーバ及び HSM においてエンティティはロールベースで権限が付与される。ACS-CA にはエンティティの ID ごとに対応するロールを管理するデータベースがあり、さらにロールと許可された鍵管理機能及び当該処理に利用する鍵 ID の対応

	を管理するデータベースがある。ACS-HSM では、PKCS #11 仕様に基づいてロールや鍵を管理しており、ACS-HSM で認証された利用者は、利用可能な鍵を検索して署名処理などを実行する。 e) CA サーバの管理者権限に関わる機能、操作ログ閲覧及び操作ログアーカイブの各機能のアクセスコントロールは ACS-CA によって管理されている。一方、HSM の管理者権限に関わる機能、HSM の操作ログ閲覧及び操作ログアーカイブの各機能のアクセスコントロールは ACS-HSM によって管理されている。それ以外の各種機能のアクセスコントロールは ACS-CA と ACS-HSM の双方によって管理されている。
E.06	CKMS セキュリティポリシーを変更する場合、CA サーバ内の ACS-CA 及び HSM 内の ACS-HSM の更新が必要となる場合がある。ACS の更新を行う場合、CKMS 管理担当が CA サーバ及び HSM の管理者権限を取得して設定を変更する。更新後にはそれぞれの更新内容が整合していることを確認する。

3.1.2 暗号モジュール

① 暗号モジュールセキュリティポリシーを定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.07	FR8.19	CKMS 設計は、以下を含む、使用する暗号モジュール及びそれぞれのセキュリティポリシーを特定しなければならない： a) それぞれのモジュールの実装形態（ソフトウェア、ファームウェア、ハードウェア、又はハイブリッド） b) それぞれのモジュールの完全性を保護するために使用されるメカニズム c) それぞれのモジュールの暗号鍵を保護するために使用される物理的及び論理的メカニズム d) それぞれのモジュール（セキュリティ機能を含む）で実行された第三者試験と検証、及びそれぞれのモジュールで使用される保護措置	8.4 節

解説・考慮点

一般にコンピュータは暗号鍵への十分な保護を提供するようには設計・実装されていない。実際、同じコンピュータ上にセキュリティが検証されていないソフトウェアが含まれていることから、当該コンピュータ上の暗号ソフトウェアでは物理的に保護されていること及び信頼できないソフトウェアによる攻撃から論理的に保護されていることが重要である。

その対策の一つとして、暗号モジュールの利用がある。暗号モジュールは、暗号境界内に実装される暗号ベースのセキュリティ機能全てを包含しており、実装形態はハードウェア、ソフトウェア、ファームウェアを問わない。

暗号モジュールの目的は、実装されたセキュリティ機能の完全性と鍵情報の保護を行うことであり、暗号モジュールセキュリティポリシーに従って、改ざんや窃取から物理的及び論理的に保護するように作られている。このため、CKMS では、暗号モジュールを使用して暗号鍵を生成し、保管、使用及び保護を行うことが望ましい。
ただし、暗号モジュールが提供するセキュリティ機能や保護レベル等は、暗号モジュールセキュリティポリシーに大きく依存することに留意されたい。

項目 E.07 は、CKMS の設計にあたって、暗号モジュールへの要求事項を暗号モジュールセキュリティポリシーの形で明確化することを求めたものである。暗号モジュールは、E.07 で定めたセキュリティポリシーに則って利用しなければならない。

上記のように、多くの汎用的なコンピュータは暗号鍵の十分な保護を提供するような設計や実装はされていない。実際に、暗号処理を実行するコンピュータ上にセキュリティ面で十分な検証がされていないソフトウェアが動作する場合、ソフトウェア自身の脆弱性をついた攻撃によって暗号鍵のセキュリティに影響が及ぶことがある。他方で、暗号モジュールは汎用的なコンピュータから暗号鍵を保護しながら暗号処理を実行することができる。暗号モジュールの実装形態はハードウェア、ソフトウェア、ファームウェアと様々であるが、典型的なものにハードウェア・セキュリティ・モジュール（HSM）が存在する。

暗号モジュールはそれ自身のセキュリティポリシーに従って構成されており、暗号モジュール内に実装されているセキュリティ処理機能の完全性保護及び保管されている暗号鍵の保護が実現される。暗号モジュールが備える保護機能の強度は、暗号モジュールの実装形態や保護機能として利用されるメカニズムに依存する。暗号モジュールのセキュリティポリシーの実例は NIST の CMVP 認証の検証済み製品のサイトでも参照できる。

暗号モジュールを対象とした第三者試験には FIPS 140-2/-3 や ISO/IEC 15408 がある。FIPS 140-2/-3 では暗号モジュールのセキュリティについてレベルが定義されている。ただし、レベルの値が大きいほど厳格な環境条件の整備が要求されるなど運用コストの上昇が想定されるため、利用ケースに応じた適切なレベルを選定することが望ましい。また、ISO/IEC 15408 では評価保証レベルが定義されており、レベルの値が大きいほど認証試験においてドキュメントや実装内容、開発環境などのより広い範囲をより厳密に評価したことを表している。

項目 E.07 は CKMS で採用する暗号モジュールに対して、そのセキュリティポリシーについて実装形態、セキュリティ機能の完全性保護メカニズム、暗号鍵の保護メカニズム、第三者認証の観点でそれぞれ明確化することを求めている。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

HSM 製品のセキュリティポリシーを確認し、CKMS セキュリティポリシーと整合したモジュール内での鍵管理、鍵保護、暗号処理の各機能を備えたデバイスを採用するものとする。第三者試験として FIPS 140-2/-3 レベル 3 の認証取得を要件とし、本プライベート CA の新規構築時及びリプレイス時に FIPS 140 認証が有効な製品を採用するものとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.07	a) HSM はハードウェアによって暗号機能を実装した暗号モジュールである。本 CKMS のセキュリティポリシーが定める各種の暗号アルゴリズムや乱数生成機能を備えた HSM 製品を選定する。 b) 本プライベート CA システムで利用する HSM は、暗号アルゴリズムや乱数生成機能の正常動作を検査するセルフテスト機能を備える。HSM の電源投入時や暗号機能の動作前、あるいは要求したタイミングでセルフテスト機能を実行する。 c) 本プライベート CA システムで利用する HSM は暗号鍵を保護するための物理的メカニズムとして、物理的なタンパー攻撃の検知機能と検知した場合のタンパー応答機能を備える。さらに、サイドチャネル攻撃や故障利用攻撃などの非侵襲攻撃に対する緩和策を備えている。論理的メカニズムとして、アクセス制御機能に基づいたエンティティの認証・認可の機能、さらに鍵情報を外部と入出力する場合に暗号学的保護によって解読や改ざんを防止する機能を備えている。 d) 第三者試験として FIPS 140-2/-3 レベル 3 の認証取得を要件とする。本プライベート CA の新規構築時及びリプレイス時に、FIPS 140 認証が有効な HSM 製品を採用する。

② 鍵情報の暗号モジュールへの入出力のための機能及び制限を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.08	FR6.58	CKMS 設計は、どのように、どのような状況で鍵情報（暗号鍵及びメタデータ）が暗号モジュールに入力されるか、入力される形式、及び入力に用いられる手段を明記しなければならない。	6.4.19 節
E.09	FR6.59	CKMS 設計は、（必要ならば）どのように入力された鍵とメタデータの完全性及び機密性が入力時に保護され検証されるかを明記しなければならない。	6.4.19 節
E.10	FR6.60	CKMS 設計は、どのように、どのような状況で鍵情報（暗号鍵及びメタデータ）が暗号モジュールから出力されるか、及び出力される形式を明記しなければならない。	6.4.20 節
E.11	FR6.61	CKMS 設計は、どのように出力された鍵とメタデータの機密性及び完全性が暗号モジュールの外部で保護されるかを明記しなければならない。	6.4.20 節

E.12	FR6.94	CKMS 設計は、平文での対称鍵又はプライベート鍵が暗号モジュールに入力又は出力される状況を明記しなければならない。	6.7.2 節
E.13	FR6.62	プライベート鍵、対称鍵、又は機密のメタデータが暗号モジュールから平文形式で出力される場合、CKMS 設計は、鍵情報（暗号鍵及びメタデータ）が提供される前に、呼び出しエンティティを認証するかどうか、及びどのように認証するかを明記しなければならない。	6.4.20 節
E.14	FR6.95	いかなる暗号モジュールにおいても、平文での対称鍵又は平文のプライベート鍵が入力又は出力される場合には、CKMS 設計は、平文鍵がどのように暗号モジュールの外部で保護され、制御されるかを明記しなければならない。	6.7.2 節
E.15	FR6.96	いかなる暗号モジュールにおいても、平文での対称鍵又は平文のプライベート鍵が入力又は出力される場合には、CKMS 設計は、そのような動作がどのように監査されるかを明記しなければならない。	6.7.2 節

解説・考慮点

暗号モジュールは、平文形式の暗号鍵への物理的保護を提供し、平文形式のまま暗号鍵が露出しないようにしている。このため、人間が平文形式の対称鍵又はプライベート鍵を見る必要が全くない暗号モジュールを使用する CKMS は、より透過的でよりセキュアである。また、暗号モジュールから出力される場合には、出力前に暗号鍵に暗号学的保護が適切に適用されなければならない。

CKMS の設計にあたって、項目 E.08 は暗号モジュールに鍵情報を入力するための条件を明確化することを、E.09 は入力される鍵情報の完全性と機密性を保護するための方法を明確化することを要求したものである。一方、E.10 は暗号モジュールから鍵情報を出力するための条件を明確化することを、E.11 は出力される鍵情報の完全性と機密性を保護するための方法を明確化することを要求したものである。
E.12～E.15 は対称鍵やプライベート鍵などが平文形式で入出力される場合の要求事項を明確化することを求めたものである。E.12 は平文形式で対称鍵やプライベート鍵などの入出力を行うための条件を、E.13 はエンティティ認証の手法を、E.14 は暗号モジュール外での保護方法を、E.15 は監査方法をそれぞれ明確化することを求めたものである。

上記のように、一般に暗号モジュールは、平文形式の暗号鍵への物理的保護を提供し、平文形式のまま暗号鍵が外部に露出しないようにしている。さらに、暗号モジュールは、その内部で対称鍵やプライベート鍵と公開鍵のペアを生成する機能を備えている。従って、暗号モジュールの外部に暗号鍵の出力を一切行わないように運用するのがよりセキュアといえるが、実際には暗号モジュールの故障対策や置き換えのために暗号鍵を外部に出力する機能や、外部から暗号鍵を入力する機能が必要となることもある。

このように、暗号モジュールにおける入力機能は、ひとつ又は複数の暗号鍵及び関連付けられたメタデータを、実使用の準備のために暗号モジュールに入力するために使用する。また、暗号モジュールにおける出力機能は、ひとつ又は複数の暗号鍵及び関連付けられたメタデータを、外部での使用もしくは保管（バックアップやアーカイブ）のために暗号モジュールから出力するために使用する。ここで、暗号モジュールから暗号鍵やメタデータを出力する場合、出力する鍵情報に対して暗号学的保護である機密性及び完全性の保護処理が適切に適用される必要がある。

外部にある鍵情報の保護は暗号学的保護により論理的に実現されるが、その暗号学的保護は元となる鍵暗号化鍵や鍵ラッピング鍵の管理状況に依存することには留意すべきである。また、レガシーシステムへの対応など何らかの理由により、平文状態で暗号鍵の入出力を許容する場合には、外部での平文状態での暗号鍵の保護がどのように実現されるかによって CKMS のセキュリティレベルに大きな影響が及ぶ可能性がある。E.12～E.15 に暗号モジュール外部への平文状態での暗号鍵の入出力に関わる FR が並んでいるのは、このためである。暗号モジュールで利用する対称鍵やプライベート鍵などの機密性保護を要する鍵に対して平文形式での入出力を行わない場合には、E.12～E.15 は検討対象外としてよい。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.08	本 CKMS では、CA プライベート鍵は原則として利用する HSM 内で生成するが、HSM のリプレイスや障害発生に備えて、HSM は鍵のバックアップ・アンド・リストアの機能を有している。鍵情報の入力は USB インタフェースに専用のストレージモジュールを接続して行われ、入力形式は HSM ベンダの独自仕様である。鍵情報の入力は、HSM のリプレイス時に、従来の HSM で使用していたプライベート鍵を新規の HSM でリストアして使用する場合などに行われる。
E.09	利用する HSM に外部から鍵情報を入力する場合、PKCS #11 API によって暗号化されて行われ、外部もしくは入力の過程における鍵情報の解読や改ざんから保護される。入力された鍵情報の完全性は HSM 内で検証する。
E.10	利用する HSM は、内部の CA プライベート鍵を含む鍵情報を外部に出力する機能を備える。本機能は鍵情報のバックアップ・アンド・リストアのためのものである。出力形式を含めて、バックアップ・アンド・リストア機能の詳細は HSM ベンダの独自仕様である。
E.11	利用する HSM から鍵情報を外部に出力する場合、PKCS #11 API によって暗号化されて行われ、外部での鍵情報の解読や改ざんから保護される。外部での保護レベルは、外部出力時に利用された鍵ラッピング鍵（key wrapping key）の管理に依存する。
E.12	本 CKMS で利用する HSM は署名用プライベート鍵の平文状態での入出力を行えないように設定されている。そのため、対象外である。

E.13	本 CKMS で利用する HSM は平文状態での署名用プライベート鍵、及び機密のメタデータの出力を行えないように設定されている。そのため、対象外である。
E.14	本 CKMS で利用する HSM は署名用プライベート鍵の平文状態での入出力を行えないように設定されている。そのため、対象外である。
E.15	本 CKMS で利用する HSM は署名用プライベート鍵の平文状態での入出力を行えないように設定されている。そのため、対象外である。

③ 暗号モジュールの危殆化に対する BCP 対策を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.16	FR6.109	CKMS 設計は、暗号モジュールの中身への物理的及び論理的アクセスがどのように認可されたエンティティに制限されるかを明記しなければならない。	6.8.4 節
E.17	FR6.110	CKMS 設計は、暗号モジュールの危殆化からの回復のために使用される方法を明記しなければならない。	6.8.4 節
E.18	FR6.111	CKMS 設計は、どの非侵襲攻撃がシステムで使用される暗号モジュールによって軽減されるかを記載し、どのように軽減が実行されるかの記載を提供しなければならない。	6.8.4 節
E.19	FR6.112	CKMS 設計は、非侵襲攻撃に脆弱であるあらゆる暗号モジュールを明記しなければならない。	6.8.4 節
E.20	FR6.113	CKMS 設計は、可能性のある非侵襲攻撃によって起きる脆弱性を受け入れる原則を明記しなければならない。	6.8.4 節

解説・考慮点

暗号モジュールの危殆化は、当該暗号モジュールに保持されている対称鍵及びプライベート鍵の危殆化の可能性を伴う。結果として、機密性の喪失、完全性の喪失、又は認証能力の喪失につながり得る。

暗号モジュールの危殆化の原因には、暗号モジュール内の暗号鍵へ直接アクセスする物理的手段、又は暗号モジュール内の暗号鍵についての知識を何らかの外部からの操作によって得る非侵襲的手段がある。
物理的手段に対する保護を提供するためには、認可されないアクセスが許可されない場所、又は認可されないアクセスが速やかに検出されるような仕組みがあるところで暗号モジュールは運用されるべきである。非侵襲的手段に対する保護を提供するためには、暗号モジュールの使用を信頼されるユーザに制限する、又は（特定の）非侵襲的手段による攻撃を防止するように設計された暗号モジュールを利用すべきである。

実際に暗号モジュールの危殆化又は危殆化の疑いがあった場合には、通常運用に戻る前に当該暗号モジュールをセキュア状態に再確立する必要がある。特に暗号モジュールの修理又は交換を行った場合には、セキュリティ状態の確認とともに機能確認のためのテストも行わなければならない。

CKMS の設計にあたって、項目 E.16 はエンティティ認証の手法を明確化することを、E.17 は危殆化が検知された後にどのような BCP[10]対策を行うかを明確化することを要求したものである。
E.18 は非侵襲的手段に対する事前対策を明確化することを、E.19 及び E.20 は対策の限界を明確化することを要求したものである。これは、あらゆる非侵襲的手段に対して完璧な対策を行うことはコスト的にも技術的にもほぼ不可能であることに原因がある。つまり、非侵襲的手段の種類によって、事前対策による被害軽減策がとられている部分と、残存リスクとして対策を取らない（あるいは不十分な対策である）部分とに予め整理しておくことに主眼がある。

上記のように、暗号モジュールの危殆化により、当該暗号モジュールに保持されている対称鍵及びプライベート鍵が危殆化する。その結果、CKMS が実現すべき機密性の喪失、完全性の喪失、又は認証能力の喪失につながり得る。

暗号モジュールの危殆化の原因には、暗号モジュール内の暗号鍵へ直接アクセスする物理的攻撃、及び、暗号モジュールの外部からの操作によって得られる暗号鍵に関わる情報を利用して最終的に暗号鍵を推定する非侵襲攻撃などがある。物理的攻撃は暗号モジュールの筐体などの囲いを除去する作業を伴うが、非侵襲攻撃はそうした作業を必要としない。ここで、非侵襲攻撃の具体例にはサイドチャネル攻撃と故障利用攻撃がある。代表的なサイドチャネル攻撃としては、暗号モジュールにおける暗号処理中の消費電力の変動を観測する電力解析と暗号処理の処理時間の変動を観測するタイミング攻撃などが知られている。また、故障利用攻撃としては、レーザ照射や電源グリッチ挿入などの外乱を与えることで暗号処理の結果を誤らせたデータを得て、正常な処理データとの差分を利用する差分故障解析などが知られている。緩和策が全くとられていない場合に、一度の処理データの観測だけで秘密鍵が判明するような攻撃もある。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.16	本 CKMS を収めた施設にはアクセス制御があり、利用する HSM への物理的攻撃及び非侵襲攻撃を実行可能な主体はアクセス制御を通過した主体に限定される。
E.17	まず、危殆化の要因が、HSM の物理的防御に関わるものか非侵襲攻撃への防御に関わるものかを明確にする。さらに、リスク評価を行って緩和策や対策の必要性、及び鍵更新の

[10] BCP: Business Continuity Plan（事業継続計画）。

	必要性を判断する。緩和策には、HSM を収容するエリアへのアクセスコントロールの見直しによって、アクセス可能な主体をより信頼できるメンバに限定すること等がある。対策は、より強固な防御機能を備えた HSM への置き換えとなる。ファームウェア更新によって防御を強化できる場合もある。なお、リスク評価の結果、鍵更新を必要とする場合は HSM 内で新たな鍵の生成を行い、それ以外はバックアップしておいた鍵をリストアする。
E.18	利用する HSM は内部に実装されている暗号処理機能に対して、サイドチャネル攻撃（DPA 及び SPA と呼ばれる電力解析）や故障利用攻撃（DFA）といった非侵襲攻撃の耐性を備えている。証明書及び CRL に付与する ECDSA 署名の生成処理において耐性がある。耐性を実現する具体的な緩和策は HSM ベンダのプロプライエタリ情報である。
E.19	利用する HSM は非侵襲攻撃に対する耐性を有している。ただし、非侵襲攻撃の種類や進化によって緩和策が有効でない場合もあり、HSM ベンダが提供する注意喚起情報を監視する必要がある。
E.20	利用する HSM は非侵襲攻撃への耐性を備えるが、非侵襲攻撃の進化によって現行の緩和策を破る手法が発見されるリスクはある。そうした場合でも多層防御として本 CKMS を収めた施設へのアクセス制御やアカウント権限管理等があり、非侵襲攻撃を実行可能な主体は多層防御を通過した主体に限定される。そのため、本 CKMS 全体でのリスクは低減される。また、非侵襲攻撃による HSM の危殆化が判明した場合は、対象となる非侵襲攻撃への耐性を強化した暗号処理 FW への更新やより強固な緩和策を備えた HSM への置き換えによってリスクの低減を図る。

3.1.3 人間による入力のコントロール

① 鍵情報の入力を人間に求める場合の要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.21	FR6.97	それぞれの鍵とメタデータの管理機能に対し、CKMS 設計は、全ての人間による入力パラメタ、そのフォーマット、及び入力が行われないときに CKMS が取るアクションを明記しなければならない。	6.7.3 節

解説・考慮点

暗号鍵又は機微なメタデータの入力を人間に求める場合、それらの入力の正確さ（場合によってはセキュリティも）が担当する人間に依存する。また、必要な時に人間が適切に動いてくれるかどうかもわからない。一方、必要なときに CKMS が自動的に実行できるのであれば、そのシステムはユーザにとってより透過的になり、よりセキュアになる可能性がある。
項目 E.21 は、CKMS の設計にあたって、鍵情報の入力を人間に求める場合の要求事項を明確化することを求めたものである。

上記の「暗号鍵及び機微なメタデータ」とは、暗号鍵そのもの、鍵情報の生成に利用可能な情報、及び暗号鍵に紐づくメタ情報とする。暗号モジュール自体は堅牢であっても、人間による暗号鍵及び機微なメタデータの入力が存在する場合、その入力の正確さの課題やセキュリティ面の問題があり、リスク要因となる。人間による入力の具体例としては、人間の操作をエントロピー源として、鍵生成時のシードの一部を入力するケースが挙げられる。

人間による鍵情報の入力としては様々なケースがあるが、ここでは人間の関与によって入力あるいは生成された鍵情報の一部に影響が及ぶかどうか、あるいは、入力処理のほとんどが機械化（自動化）して行われ人間が関与する処理はごく一部に過ぎないかどうか（例えば、鍵情報の記憶された媒体をアタッチする操作のみであれば人間の関与は少ない）、によって項目 E.21 の該否を判断するとよい。

項目 E.21 は人間による鍵情報の入力を求める場合の入力パラメータ・フォーマット、入力が行われないときの CKMS のふるまいを明確にすることを求めている。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

一般に HSM 製品の利用においては項目 E.21 に該当するケースがないことが望ましい。しかしながら、HSM 製品を利用するユースケースの中には、マルチパーティコントロールによって分割した鍵コンポーネントを HSM に入力する際に、人間がキーボードから鍵コンポーネントを入力する運用事例がある。そのような運用を行う場合には項目 E.21 を検討対象とすべきである。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.21	本プライベート CA では該当するケースはないので対象外。

3.1.4 マルチパーティコントロール

① 暗号鍵管理機能を実行するために複数のエンティティの協力を必要とする場合の概要を明確化しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.22	FR6.98	CKMS 設計は、マルチパーティコントロール（multiparty control）を要求する全ての機能を明記し、それぞれの機能に対して k と n を規定しなければならない。	6.7.4 節
E.23	FR6.99	それぞれのマルチパーティ機能に対して、CKMS 設計は、なぜ n 個中任意の k 個のエンティティで望む機能を有効にできるが k-1	6.7.4 節

		個のエンティティでは有効にできないのかを示すあらゆる既知の論拠（論理、数学）を引用又は明記しなければならない。	

解説・考慮点

ある種の暗号鍵管理機能を実行するために複数のエンティティの協力を必要とする場合に利用する一手法であり、当該機能を実行する前に、n 人中 k 人のエンティティが ACS で認証・認可されることを要求する。暗号鍵管理機能の中でも高度に機微な機能が対象となる。
CKMS の設計にあたって、項目 E.22 はマルチパーティコントロールで管理される機能及び利用条件を明確化することを、E.23 は採用する方式の安全性を明確化することを要求したものである。

マルチパーティコントロールは複数のエンティティの協力によって、ある種の処理を実行可能とする機能である。ここで、n エンティティのうち任意の k エンティティ以上の協力が得られれば所定の処理を実行できるが、k エンティティ未満の協力下では処理を実行できない。

マルチパーティコントロールは、機微な権限の取得や高度な管理を要する鍵情報の分散管理に利用されることが多い。例えば、管理者権限のエンティティ認証・認可やマスター鍵/鍵導出鍵のバックアップ・アンド・リストアに利用される。ここで暗号鍵自体をマルチパーティコントロールによって分割する場合は次の②の項目 E.24 と E.25 にも該当する。

なお、一般にマルチパーティコントロールの実現メカニズムは、対象機器やシステムを構築するベンダのプロプライエタリ情報である。従って、マルチパーティ機能の原理説明に関わる項目 E.23 を CKMS 要件とするかどうかはシステム設計者や調達者が判断し、CKMS 要件とする場合はベンダに説明を要求することとなる。その際、メカニズムの説明は原理を説明する論文へのポインタ情報程度でもよい。

マルチパーティコントロールで管理される機能がない場合には本節の FR は対象外である。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

本トイモデルにおけるマルチパーティコントロールは HSM により提供される。HSM により提供されるマルチパーティコントロールの実現メカニズムの内容は、一般に HSM ベンダのプロプライエタリ情報である。ここでは HSM ベンダによる説明が得られたケースを想定して E.23 の例を記載した。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.22	本 CKMS では利用する HSM の管理者権限の取得にマルチパーティコントロールを使用

	する。その際、CKMS 管理担当の中で HSM 管理者として予め登録した 3 エンティティのうち 2 エンティティの認証を必要とする。 本 HSM 内の鍵情報のバックアップ・アンド・リストアを実施する場合には HSM の管理者権限を必要とするように設定されている。
E.23	本 CKMS で利用する HSM では次のメカニズムによってマルチパーティコントロールを実現している。予め Shamir の秘密分散（2 out of 3）により、管理者権限取得のための認証情報が 3 つの share に分割され、異なる USB ドングルに格納される。各エンティティは USB ドングルを順次接続した上で、USB ドングルのアクティベート PIN を入力することにより、share が HSM に転送され、HSM 内で集まった share 群から認証情報が復元される。

② 暗号鍵を鍵分割する場合の概要を明確化しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.24	FR6.100	CKMS 設計は、鍵分割技術を使用して管理される全ての鍵を明記しなければならず、またそれぞれの技術に対して n と k を明記しなければならない。	6.7.5 節
E.25	FR6.101	使用しているそれぞれの（k, n）鍵分割技術に対して、CKMS 設計は、鍵分割がどのように行われ、なぜ n 個中任意の k 個の分割鍵で鍵を構成できるが k-1 個の分割鍵では鍵に関する情報を何ら提供しないのかを示すあらゆる既知の論拠（論理、数学）を明記しなければならない。	6.7.5 節

解説・考慮点

CKMS の設計にあたって、項目 E.24 は鍵分割で管理される対象の暗号鍵及び鍵分割の利用条件を明確化することを、E.25 は採用する方式の安全性を明確化することを要求したものである。

本節の①に記載したように、マルチパーティコントロールの一形態として鍵分割がある。n 個の分割鍵（share）のそれぞれが n 人のエンティティに割り当てられ、そのうち任意の k 人のエンティティが協力しない限り元の暗号鍵が構成できない仕組みである。ここで k 人未満のエンティティの協力（すなわち k-1 個以下の share）では元の暗号鍵の情報は一切得られない（すなわち share が全くない状態と情報理論的に情報量が変わらない）ことが特徴である。

鍵分割によるマルチパーティコントロールが効果を発揮するためには分割鍵の管理を適正に行うことが重要である。例えば、複数の分割鍵を一人のエンティティが利用できるような状況にあってはならないし、分割鍵へのアクセス権限を持つエンティティ間で結託が行われるような状況があってはならない。

鍵分割は、多くの他の暗号鍵を保護し、その危殆化が深刻な悪影響をもたらすようなマスター鍵/鍵導出鍵のバックアップ・アンド・リストアを実施するために、分割鍵を暗号モジュールに入出力する場面で使用されることが多い。

本節の①と同様に、一般に鍵分割の実現メカニズムは、対象機器やシステムを構築するベンダのプロプライエタリ情報である。従って、鍵分割機能の原理説明に関わる項目 E.25 を CKMS 要件とするかどうかはシステム設計者や調達者が判断し、CKMS 要件とする場合はベンダに説明を要求することとなる。その際、メカニズムの説明は原理を説明する論文へのポインタ情報程度でもよい。

鍵分割技術を利用して管理される暗号鍵がない場合には検討対象外である。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.24	本 CKMS で利用する HSM の管理者権限の取得にマルチパーティコントロールを使用する。その際、E.23 に記載のように認証情報の分割に暗号鍵分割に相当する手法を利用するが、分割対象は暗号鍵ではないため、E.24 は対象外である。
E.25	E.24 と同じ理由により、E.25 は対象外である。

3.2 セキュリティ評価・試験

解説・考慮点

本節は、SP 800-130 の 9.1 節から 9.7 節に記載されている事項について解説したものである。

セキュリティ評価・試験におけるテストスイートに合格することの価値は、選択したテストケースの包括性及び代表性に直接関連する。一方、全ての可能性の組み合わせ数よりはるかに少ない有限個のケースに限定されるため、デバイス又はシステムが全てのケースにおいて正しい又はセキュアであることを保証しないことに留意されたい。
調達者又はユーザは、どのテスト結果を必要とするのかを事前に決める必要がある。さらに提供されたテスト結果をレビューし受入可能かどうかを判断するのか、事前に満たすべき条件を指示しておくのか決めておくべきである。

本節の各テストの概要は SP 800-130 の該当節も参照されたい。SP 800-130 の該当節の記載では、ここでのテスト対象が CKMS 全体なのか、CKMS の構成要素であるデバイスなのかがあいまいであるが、原則としていずれも CKMS 全体を対象としたテスト項目と捉えるのが良い。ただ

し、テスト項目の中でデバイスでのテストを実施すれば十分と判断できるものや、デバイスでのテスト結果をベンダに要求すべきと判断するものはデバイスでのテスト項目として実施する。

暗号モジュールに相当するデバイスを対象にテストを実施する場合には、本節のテスト項目のうち①ベンダテスト、③機能テスト及びセキュリティテスト、④環境テスト、⑤セルフチェックテスト、⑦第三者テストを選定するとよい。なお、一般に、FIPS 140-2/-3 や ISO/IEC 15408 などのセキュリティ認証を取得済の暗号モジュールであれば、これらのテストは認証取得時の試験によって満たしていると判断するのも妥当である。

① ベンダテストの実施概要を明確化しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.26	FR9.1	CKMS 設計は、システムで実行され合格した非プロプライエタリベンダテストを明記しなければならない。	9.1 節

解説・考慮点

ベンダが自ら実施するテストである。テストの技術及び仕様は、ベンダによるプロプライエタリ情報と見なされることが多く、一般に公開されない。
項目 E.26 は、CKMS の設計にあたって、調達者又はユーザがレビュー可能なベンダテストの実施概要を明確化することを要求したものである。

調達側としては、非プロプライエタリな情報で提供可能なテスト結果の有無をベンダに確認することが本 FR の対応となる。なお、FIPS 140-2/-3 や ISO/IEC 15408 などのセキュリティ認証を取得した製品であれば認証時の試験内容で十分と判断することもできる。一方、セキュリティ認証を取得していない製品を採用する場合や、特に確認の必要なテスト内容がある場合には、ベンダに相談すべきである。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.26	利用する HSM は FIPS 140-2/-3 レベル 3 のセキュリティ認証を取得しており、認証時の試験によって本件に相当するテストはカバーされていると考え、対象外とする。

② 相互運用性テストの実施概要を明確化しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.27	FR9.3	CKMS が他のシステムとの相互運用性を主張する場合、CKMS 設計は、その主張を検証するために実行し合格したテストを明記しなければならない。	9.3 節
E.28	FR9.4	CKMS が他のシステムとの相互運用性を主張する場合、CKMS 設計は、相互運用性に必要な、あらゆる構成設定（configuration settings）を明記しなければならない。	9.3 節

解説・考慮点

2 つ以上のデバイスを相互接続し、互いに運用することができるかどうかのテストである。ただし、個々のデバイスの内部機能自体をテストしているわけではないので、その機能が正しく動作することを検証しているとは限らない。テスト対象デバイスと保証ベースラインデバイスが異なる組織によって独立に設計・実装されていれば、このテストはより信用できる。
CKMS の設計にあたって、項目 E.27 は、相互運用性テストの実施概要を明確化することを、E.28 は相互運用するための要求条件を明確化することを求めたものである。

相互運用性テストの一般的な形式は、対向接続試験である。SP 800-130 には、テスト対象装置（device under test）と保証ベースライン装置（assured baseline device）との相互運用をテストすることにより、テスト対象装置の正常動作を確認するケースが示されている。

暗号モジュールなどのデバイスレベルのテストは、それを利用する上位のシステムとの間で API の機能テストを実施することが一般に行われており、次の③の機能テストと捉えることができる。

本ガイダンスにおいて相互運用性テストとは、CKMS レベルで他の CKMS との連携をテストするケースや保証ベースライン装置のようなゴールデンサンプル（ゴールデンデバイスと呼ばれることもある）との接続試験を行うものとして解釈することとする。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

本 CKMS の直接の連携先は IoT 製品向け証明書管理端末に限定される。ここで証明書管理端末向けに提供するサービスは IoT 製品向けの証明書発行及び証明書の失効に関わる CRL の生成である。これらのサービスに関わるプロトコルは単純であるため、本 CKMS の提供するサービス自体は機能レベルのテストとして実施することができ、本ガイダンスでの解釈に該当する相互運用性テストを実施する必要はない。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.27	本 CKMS では相互運用性テストに相当する特段のテストを実施する必要はなく、非該当とする。

E.28	同上

③ 機能テスト及びセキュリティテストの実施概要を明確化しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.29	FR9.7	CKMS 設計は、システムで実行された機能テスト及びセキュリティテスト、並びにそのテスト結果を明記しなければならない。	9.6 節

解説・考慮点

機能テストとはある機能の実装が正しく動作することを検証するテストであり、セキュリティテストとはある機能の実装がセキュアに機能することを検証するテストである。このため、暗号アルゴリズムの実装が正しく機能する（機能テストに合格）一方で、暗号処理中の電力消費の変動等が暗号鍵の危殆化につながり得ると判定（セキュリティテストに不合格）することがある。
ペネトレーションテストは特別な種類のセキュリティテストである。ペネトレーションテストのエキスパートチームが攻撃シナリオを開発して、ペネトレーション成功のリスクを評価する。初期運用開始前及び大規模変更後の運用再開前にペネトレーションを実施し、発見されたあらゆる課題に事前に対処すべきである。なお、スコープには、人的、設備及び手続きを含むべきである。
項目 E.29 は、CKMS の設計にあたって、機能テスト及びセキュリティテストの実施概要を明確化することを要求したものである。

機能テストは、CKMS の適切な動作を保証するために様々なレベルで実施される。これには、個々のデバイスやコンポーネントなど、CKMS の構成要素レベルでのテスト及び CKMS 全体が提供する機能レベルでのテストが含まれる。これらのテストは、システム開発における単体試験、結合試験及び総合試験に対応しており、個々の機能からシステム全体の統合的な動作までを評価することを目的とする。

セキュリティテストは、システムや実装の堅牢性を評価し、機密性・完全性・可用性といったセキュリティ要件を満たしているかを確認するための重要なプロセスである。このテストでは、暗号鍵やパスワードなどの秘匿すべき情報が、権限のないエンティティから適切に保護されているかを検証する。また、暗号鍵、プログラムコード、権限設定情報など、保護すべきデータが改ざんから適切に保護されているかを評価する。

セキュリティテストを実施する際には、どのような攻撃を想定するか、利用する解析手法やツールを明確に定義することが求められる。攻撃者のスキルレベルは、初歩的な IT エンジニアから国家レベルの専門家集団まで幅広く存在し、保護対象の資産価値に応じて想定する攻撃者像を設定する必要がある。この攻撃者像に基づいて、要求されるセキュリティレベルが大きく変化する。

ペネトレーションテストは、こうした攻撃者を想定したセキュリティテストの代表的な方法であり、専門家チームが対象システムに実際に侵入を試み、攻撃が成功する可能性を評価する。こ

のプロセスでは、攻撃者のスキル、使用するツール、攻撃に要する時間、攻撃手順といった要素を定量的に採点し、防御能力をスコアリングする場合もある。外部機関にペネトレーションテストを依頼する場合、コストや時間がかかることが多いため、設計情報を活用した簡易的な内部テストで代替するケースもある。

また、ペネトレーションテスト以外にも脆弱性診断の様々な手法が存在する。例えば、静的解析ツールを使用してプログラムコードの脆弱性を検出する方法や、ファジングを用いて異常なデータ入力への耐性を評価する方法がある。ファジングは、通信プロトコルだけでなく、ファイル形式、API（アプリケーションプログラミングインタフェース）、CLI（コマンドラインインタフェース）、OS やデバイスドライバなど、広範な対象に適用される手法である。これにより、入力検証の不備やエラーハンドリングの欠陥を検出できる。

一方で､暗号アルゴリズムそのものや､暗号を利用したセキュリティプロトコル全体の解析は、セキュリティテストには通常含まれない。これらの要素については、業界で推奨されている方式であるか、選定時に既知の脆弱性が報告されていないかを確認することで対応する。

調達するデバイスの機能テスト及びセキュリティテストについては、ベンダに提供可能な情報を確認することが項目 E.29 の対応となる。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.29	● 利用する HSM の機能試験の結果及び実装の堅牢性に関わるテスト結果について、ベンダに提供可能な情報を確認する。 ● 本プライベート CA のシステムレベルの機能テストとして、結合試験、総合試験の結果を報告書で確認する。 ● 本プライベート CA のセキュリティテストとして、プログラムコードの脆弱性診断を含むセキュリティテスト仕様書を作成し、内部で実施したテスト結果を報告書にまとめる。

④ 環境テストの実施概要を明確化しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.30	FR9.8	CKMS 設計は、CKMS が使用される設計上の環境条件を明記しなければならない。	9.7 節
E.31	FR9.9	CKMS 設計は、CKMS デバイスで実行された環境テストの結果を、設計上の条件を超えたストレスをデバイスに与えた時の全てのテストの結果も含めて、明記しなければならない。	9.7 節

解説・考慮点

デバイスやシステムに対して特定の利用環境（例えば、温度範囲及び電圧範囲）を仮定することが多い。この場合、当該デバイスやシステムはその利用環境用に構築され、決められた範囲内でのみテストされる。もし範囲外の利用環境で当該デバイスやシステムが使用されると、セキュアな運用が失われる可能性がある。 CKMS の設計にあたって、項目 E.30 は設計上の利用環境条件を明確化することを、E.31 は環境テストの実施概要を明確化することを要求したものである。なお、E.31 では、設計上の利用範囲外の環境でのテストを実施した場合にはその結果も含めることを要求していることに留意されたい。ただし、利用範囲外の環境でのテスト結果が悪かったとしても、それ自体に問題があるわけではない。

環境テストは、温度や供給電圧などの環境条件を変えたときの CKMS の挙動を検査するものである。一般に環境条件は設計上の動作可能範囲として定められており、その範囲で正常な動作を行うこと、セキュアに機能することが確認できれば良い。

項目 E.31 は設計上の環境条件を超えた場合のテスト結果に関わる FR である。FIPS140-3 レベル 3 の物理セキュリティには環境故障保護(Environment Failure Protection)の機構を備えるか、環境故障試験(Environment Failure Test)に合格することの要件があり、これらが相当する。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.30	本 CKMS を構成する各デバイスの仕様上の環境条件に基づいて、CKMS として設計上の環境条件を定める。
E.31	利用する HSM は FIPS 140-3 レベル 3 の認証を取得した製品であり、セキュリティポリシーによって HSM の環境条件を超えた場合の保護機構もしくは試験結果を確認する。

⑤ セルフチェックテストの概要を明確化しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.32	FR9.5	CKMS 設計は、設計者によって作成及び実装された全ての自己テスト、及びそれが正しい動作を検証する対象の CKMS 機能を明記しなければならない。	9.4 節

解説・考慮点

セルフチェックテストとは、完全性及びセキュリティ障害に対してデバイスが自分自身を定期的にテストする機能である。

項目 E.32 は、CKMS の設計にあたって、セルフチェックテストの概要を明確化することを要求したものである。

セルフチェックテストの例として、FIPS 140-2/-3 において要件としている自己テスト機能があり、動作前自己テストと条件自己テストを要求している。動作前自己テストでは電源投入時やリセット直後にファームウェア完全性のテストを実施する。条件自己テストでは暗号アルゴリズムを最初に利用する前に暗号アルゴリズム自己テストの実施が要求される他に、公開鍵・プライベート鍵ペアを生成した際に鍵ペア整合性テストの実施などが要求される。

汎用的な PC において実施されるブート時に起動するソフトウェアの完全性を検査するセキュアブート処理もセルフチェックテストに該当する。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.32	● 本プライベート CA のサーバはセキュアブート処理により、電源投入時やリセット後にロードするファームウェアやプログラムコードを検証する。 ● 採用する HSM のセキュリティポリシーを参照して、搭載されている自己テストの内容と各自己テストが実施される条件を確認する。

⑥ スケーラビリティテストの実施概要を明確化しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.33	FR9.6	CKMS 設計は、今までにシステムで実行された全てのスケーラビリティ分析及びテストを明記しなければならない。	9.5 節

解説・考慮点

プロセスが増大する負荷に適応してデバイスやシステムの処理能力を拡大する必要があるため、与えられた時間内で処理するトランザクション数又は取り扱うユーザ数が劇的に増加したときにデバイスやシステムがどのように反応するかを見極めるために行うテストである。デバイスやシステムが完全に運用される前にスケーラビリティの問題を認識して、必要な負荷軽減策を検討するために行われる。 項目 E.33 は、CKMS の設計にあたって、スケーラビリティテストの概要を明確化することを要求したものである。

項目 E.33 は CKMS のシステムレベルのスケーラビリティテストに関わる FR である。一般に負荷テストと呼ばれる。CKMS の負荷が増大した場合にどこがボトルネックとなるか、高性能なデバイスへの更新やデバイスの増設で対処可能かを予めテストすることを求めている。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.33	本プライベート CA を構成するサーバや HSM、ネットワーク機器などをモジュール化した設計とし、負荷の分散を可能としておく。ロードテストやストレステストを実施し、ピーク性能の確認や負荷の集中による障害発生時の回復作業を確認する。

⑦ 第三者テストの概要を明確化しなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.34	FR9.2	CKMS 設計は、CKMS 又はデバイスが今までに合格した全ての第三者テストプログラムを明記しなければならない。	9.2 節

解説・考慮点

ベンダが自身のテスト手順のなかで欠陥を見逃していないことの信頼性を提供するために第三者によって行われるテストのことである。
項目 E.34 は、CKMS の設計にあたって、第三者テストの概要を明確化することを要求したものである。

暗号標準や推奨事項への製品適合の検証プログラムとして、ISO/IEC 15408 認証、FIPS 140-2/-3 認証（CMVP 認証）、CAVP 認証等が代表例である。これらの認証を取得した製品は CCRA や NIST のサイトで検索可能である。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.34	本プライベート CA システムで採用する HSM は FIPS 140-2/-3 レベル 3 認証を取得した製品である。ベンダに対して認証書やセキュリティポリシーの提供を依頼する。

3.3　暗号モジュールの障害時の BCP 対策

解説・考慮点

暗号モジュールには、セキュリティ機能、鍵情報（暗号鍵やメタデータ）など、CKMS のセキュリティを確保するための様々な情報が内包されている。このため、暗号モジュールが障害を起こすことは CKMS のセキュアな運用ができなくなることを意味する。本節では、暗号モジュールに障害が発生した場合の対策を取り扱う。

暗号モジュールの障害は CKMS の運用に影響を及ぼし、さらには CKMS がサービスを提供するシステムの機能不全や情報喪失につながるおそれがある。

① 暗号モジュール障害発生時における BCP 対策を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
E.35	FR10.8	CKMS 設計は、モジュールのエラー検知及び完全性検証のために、それぞれの暗号モジュールがどの自己テストを使用するかを明記しなければならない。	10.6 節
E.36	FR10.9	CKMS 設計は、それぞれの暗号モジュールがどのように検知したエラーに応答するかを明記しなければならない。	10.6 節
E.37	FR10.10	CKMS 設計は、障害が起こった暗号モジュールの修理又は交換の方策を明記しなければならない。	10.6 節

解説・考慮点

暗号モジュールは、ハードウェア、ソフトウェア又はファームウェアの障害を検知するために適切に組み込まれたテスト機能を備えるべきである。テストの結果、暗号モジュールがエラー状態にある間は、機微なデータが暗号モジュールから出力されるべきではない。

CKMS の設計にあたって、項目 E.35 は暗号モジュールの障害検知のために組み込まれたテストに関する概要について明確化することを、E36 は障害を検知した時に直ちに取るべき対応策を明確化することを、E.37 は復旧に向けて障害が検知された後にどのような BCP 対策を行うかを明確化することを要求したものである。
なお、E.37 については、E.17 と同様、通常運用に戻る前に当該暗号モジュールをセキュア状態に再確立する必要がある。エラーが回復可能なものであるならば、暗号モジュールを再起動した後、通常処理を続行する前に全てのパワーアップセルフチェックテストを実施してエラーが解消されたことを確認しなければならない。また、暗号モジュールの修理又は交換を行った場合には、セキュリティ状態の確認とともに機能確認のためのテストも行わなければならない。

暗号モジュールに対するセルフテスト（自己テスト）機能として FIPS 140-2/-3 では以下に示すことを要件としている。暗号モジュールの自己テストとして、動作前自己テスト及び条件自己テストがあり、暗号モジュールはこれらの自己テストを実行して成功か失敗かを判定する。自己テストに失敗した場合はエラー状態に遷移し、エラーインジケータが出力される。エラー状態では暗号処理やデータ出力を行うことはできない。

暗号モジュールが再起動をしてもセルフテストによるエラー状態から回復しない場合は暗号モジュールを交換することとなる。そうした状況に備えて、暗号モジュール内の鍵情報のバックアップを事前に取得しておくべきであり、交換後の暗号モジュールへの鍵情報のリストアやエラー状態となった暗号モジュールの廃棄処理などの手順を定めておくべきである。

本節の FR は上記のような FIPS 140-2/-3 の要件を満たす暗号モジュールを念頭にしたものと捉えると理解しやすい。

なお、暗号モジュールの汎用的な BCP 対策として、CKMS がサービスを提供するシステムの要件に合わせて、暗号モジュールの提供ベンダと適切な保守契約を結ぶことを推奨する。こうした保守契約がないと、ベンダから最新のファームウェアが提供されないことや暗号モジュールに不具合が発生した際の回復がベンダのサポートがないために実行できない等の事態が懸念されるためである。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

E.35	本プライベート CA システムで利用する HSM は FIPS 140-2/-3 レベル 3 認証を取得しており、実装されている暗号処理に関わるセルフテスト機能を備えている。セルフテストには動作前自己テストと条件自己テストの両テストを備えている。
E.36	本 HSM はセルフテストでエラーを検知した場合はエラー状態となり、エラーインジケータが出力される。エラーインジケータとして処理要求に対して応答不能コードが返される。エラー状態では暗号に関わるデータ出力や暗号処理を行えない。
E.37	● 本 HSM がエラーを出力した場合、HSM 管理者は HSM の再起動により、エラー解除を試みる。エラー状態から回復しない場合は、HSM の初期化、破壊などマニュアル記載の HSM 廃棄に関わる手続きを実施する。 ● 予め HSM 障害時の対応を想定して HSM 内の鍵情報のバックアップを作成しておく。HSM のエラーが回復しない場合は、新たな HSM をセットアップし、バックアップした鍵情報をリストアして、新たな HSM に交換する。 ● HSM ベンダと障害時の対応を踏まえた保守契約を締結しておき、障害発生時にベンダサポートを受けられるようにしておく。

4 暗号鍵管理システム（CKMS）のオペレーション対策

本章の目的・趣旨

本章は、「設計指針」の 9 章に記載されている要求事項（各節での灰色枠内で示している内容）について解説したものである。

CKMS 全体に対して、必要に応じて検討する項目（F.01～F.57）を集めている。ここには、主に以下のような検討項目を含んでいる。

- CKMS 全体に対する包括的なセキュリティ対策（物理的対策、マルウェア対策、脆弱性対策、侵入防御対策、システム監査など）をどうするか
- CKMS 全体のセキュリティアセスメントをどのように実施するか
- CKMS への危殆化・障害・災害発生時の BCP 対策をどのように準備するか

ここでの検討項目も「広義」の意味での暗号鍵管理に相当するものの一つであり、CKMS 全体を対象としている。個々の暗号鍵管理のためではなく、システムとしての暗号鍵管理が正常に機能するようにするための検討項目になっており、CKMS 全体のオペレーション対策や物理的な対策を含めた総合的な対応を対象としたシステム設計を行う場合に検討する必要がある。

4.1 CKMS へのアクセスコントロール

解説・考慮点

本節は、SP 800-130 の 8.1 節、8.2 節、8.3 節に記載されている事項について解説したものである。
アクセスコントロールには、セキュリティ境界において、認可されたエンティティのみがセキュリティ境界内部に入れるようにするための門番としての役割がある。これらが、暗号モジュールやセキュリティ境界内部の CKMS デバイス等と連携して CKMS のセキュリティを確保している。本節では、セキュリティ境界内部の CKMS デバイス等をセキュアに保つためのアクセスコントロールの方法について取り扱う。

4.1.1 物理セキュリティコントロール

① CKMS コンポーネント及びデバイスに対する物理セキュリティの方法を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.01	FR8.1	CKMS 設計は、それぞれの CKMS デバイスと意図する目的を明記しなければならない。	8.1 節

F.02	FR8.2	CKMS 設計は、CKMS コンポーネントを含むそれぞれのデバイスを保護するための物理セキュリティコントロールを明記しなければならない。	8.1 節
F.03	FR6.120	CKMS 設計は、全ての CKMS コンポーネント及びデバイスがどのように認可されない（不正な）物理アクセスから保護されるかを明記しなければならない。	6.8.8 節
F.04	FR6.121	CKMS 設計は、CKMS がどのように認可されない（不正な）物理アクセスを検知するかを明記しなければならない。	6.8.8 節

解説・考慮点

CKMS では、コンポーネント、デバイス及び CKMS 内に含まれる機微なデータの窃取及び改ざん、又はハードウェアやソフトウェアの改ざんから保護するため、CKMS コンポーネント及びデバイスは物理的に保護されるべきである。それらのセキュリティの重要性に応じて、一つ以上の物理的保護メカニズムが選択される。

CKMS の設計にあたって、項目 F.01～F.04 は、CKMS コンポーネント及びデバイス等に対する物理セキュリティの要求事項を明確化することを求めたものである。F.01 は CKMS デバイスの利用環境・場所や利用目的、F.02 はコンポーネント及びデバイスに対する保護手段についての検討項目であり、SP 800-130、8.1 節に保護手段の参考例が掲載されている。F.03 及び F.04 は保護手段の運用条件に関する検討項目である。

CKMS における物理的セキュリティ保護メカニズムを整理することを要求している。CKMS を構成するデバイスやコンポーネントの盗難やすり替え、物理的攻撃による改ざん、物理的攻撃による内部の機微な情報へのアクセスなどを防ぐために物理的保護が実施される。暗号モジュール自体が物理的保護メカニズムを備えている場合もあるが、多層防御として CKMS 全体を収容する施設（ファシリティ）としての物理的保護メカニズムも構築される場面が多い。

上記のように、SP 800-130、8.1 節には施設や収容エリアへの物理アクセスに関わる保護メカニズムや物理的侵入の検知メカニズムが例示されている。

施設に関わる物理セキュリティの要件や基準について、以下のような文献が参考になる。

- 「データセンター　セキュリティ　ガイドブック」、日本データセンター協会
- 「金融機関等コンピュータシステムの安全対策基準・解説書」、金融情報システムセンター

一般に施設レベルの物理的セキュリティ保護メカニズムとして、複数の段階の保護が設けられる。ある段階の物理的保護を解除するための認証情報や物理的鍵により、複数の段階の物理的保護を解除できることがないように保護メカニズムを構築することが重要である。CKMS を収容するエリアや CKMS を構成するデバイスに関わる物理的保護については、CKMS の利用や運用に関わるエンティティに物理アクセスを制限するように構築されるが、建物や施設全体は

CKMS に関わらない従業員も入構が許可されるので、物理アクセスを許可する範囲が段階によって異なる性質がある。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。図 4-1 にトイモデルで実施されている物理アクセスコントロールを示す。各保護エリアにアクセスするために必要となるクレデンシャル（認証用の識別情報）を白矢印に示しており、エリアごとに異なるクレデンシャルが必要となるように設計されている。

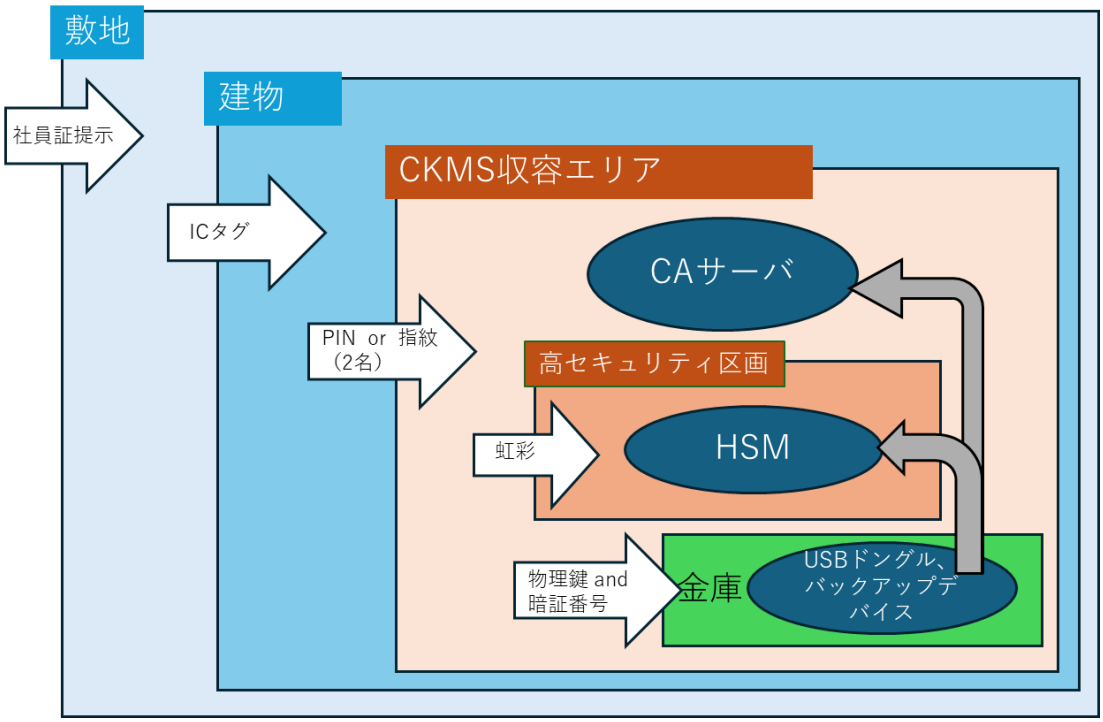


図 4-1 トイモデルにおける物理アクセスコントロール

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.01	当該 CKMS を構成する HSM、CA サーバ、さらに、HSM 内鍵情報のバックアップ用デバイス、及び HSM の管理者権限取得に必要となる USB ドングルは物理的保護を必要とする。
F.02	当該 CKMS では、物理的アクセスコントロールとして、敷地のフェンスと警備員、建物と CKMS を収容した部屋（図 4-1 の CKMS 収容エリア）の入退室管理機能、HSM を収容したラック扉（図 4-1 の高セキュリティ区画）の施錠機能、HSM の管理者権限取得に必要な USB ドングルや鍵情報のバックアップ用デバイスを保管した金庫がある。 さらに、暗号モジュールの物理的保護メカニズムとして FIPS 140-2/-3 レベル 3 の HSM における筐体の物理的保護がある。
F.03	F.02 に記載した物理的アクセスコントロールについて、敷地内にアクセスするには敷地

	入り口での社員証提示、建物に入る際は社員証内タグによるドア開錠、CKMS 収容エリアへの入室には PIN もしくは指紋認証による入り口扉の開錠、HSM を収容したラック扉の開錠には虹彩認証、USB ドングルやバックアップデバイスを保管した金庫の開錠には暗証番号及び物理鍵がそれぞれ必要である。ここで、CKMS 収容エリアの入り口扉の開錠には、内部不正の相互監視を主たる目的として複数名での入室認証を必要とする。また、メンテナンスや監査などの目的で部外者を CKMS 収容エリアに入室させる場合には、PC やスマートフォンなどの情報機器の持ち込みを含めて事前登録を必要とし、敷地入り口で警備員が確認する。
F.04	CKMS 収容エリアの入り口や室内には監視カメラが設置され、不審者を警備員が監視している。また、暗証番号の不一致が一定回数連続した場合には不審なアクセスとして警備員に通知される。

4.1.2 コンピュータシステムセキュリティコントロール

① OS に対するセキュリティの要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.05	FR8.3	CKMS 設計は、それぞれの CKMS デバイスに対して、全てのセキュアな OS の要求事項（いかなる必要な OS 設定も含む）を明記しなければならない。	8.2.1 節
F.06	FR8.4	CKMS 設計は、下記のどの堅牢化機能が CKMS によって実行されているかを明記しなければならない： a) 全ての必須でないソフトウェアプログラムとユーティリティをコンピュータから削除する b) 危殆化を受けやすいシステム機能及びアプリケーションに対するアクセスコントロールに最小権限の原則を適用する c) 危殆化を受けやすいシステム及びアプリケーションのファイルとデータに対するアクセスコントロールに最小権限の原則を適用する d) ユーザアカウントを合理的な運用に必要なだけに制限する、すなわち、もはや必要のないアカウントは無効化又は削除する e) 最小権限の原則でアプリケーションを動作させる f) 全てのデフォルトパスワード及びデフォルト鍵をそれぞれ強力なパスワード及びランダムに生成された鍵で置き換える g) システムの運用に必要でないネットワークサービスを無効化又は削除する	8.2.1 節

		h) システムの運用に必要でない全ての他の処理（service）を無効化又は削除する i) リムーバブルメディアを無効化する、又はリムーバブルメディアにおける自動実行機能を無効化しメディア挿入時の自動マルウェアチェック機能を有効にする j) システム運用に必要でないネットワークポートを無効化する k) オプションのセキュリティ機能を適切に有効化する l) セキュアにする他の設定オプションを選択する	
F.07	FR8.5	CKMS 設計は、OS の正しいインスタンス化を保証する BIOS 保護機能を明記しなければならない。	8.2.1 節

解説・考慮点

セキュアな OS はセキュアなコンピュータシステムの基礎であり、それなしにコンピュータシステム上でプログラム及びデータのセキュリティを保証することができない。
CKMS の設計にあたって、項目 F.05～F.07 は、CKMS デバイス等に搭載される OS に対するセキュリティの要求事項を明確化することを求めたものである。これには、OS 自体のセキュリティだけでなく、当該 OS 上で動作するソフトウェアやユーザ等の管理に対する要求事項も含む。SP 800-130、8.2.1 節にセキュリティ機能の参考例が掲載されている。

CKMS はコンピュータデバイス上に構築され、コンピュータデバイス上の OS によって、コンピュータ上のエンティティ認証とアカウント管理、各種操作に対する権限管理やプロセス管理、ソフトウェアコードのインテグリティチェック、ソフトウェアアップデート時のアップデートコードの署名検査、操作ログの管理、など様々なセキュリティコントロールに関わる機能が提供される。本節は CKMS が動作するコンピュータ環境におけるセキュリティコントロールのうち、OS によって実施される内容を整理することを求めている。

なお、IoT 向けの小規模デバイスなどで OS がないデバイスや、ファームウェアの更新やソフトウェアの追加を不可とするよう機能が制限されたデバイスもある。そのように機能が限定されたコンピュータデバイスでは本節の項目の多くが非該当となる。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.05	当該 CKMS では CA サーバ上の Linux OS において CKMS のエンティティ管理と権限管理などのセキュリティコントロールが行われる。

F.06	当該 CKMS の CA サーバ上の Linux OS によって次の堅牢化設定を行う。 a) CA としての機能を稼働させるために必須ではないアプリケーションを削除する b) システム機能及びアプリケーションに最小権限の原則を適用する c) ファイルとデータに最小権限の原則を適用する d) ユーザアカウントは CKMS のエンティティに限定する。不要なアカウントは無効化または削除する。また、sudo による管理者権限を持つアカウントを限定し、それらのアカウントで実行可能なコマンドを制限する f) 十分なエントロピーを確保できるように、パスワードの長さや複雑さ、及び鍵に関わる規則を設け、デフォルトのパスワード及び鍵を置き換える g) CKMS の運用に不要なネットワークサービスを無効化または削除する h) CKMS の運用に不要な他の処理を無効化または削除する i) CA サーバにおいて USB メモリなどのリムーバブルメディアの無効化または自動実行機能を無効化する j) 不要なネットワークポートを無効化する k) SELinux、AppArmor などのオプションのセキュリティ機能を有効化する l) SSH の設定強化、ルートログインの無効化、システムのアップデートなど、その他のセキュリティ設定オプションを有効化する
F.07	当該 CKMS の CA サーバのブート時にはセキュアブート処理を実施する。ROM 内ブートローダから BIOS、OS、CA アプリケーションの起動において実行コードのインテグリティチェックを行う。

② デバイスに対するセキュリティの要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.08	FR8.6	CKMS 設計は、それぞれの CKMS デバイスに必要なセキュリティコントロールを明記しなければならない。	8.2.2 節
F.09	FR8.7	CKMS 設計は、堅牢化の基となるデバイス／CKMS のセキュリティ設定要求事項及びガイドラインを明記しなければならない。	8.2.2 節

解説・考慮点

CKMS を構成する各々のデバイスに対して、認可されない使用から自らを保護するように設計されているか、外部から適用される保護が必要である。
CKMS の設計にあたって、項目 F.08 及び F.09 は、CKMS デバイス等に対するセキュリティの要求事項を明確化することを求めたものである。SP 800-130、8.2.2 節にセキュリティ機能の参考例が掲載されている。

CKMS デバイスにはそれをコントロールするホスト OS と独立したセキュリティコントロールを備えるものがある。本節はそのようなデバイスにおけるセキュリティコントロールの内容やデバイスの堅牢化に関わる機能を定めることを求めている。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.08	当該 CKMS を構成するデバイスのセキュリティコントロールには HSM で実施されるものがある。HSM 上の OS において HSM 内の処理に関わるユーザ認証と権限管理が実施される。また、HSM におけるイベントログ情報が取得される。
F.09	上記 F.08 の HSM における堅牢化に関わるセキュリティ設定については HSM の設定マニュアルに記載されている。

③ マルウェア感染防止に対する要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.10	FR8.8	CKMS 設計は、CKMS デバイスに対する以下のマルウェア防御能力を明記しなければならない： a) ウイルス対策ソフトウェア。アンチウイルススキャン、ソフトウェア更新、及びウイルスシグネチャデータベース更新を開始する時間周期及びイベントの指定を含む。 b) スパイウェア対策ソフトウェア。アンチスパイウェアスキャン、ソフトウェア更新、及びウイルスシグネチャ更新を開始する時間周期及びイベントの指定を含む。 c) ルートキット検出及び防御ソフトウェア。ルートキット検出、ソフトウェア更新、及びシグネチャ更新を開始する時間周期及びイベントの指定を含む。	8.2.3 節
F.11	FR8.9	CKMS 設計は、OS 及び CKMS アプリケーションソフトウェアに対する以下のソフトウェア完全性チェックの情報を明記しなければならない： a) ソフトウェア完全性がインストール時に検証される場合、検証がどのように実行されるかを記載する b) ソフトウェア完全性が定期的に検証される場合、検証が実行される頻度を記載する	8.2.3 節

解説・考慮点

データやファイル等をネットワーク（特に、保護されていないネットワーク）等を通して受信する CKMS デバイスは、受信した情報のマルウェア感染防止のための対策をすべきである。CKMS の設計にあたって、項目 F.10 及び F.11 は、CKMS デバイス等へのマルウェア感染防止に対する要求事項を明確化することを求めたものである。

上記のように、CKMS コンピュータシステムにおけるマルウェア感染の防御機能を定めることを求めている。ネットワークやリムーバブルメディアからマルウェアが持ち込まれる可能性があり、マルウェア（ウイルス、スパイウェア、ルートキット検出など）の検知ツールを利用するなどの緩和策がある。また、OS やアプリケーションソフトのインテグリティチェックもそれらのコードへのマルウェア感染の検出に有効である。

意図した相手と意図した通信しか行わず、通信内容も暗号学的に保護されている場合には外部からネットワーク経由でマルウェアが侵入する可能性は低減できるが、マルウェアの侵入経路は多様であることに注意すべきである。例えば、外部から持ち込まれたリムーバブルメディアや PC 端末が侵入口となり得ること、CKMS に関わるエンティティが悪意を持って感染させることなどもあり得る。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.10	当該 CKMS の CA サーバが定常的に通信を行うのは外部の IoT 向け証明書管理端末のみであるが、多層防御の一つとしてマルウェア検出ツールを CA サーバで動作させる。a)ウイルス、b)スパイウェア、c)ルートキットのいずれに対しても検出力を備えたツールである。シグネチャの更新はツールベンダの更新サイクルに依存する。
F.11	当該 CKMS の CA サーバのセキュアブート処理において、起動時に OS と CA アプリケーションはインテグリティチェックが実施される。また、OS と CA アプリケーションに対するソフトウェアパッチが配信され、コードにパッチを適用する際にはソフトウェアパッチのインテグリティチェックが実施される。

④ 監査機能に対する要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.12	FR8.10	CKMS 設計は、サポートされている監査可能イベントを明記し、それぞれのイベントは固定されているか選択可能であるかを示さなければならない。	8.2.4 節

F.13	FR8.11	それぞれの選択可能な監査可能イベントに対し、CKMS 設計は、イベントを選択する能力を持つ役割を明記しなければならない。	8.2.4 節
F.14	FR8.12	それぞれの監査可能イベントに対し、CKMS 設計は、記録されるデータを明記しなければならない。	8.2.4 節
F.15	FR8.14	CKMS 設計は、システムファイルの改変又はアクセスコントロールリストのようなセキュリティ属性のあらゆる改変について検知や防止をするため､危殆化を受けやすいシステムファイルに対するシステム監視要求事項を明記しなければならない。	8.2.4 節
F.16	FR8.13	CKMS 設計は、CKMS の正しい運用及びセキュリティを評価するために､どの自動化ツールが提供されているかを明記しなければならない。	8.2.4 節

解説・考慮点

CKMS では、イベント、イベントの発生日時、及びイベントを発生させたエンティティの識別子（ID）又は役割を検知及び記録することによって、セキュリティ関連イベントを監査すべきである。そのためには、監査管理者に対して可能な限り速やかに調査すべきあらゆる異常なイベントを検知し報告するとともに、監査の完全性が保証できるように監査ログの改ざんから保護されることが必要である。
また、セキュリティ設定共通化手順（Security Content Automation Protocol；SCAP）に規定されているような自動評価ツールは、現在のステータス及びコンピュータシステムの完全性の評価に有効な手段であり、システムファイル又はそれらのアクセスコントロール属性の改変、データファイルの完全性及び機密性の侵害等を検知し、警告及び監査イベントを発する監視ツールとしても利用できる。

CKMS の設計にあたって、項目 F.12～F.15 は、監査機能に対する要求事項を明確化することを求めたものである。F.16 は SCAP を利用する場合に SCAP に対する事項を明確化することを要求したものである。SCAP を利用しない場合には、F.16 は検討対象外である。

異常な操作や処理の検出や解析にイベントログが利用される。イベントログを利用した監査機能について明確にすることを求めている。汎用 OS には一般にイベントログとして取得する対象のイベントや監視報告に関わるコマンドやツールが提供されている。アプリケーションプログラムにもアプリケーションに関わるイベントのログ機能を備えるものがある。

セキュリティ設定共通化手順（Security Content Automation Protocol；SCAP）は米国 NIST が策定した情報セキュリティ対策の自動化と標準化を実現する技術仕様群である。脆弱性対応で利用される CVE（Common Vulnerabilities and Exposures：共通脆弱性識別子)、CVSS（Common Vulnerability Scoring System：共通脆弱性評価システム)、CPE（Common Platform Enumeration：共通プラットフォーム一覧）に加えて、CCE（Common Configuration Enumeration：共通セキュリティ設定一覧）や OVAL（Open Vulnerability and

Assessment Language：セキュリティ検査言語）等も標準仕様として含まれる。SCAP については IPA のサイトに概説がある[11]。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.12	当該 CKMS では、CA サーバの Linux OS においてユーザ管理、アプリケーション管理、堅牢化に関わる設定管理(F.06 に記載)などのイベントログを監査対象とする。また、CA ソフトウェアにおいて証明書の発行や失効処理に関わるイベントログが監査対象として利用できる。さらに、HSM でも操作ログや実行処理のログが取得され、監査対象となる。これらの監査対象のログは選択可能である。
F.13	上記 F.12 に記載したイベントログの選択は、CA サーバについては CKMS 管理者、HSM については HSM 管理者が実施できる。HSM 管理者は CKMS 管理者のうち HSM 管理者としてアサインされた担当者であるが、マルチパーティコントロールがされており、1 名では権限を取得できない。
F.14	上記 F.12 に記載したイベントログとして記録されるデータには、実行者、時刻、処理対象及び処理内容が含まれる。具体的なデータ形式は管理マニュアルを参照。
F.15	上記 F.12 に記載したイベントログのうち、ユーザ管理、特にアクセス権限管理に関わるイベントはリアルタイムでの監視対象とし、イベント発生時に CKMS 管理者及び CKMS 責任者に通知されるように設定する。
F.16	当該 CKMS では脆弱性情報に関わる CVSS や CVE を参照するが、自動化ツールは利用しない。本項目は対象外である。

4.1.3 ネットワークセキュリティコントロール

① セキュリティ境界をコントロールするためのネットワークセキュリティコントロールデバイスに対する要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.17	FR8.15	CKMS 設計は、CKMS によって採用される境界保護メカニズムを明記しなければならない。	8.3 節
F.18	FR8.16	CKMS 設計は、以下を明記しなければならない： a) 使用されるファイアウォールのタイプとファイアウォールを介して許可されるプロトコル。それぞれのプロトコルタイプの発信元（source）と宛先（destination）を含む	8.3 節

[11] セキュリティ設定共通化手順 SCAP 概説、https://www.ipa.go.jp/security/vuln/scap/scap.html

		b) 使用される侵入検知・防止システムのタイプ。ログ及びセキュリティ侵害対応の機能を含む	
F.19	FR8.17	CKMS 設計は、CKMS デバイスをサービス拒否（DoS）攻撃から保護するために使用される方法を明記しなければならない。	8.3 節
F.20	FR8.18	CKMS 設計は、使用されるそれぞれの方法がどのようにサービス拒否攻撃から保護するかを明記しなければならない。	8.3 節

解説・考慮点

ネットワーク化された CKMS デバイスへの外部からの攻撃を防護するためには､それら CKMS デバイスをセキュリティ境界内部に配置するとともに、ファイアウォール及び侵入検知・防御システム等のネットワークセキュリティコントロールデバイスをいくつか組み合わせてセキュリティ境界内部を保護する必要がある。そのため、ネットワークセキュリティコントロールデバイスは物理的にセキュアな場所に配置され、セキュアな操作に必要なユーザアカウント及びネットワークサービスのみを提供すべきである。
また、CKMS デバイスへの DoS/DDoS 攻撃は CKMS のサービス提供が停止することにつながる可能性があるので、DoS/DDoS 攻撃を防止することも必要である。

CKMS の設計にあたって、項目 F.17 及び F.18 は、セキュリティ境界をコントロールするためのネットワークセキュリティコントロールデバイスに対する要求事項を明確化することを求めたものである。F.19 及び F.20 は DoS/DDoS 攻撃への対策のための要求事項を明確化することを求めたものである。

上記のように、CKMS へのネットワーク経由の侵入や攻撃を防御するために利用するネットワーセキュリティコントロールのメカニズムを明確にすることを求めている。具体的なメカニズムには SP 800-130 に例示されているように、ファイアウォール、フィルタリングルータ、仮想プライベートネットワーク（VPN)、侵入検知システム（IDS)、侵入防止システム（IPS）などがある。これらの機能に加えてディープ・パケット・インスペクションによるアプリケーションレベルでのパケットの検査や外部からの脅威インテリジェンスの活用などの機能を加えたネクスト・ジェネレーション・ファイアウォールと呼ばれる製品もある。

境界コントロール型の防御は、内部ネットワークは安全な領域として外部ネットワークとの接続点での防御に注力するものであるが、内部ネットワークに持ち込まれるデバイスに様々なものがあるなど侵入経路が多様化しており、クラウドサービスの普及などにより物理的なネットワーク境界が明確でなくなっている場合がある。このような状況で、従来のシグネチャベースやフィルタリングルールでは検知できない脅威も存在する。マルウェアに感染し外部からコントロールされている内部ネットワークの端末や、内部犯によるデータ窃取がその一例である。そのような不正行為を試みるトラヒックの振る舞いを定常時・非定常時の差分などから検知して対処する Network Detection and Response（NDR）というカテゴリの製品も登場している。NDR では人工知能や機械学習を用いて異常なふるまいを検知しており、従来の攻撃方法を分析してシグネチャをアップデートする手法の欠点であったゼロデイ攻撃に対するアプローチにもなる。

関連した動向として、従来の内部ネットワークは安全という考え方に基づかないセキュリティアーキテクチャとして Zero Trust Architecture（ZTA）が提唱されている。SP 800-207 に基本的な概念が整理されており、ZTA では必要最小限のアクセス権のみを付与することでリソースを制限し、暗黙的に信頼が付与される状況を無くすことでリソースが保護され、さらに継続的な状態評価を実施することを目標としている。

実際のネットワーク構成やセキュリティモデルを踏まえて、それに適したネットワークコントロールを実現するように留意されたい。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.17	当該 CKMS では、ネットワークセキュリティコントロールデバイスとして、フィルタリング機能を搭載したルータを利用する。さらに、侵入検知用に IDS を利用する。また、緊急時に外部からシステム管理を行えるように VPN を設置する。
F.18	当該 CKMS で利用するファイアウォールと侵入検知システムは以下のように設定する。 a) ファイアウォールはルータのパケットフィルタリング機能で実現する。通信パケットの発信元と宛先は CA サーバと IoT 製品向け証明書管理端末、及びシステム管理用に予め登録した PC 端末に限定する。 b) 侵入検知システム（IDS）は、CA サーバ上で稼働させるホスト型を利用する。
F.19	当該 CKMS において、DoS/DDoS 攻撃対策に関わるメカニズムは F.18 に記載のフィルタリング機能を搭載したルータ及び IDS である。
F.20	当該 CKMS において、DoS/DDoS 攻撃対策は、ルータにおいて予め設定したアドレス以外から送受信したパケットを遮断すること、CA サーバ上の IDS で予め設定した正常なアクセスパターン以外の通信を検知すること、の 2 つのメカニズムによる。

4.2 システム保証

解説・考慮点

本節は、SP 800-130 の 9.8 節に記載されている事項について解説したものである。
本節で取り扱うシステム保証とは、CKMS で利用するコンポーネント及びデバイスがそもそもセキュアなものであり、不正な組み込みがされていないことを保証するためのプロセスである。

本節で検討される項目は、CKMS 全体やその構成要素であるデバイスやコンポーネントの開発環境やメンテナンスに関わるセキュリティを保証するために検討すべきものである。特に CKMS の開発工程において考慮すべき項目は「製品セキュリティ」と呼ばれる活動と共通するものがある。例えば、「脆弱性対処に向けた製品開発者向けガイド（IPA）」に開発工程における脆弱性の混

入を抑えるための施策が書かれている。また、FIPS 140-2 には暗号モジュールに対するセキュリティ要件として、設計保証（Design Assurance）の項目があり、その中に構成管理、配送、開発の小項目がある。これらの要件が本節に関わるものと考えられる。

CKMS の構築において、システム開発を委託した SI 業者、及びデバイスやコンポーネントの調達先であるベンダに対して適切な保守契約を締結して、運用後のメンテナンスにおけるセキュリティ維持のサポート手段を確保することが望ましい。

① 構成管理に関する要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.21	FR9.10	CKMS 設計は、以下を明記しなければならない： a) 構成制御の下に置かれているデバイス（ソースコード、スクリプト実装、実行コード、ファームウェア、ハードウェア、文書、及びテストコードを含む） b) 構成制御の下でコンポーネント及びデバイスへの認可された変更だけが行われたことを保証するための保護要求事項（例えば、形式的認可及び適切な記録保持）	9.8.1 節

解説・考慮点

構成管理は、製品への認可されていない又は意図しない変更によってセキュリティが低下せず、かつ機能的欠陥が取り込まれることがないことを保証するための手法である。
項目 F.21 は、CKMS の設計にあたって、管理対象や構成変更方法等、構成管理に関する要求事項を明確化することを求めたものである。

上記のように、CKMS の要素となるデバイスやコンポーネントに対して、構成管理の対象とするものを定めることを要求している。構成管理を適切に実施して製品を開発したり、保守を行ったりするのはベンダ側の対応事項であるが、利用者側でも利用するデバイスやコンポーネントの型番やファームウェアバージョンを管理することは重要である。これらの情報は、CKMS の運用中に発見された脆弱性情報や欠陥などが対象のデバイスやコンポーネントに該当するものかどうか、該当する場合にファームウェア更新などによる機能更新を実施すべきかどうかを迅速に判断して対処することに利用される。

また、自社開発のソフトウェアに対しては社内で構成管理を行うことが必要となる。なお、暗号モジュールに関する構成管理について、FIPS 140-2 では小節を設けて簡単に要件が記載されている。

構成管理に関連して SBOM（Software Bill of Materials：ソフトウェア部品表）の利活用が注目されている。OSS（Open Source Software）の利用が拡大するなどソフトウェアのサプライチェーンが複雑化している中で、SBOM の活用によってソフトウェアの脆弱性管理を効率化することが期待されている。SBOM については経済産業省が手引書を公開している。

- 「ソフトウェア管理に向けた SBOM の導入に関する手引」、経済産業省

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。このトイモデルでは、CA システム自体は社内でシステム構築を行うが、サーバ、HSM、ルータなどのデバイス及び CA ソフトやサーバ OS などのソフトウェアコンポーネントは外部から調達した製品を利用することを想定している。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.21	当該 CKMS における構成管理の対象を以下のように定める。 a) 構成管理の対象は、サーバ、HSM、ルータ、CA ソフトウェア、サーバ OS である。特に、ソフトウェアコンポーネントである CA ソフトウェアとサーバ OS は脆弱性が発見される可能性が比較的高く、バージョン情報を元に脆弱性情報の監視を行う。 b) 上記のデバイスやコンポーネントに対してファームウェア更新などによってバージョンが変更された場合は、システムインベントリの管理情報を適切に更新する。

② セキュアな配送に関する要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.22	FR9.11	CKMS 設計は、以下を含む、CKMS で使用される製品のセキュアな配送[12]の要求事項を明記しなければならない： a) 配送プロセス中に製品がタンパー（tamper）されていない、又はタンパーされたことが検知されることを保証するための保護要求事項 b) 配送プロセス中に製品が交換されていない、又は交換されたことが検知されることを保証するための保護要求事項 c) 要求されていない配送が検知されることを保証するための保護要求事項 d) 製品の配送が差し止め又は遅延していない、及び差し止め又は遅延が検知されることを保証するための保護要求事項	9.8.2 節

解説・考慮点

CKMS で使用される製品には、セキュアな配送の保証（受領した製品が間違いなく注文した製品であり、改ざんされていないこと）が必要である。

[12] 「暗号鍵管理システム設計指針（基本編）」では「セキュアな配付」と表記されている。SP 800-130 では secure delivery となっており、CKMS で利用するデバイスやコンポーネントなどの製品が対象とすると「セキュアな配送」と表記するのが適当と考え、本書ではそのように表記した。

項目 F.22 は、CKMS の設計にあたって、セキュアな配送を保証・確認するための要求事項を明確化することを求めたものである。

上記のように、CKMS の要素であるデバイスやコンポーネントの配送においても製品のすり替えや改ざんがされていないことを確認できることが望ましい。本節の対象となるのはハードウェア製品やメディアに記録したソフトウェア製品である。物理的な製品梱包における開封検知の仕組みや初期起動に必要な鍵情報を別送するなどの手段をベンダ側で実施していることがある。また、製品の配送状況のトラッキングを可能とするサービスなども関連する。CKMS 設計側ではベンダに実施可能な手段を確認するのが基本的な対応となる。

なお、暗号モジュールに関するセキュアな配送について、FIPS 140-2 では小節を設けて配送とインストール、初期化について簡単に要件が記載されている。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.22	a) 当該 CKMS で調達する HSM、サーバ、ルータについて配送上のセキュリティ対策をベンダ側で用意しているかを確認する。例えば、製品梱包におけるタンパーシールや初期起動に必要な鍵情報の配送方法など。 b) 上記と同様。 c) 上記と同様。 d) 製品配送の差し止めや遅延の確認は購買管理システムによって行う。

③ 開発環境及びメンテナンス環境におけるセキュリティに関する要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.23	FR9.12	CKMS 設計は、以下を含む、CKMS の開発環境及びメンテナンス環境におけるセキュリティ要求事項を明記しなければならない： a) 物理セキュリティ要求事項 b) 開発者、試験者、及び保守員に対する身分照会及びバックグラウンドチェックのような人的セキュリティ要求事項 c) 複数人員（multi-person）による制御、及び職掌分散（separation of duties）のような手続き的セキュリティ d) 開発環境及びメンテナンス環境の保護、及び認可されたユーザにアクセスを許可するアクセスコントロールの提供のためのコンピュータセキュリティコントロール	9.8.3 節

		e) ハッキングの試みから開発環境及びメンテナンス環境を保護するためのネットワークセキュリティコントロール f) 開発下のソフトウェア及びその制御データの完全性を保護するための暗号学的セキュリティコントロール g) ツール（例えば、エディタ、コンパイラ、ソフトウェアリンカ、ローダ等）が信頼でき、マルウェアのソースでないことを保証するために利用する手段	

解説・考慮点

CKMS 開発環境及びメンテナンス環境は、物理的、人的、及びハッキングの脅威から適切に保護されなければならない。また、コンパイラ、ソフトウェアリンカ、テキストエディタといった開発ツールを自動的に信頼すべきではない。
項目 F.23 は、CKMS の設計にあたって、セキュアな開発環境及びメンテナンス環境を実現するための要求事項を明確化することを求めたものである。これには、物理的セキュリティ、人的セキュリティ、及びシステムセキュリティの全てを含む。

CKMS の開発や構築の過程、あるいはメンテナンスの過程でマルウェアを混入させてしまったり、バックドアが作り込まれてしまったりすることを防止するための要求項目である。開発やテスト、メンテナンスなどの人員が悪意を持って不正行為を行う可能性もあれば、意図せずにマルウェアが混入する可能性もある。開発を実施するエリアを外部アクセスから厳格に管理する、外部とのネットワーク接続を遮断するなどの物理的対策、悪意を持った人員による不正操作を抑止する相互監視などの人的対策、開発に利用するコンパイラなどの開発ツールを経由したマルウェア感染防止に関わる対策など様々な緩和策が考えられる。

なお、調達したデバイスやコンポーネントにマルウェアが感染している可能性もある。これはサプライチェーンセキュリティと呼ばれる領域に関わる事項であるが、責任はベンダ側が負うものであり、利用側での緩和策は信頼できるベンダの製品を採用すること等に限定される。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.23	当該 CKMS の開発に当って、開発環境のセキュリティとして以下の内容を実施する。 a) CKMS の開発に利用するエリアは、CKMS 運用時と同じエリアを利用し、入退室管理を施している。 d) CKMS の開発を担当する人員は、CKMS の管理者ロールを有する人員の一部であり、開発環境やツールへのコンピュータアクセスコントロールを施している。 e) 開発環境は外部アクセスからネットワークセキュリティコントロールを施している。

	g) 開発ツールからのマルウェア混入のリスクを低減させるため、開発ツールのうち、コンパイラ、リンカなどの実行コードを生成するツール、及びインストーラは有償版（サポート契約付き）を利用する。また、自社開発のツールもセキュアコーディングや脆弱性検査などを実施して開発されたツールであることを確認する。

④ 欠陥修正能力に関する要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.24	FR9.13	CKMS 設計は、以下を含む、システムの欠陥を検知する CKMS の能力を明記しなければならない： a) 既知解テスト b) エラー訂正コード c) 異常故障診断技術 d) 機能テスト	9.8.4 節
F.25	FR9.14	CKMS 設計は、以下を含む、欠陥を報告する CKMS の能力を明記しなければならない：ステータスレポートメッセージを機密性、完全性、及びソース認証保護付きで作成する能力、及び認可されない遅延を検知する能力	9.8.4 節
F.26	FR9.15	CKMS 設計は、欠陥を分析し、かつ起こりやすい又はよく知られている欠陥に対する修正を作成／取得する CKMS の能力を明記しなければならない。	9.8.4 節
F.27	FR9.16	CKMS 設計は、機密性、完全性、及びソース認証保護付きで修正を送信し、かつ認可されない遅延を検知する CKMS の能力を明記しなければならない。	9.8.4 節
F.28	FR9.17	CKMS 設計は、時宜を得て修正を実装する CKMS の能力を明記しなければならない。	9.8.4 節

解説・考慮点

CKMS は、迅速かつセキュアな方法でシステムの欠陥を検知、報告及び修正する能力を持つべきである。特に、自動化された技術であることが望ましい。
CKMS の設計にあたって、項目 F.24～F.28 は、欠陥修正能力に関する要求事項を明確化することを求めたものである。F.24 は検知、F.25 は報告[13]、F.26～F.28 は修正・対処に相当する。

上記のように欠陥の検知、報告、修正に関わる対応を明確にすることを要求している。

[13] 「暗号鍵管理システム設計指針（基本編）」では F.25 に対して「通知」と「報告」の表記が混在しているが、SP 800-130 では report となっており、本書では「報告」で統一した。

FIPS 140-2 には、暗号モジュールのセキュリティ要件として、自己テストに関わる要件がまとめられている。暗号アルゴリズムの実装に関わる既知解テストやソフトウェアのインテグリティテストがまとめられており、インテグリティテストではエラー訂正コードに基づく方法と、MAC やデジタル署名などの認証データに基づく方法が書かれている。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.24	当該 CKMS で利用する HSM は FIPS 140-2/-3 レベル 3 の認証取得を要件としている。その要件として、HSM 内に実装されている ECDSA 署名アルゴリズム及びその内部で用いるハッシュ関数 SHA-256 と SHA-384 では既知解テストを実施している。また、ECDSA の鍵生成や署名生成時のナンス生成に利用する乱数生成器においてエントロピー試験やポスト処理部の既知解テストを実施している。また、HSM 起動時にファームウェアの完全性テストを実施している。 CA サーバで動作するサーバ OS 及び CA ソフトウェアも起動時にセキュアブートを実施しており、ブート時に起動するコードに付与されたデジタル署名の検証を実施する。また、CA サーバを継続的に動作させる場合、サーバ監視プログラムを動作させ、状態監視及び異常検知を継続的に実施する。可能であれば、サーバ OS 及び CA ソフトウェアに関する定期的なテストを実施する。
F.25	当該 CKMS で利用する HSM において、自己テストでエラーを検出した場合の状況を表す診断コードなどのステータスレポートは、完全性保護や内容の秘匿などを実施した上でベンダに送付される。
F.26	当該 CKMS では、HSM 及び CA ソフトウェア、サーバ OS はそれぞれのベンダと保守契約を締結する。これらのデバイス及びソフトウェアコンポーネントにおける欠陥の分析、修正コードの作成やデバイス交換などの修正対応はベンダ側の体制に委ねる。
F.27	当該 CKMS では、HSM のファームウェア更新において、更新ファームウェアイメージの完全性と作成元の認証を行えるよう、セキュアなアップデート機能のサポートを要求する。 CA サーバのサーバ OS 及び CA ソフトウェアに対する修正パッチの配布においても同様の機能がサポートされる。
F.28	当該 CKMS では、F.26 に記載のように、ベンダと有償のサポート契約を締結する。サポート契約において、欠陥や脆弱性の対応条項があることを確認する。欠陥の検知時にはタイムリーに修正対応を実施できるようベンダと調整する。

4.3　セキュリティアセスメント

解説・考慮点

本節は、SP 800-130 の 11 章に記載されている事項について解説したものである。
CKMS のセキュリティを維持するため、CKMS のセキュリティライフタイムを通して様々なタイミングでいくつかのセキュリティアセスメントが実施される。また、必要に応じて、メンテナンスも実施しなければならない。本節では、セキュリティアセスメント及びメンテナンスについて取り扱う。

CKMS はシステムの初期立ち上げ時だけではなく、定期的または必要に応じてセキュリティアセスメントを実施するべきである。例えば、CKMS の構成要素であるデバイスやコンポーネントに対する新たな脆弱性の発見や、新たな攻撃手法の考案などにより、設計当初と比較してセキュリティが低下する恐れがある。また、運用の過程で、担当者の退職や異動による変更がシステムのアクセス権や通知先などに速やかに反映されていないことなどを起点に、不正操作やセキュリティインシデントのリスクが発生する可能性も生じる。これらのリスクを特定するためには、システム構成や運用状況の変化を監視し、必要に応じてセキュリティアセスメントを実施することが重要である。

本節では、セキュリティアセスメントを実施する状況とアセスメントにおけるスコープを整理することを求めている。また、各種のセキュリティコントロールによる堅牢化を維持するためのメンテナンス作業も定めることを求めている。

① 完全セキュリティアセスメントで実行される要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.29	FR11.1	CKMS 設計は、完全な CKMS セキュリティアセスメントの前又は同時に行われる、必要な保証実行策を明記しなければならない。	11.1 節
F.30	FR11.2	CKMS 設計は、完全なセキュリティアセスメントが繰り返される状況を明記しなければならない。	11.1 節
F.31	FR11.3	CKMS 設計は、あらゆる CKMS デバイスについて、認証を受けた全ての認証プログラムを明記しなければならない。	11.1.1 節
F.32	FR11.4	CKMS 設計は、認証済みデバイスに対する全ての認証番号を明記しなければならない。	11.1.1 節
F.33	FR11.5	CKMS 設計は、完全なセキュリティアセスメントの一部として、アーキテクチャレビューを必要とするかどうかを明記しなければならない。	11.1.2 節
F.34	FR11.6	アーキテクチャレビューが必要である場合、CKMS 設計は、アーキテクチャレビューチームに必要なスキルセットを明記しなけれ	11.1.2 節

		ばならない。	
F.35	FR11.7	CKMS 設計は、必要な全ての CKMS の機能テスト及びセキュリティテストを明記しなければならない。	11.1.3 節
F.36	FR11.8	CKMS 設計は、今までに実行された全ての機能テスト及びセキュリティテストの結果を報告しなければならない。	11.1.3 節
F.37	FR11.9	CKMS 設計は、今までに実行されたあらゆる完了したペネトレーションテストの結果を明記しなければならない。	11.1.4 節

解説・考慮点

配備前又は配備時に実施すべきセキュリティアセスメントであり、セキュリティアセスメントに課すことができる実行策には以下のものが含まれる。

- 第三者検証のレビュー（CAVP、JCMVP/CMVP、CC などの検証プログラム、等）
- システム設計のアーキテクチャレビュー
- 機能テスト及びセキュリティテスト
- ペネトレーションテスト

CKMS の設計にあたって、項目 F.29～F.37 は、完全セキュリティアセスメントで実行される要求事項を明確化することを求めたものである。F.29 はアセスメントの内容、F.30 はアセスメントの実施条件、F.31 及び F.32 は検証プログラム、F.33 及び F.34 はアーキテクチャレビュー、F.35 及び F.36 は機能テスト及びセキュリティテスト、F.37 はペネトレーションテストに関する要求事項がそれぞれ対象である。なお、F.31～F.37 で該当しない項目は検討対象外である。

上記のように、SP 800-130 ではセキュリティアセスメントとして、第三者検証のレビュー、システム設計のアーキテクチャレビュー、機能テスト及びセキュリティテスト、ペネトレーションテストの 4 つを具体的な実行策として記載している。このうち、第三者検証のレビュー、機能テスト及びセキュリティテストは、本ガイダンスの「3.2 セキュリティ評価・試験」でも検討項目としている。これらのテストについて、評価の対象やスコープが同一ならば既に実施した試験結果や検証資料を確認するだけでよい。

本節では、完全なセキュリティアセスメントとして実施する実行策を定めることと、どのような場面で完全なアセスメントを実施するか（再度実施するか）を定めることを求めている。実行策は上記の 4 つ以外を実施してもよいが、上記 4 つの実行策のそれぞれについては実施するかどうか、実施する場合には実施内容を定めること、これまでの実施結果を確認することを求めている。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.29	当該 CKMS において、完全セキュリティアセスメントでは次の 4 つを実施する。利用する HSM に関わる第三者認証のレビュー、CKMS 全体のアーキテクチャレビュー、CKMS 提供機能に関わる機能テスト及びセキュリティテスト、CKMS を模擬した環境に対するペネトレーションテストである。
F.30	当該 CKMS において、完全セキュリティアセスメントはシステム構築完了後、運用開始前に実施する。十分な結果でなかったテストは合格するまで再度実施する。運用開始以降は、原則として完全セキュリティアセスメントは実施しない。
F.31	完全セキュリティアセスメントの一項目である第三者検証のレビューとして、利用する HSM が FIPS 140-2/-3 レベル 3 認証を取得済であり、現在も有効であることを確認する。
F.32	F.31 において、HSM ベンダから認証情報の提供を受け、CMVP の認証番号を確認する。
F.33	当該 CKMS において、完全セキュリティアセスメントの一項目であるアーキテクチャレビューは、プライベート CA システム全体を対象として実施する。社内でセキュリティシステムの構築経験を持つ専門家によるレビューを実施する。適当な専門家がアサインできない場合は外部のセキュリティコンサルタントにレビューを委託する。
F.34	F.33 のアーキテクチャレビューにおいて、レビューアーは以下の技術に関する十分な実務経験を持つことを条件とする：コンピュータセキュリティ、ネットワークセキュリティ、暗号と情報セキュリティ、コンピュータシステム構築、セキュリティアーキテクチャ設計、リスク管理、セキュリティテスト。 また、公的な資格として、CISSP (Certified Information Systems Security Professional)、情報処理安全確保支援士（Registered Information Security Specialist, RISS）、CISM (Certified Information Security Manager)、ISO/IEC 27001 Lead Auditor などアーキテクチャレビューに必要な知識を有していることが望ましい。
F.35	当該 CKMS において、完全セキュリティアセスメントの一項目である機能テスト及びセキュリティテストは、プライベート CA の機能である、IoT 製品向け証明書の発行、同証明書に関わる失効リスト（CRL）の発行の 2 つに対して実施する。具体的には、本ガイダンス 3.2 節の③機能テスト及びセキュリティテストと同一であり、E.29 に記載している。
F.36	F.35 の機能テスト及びセキュリティテストは、本ガイダンス 3.2 節の E.29 として先に実施済であれば、完全セキュリティアセスメント時にはテスト結果を確認し、それが現在の設計及び運用環境において有効であることを評価する。

F.37	当該 CKMS において、ペネトレーションテストはセキュリティアセスメントの一環として実施を検討する場合がある。これまでに実施した結果はないが、セキュリティアセスメントにおける他のテスト結果や運用状況を踏まえて、必要に応じて実施する。 なお、運用前または運用中のアセスメントにおいて、セキュリティコントロールに重大な脆弱性が見つかった場合には、ペネトレーションテストを実施することで、脆弱性の悪用が可能かどうかを評価する。この場合、検証用のテスト環境を構築して以下のゴールを設定したテストを実施する。 ● 脆弱性がセキュリティコントロールに与える影響の確認 ● セキュリティコントロール機能により、システム障害や脅威が発生した場合でも、設計されたセキュリティ要件を満たす状態に回復するかの確認 また、ペネトレーションテストの結果は文書化し、セキュリティアセスメントの一部として関係者に報告する。

② 定期的なセキュリティレビューで実行される要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.38	FR11.10	CKMS 設計は、セキュリティレビューの周期を明記しなければならない。	11.2 節
F.39	FR11.11	CKMS 設計は、CKMS デバイスの観点から、セキュリティレビューのスコープを明記しなければならない。	11.2 節
F.40	FR11.12	CKMS 設計は、レビュー対象のそれぞれの CKMS デバイスに対して行われる実行策の観点で、定期的なセキュリティレビューのスコープを明記しなければならない。	11.2 節
F.41	FR11.13	CKMS 設計は、定期的なセキュリティレビューの一部として実行される機能テスト及びセキュリティテストを明記しなければならない。	11.2 節

解説・考慮点

システムコントロール、物理コントロール、手続き的コントロール及び人間によるコントロールが規定等に整備され、その通りに運用していることを保証するために、定期的にレビュー[14]を実施すべきである。このレビューでは、少なくとも、前回のセキュリティレビューからのシステム変更箇所の検査、及び定期的な機能テスト及びセキュリティテストの実行が行われる。CKMS の設計にあたって、項目 F.38～F.41 は、定期的なセキュリティレビューで実行される要求事項を明確化することを求めたものである。F.38 はレビューの実施条件、F.39 及び F.40

[14] 「暗号鍵管理システム設計指針（基本編）」では「定期的なセキュリティアセスメント」と表記されている。SP 800-130 では periodic security review となっており、「定期的なセキュリティレビュー」と表記するのが適当と考え、本書では節のタイトルを含め、そのような表記で統一した。

はレビューの範囲及び内容、F.41 は機能テスト及びセキュリティテストに関する要求事項がそれぞれ対象である。

上記のように、定期的なセキュリティレビューとして実施する内容と頻度（実施時期）を定めることを要求している。定期的なセキュリティレビューは完全なセキュリティアセスメントを補完するものであり、CKMS の設計時に定めたセキュリティ要件が運用中に適切に維持されているかを確認するものである。例えば、デバイスに最新のアップデートを適用した結果、提供されていた機能やメカニズムの一部に仕様変更があり、対応が必要であったり、それによって CMVP 等の第三者セキュリティ認証が失効したりする可能性がある。前回レビュー時点からの変更箇所の検査や、機能テスト及びセキュリティテストの実施によって、そのようなリスクを早期に検出できる。

なお、定期的レビューに関わる CKMS 運用中の活動として以下が挙げられる。

a) CKMS 内のデバイスやコンポーネント向けにベンダから提供されたアップデートパッチを適用すべきかどうかを検証環境で確認・検討し、必要なパッチを本番環境に適用する活動
b) CKMS 内で取得した実行要求や操作のログを分析して、不正操作や異常な要求がないかを確認する活動
c) CKMS エンティティとして権限を有する担当者の変更管理を行い、不要な権限を削除し、新しい担当者への適切な権限を付与する活動

これらの活動は、日常的に実施されるセキュリティメンテナンスとして位置づけられる。定期的なセキュリティレビューの一項目として、これらのメンテナンスが適切に実施されていることを確認し、必要に応じて改善提案を行う。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.38	当該 CKMS の管理担当は、定期的なセキュリティレビューとして以下の内容を実施する。 ● セキュリティパッチの適用状況のチェック HSM、CA ソフトウェア、及び CA サーバの OS のそれぞれに対してセキュリティパッチが適用されているか確認する ● 操作ログのレビュー CA サーバ及び HSM における操作ログを定期的に確認し、異常な操作やアクセスがないかどうかをチェックする ● エンティティ及び ACL の最新化チェック CA サーバ及び HSM に登録されたエンティティ及び ACL が運用ポリシーに基づき正確かつ最新であることを確認する。レビュー時には変更履歴を照合し、未承認の変更がないことを確認する

	これらのレビューは、F.44 で定義されたセキュリティメンテナンスの一環として実施される日常的な活動を定期的に振り返る目的で行う。具体的な頻度は、各組織のセキュリティポリシーやリスク評価に基づき、システムの重要度や使用状況に応じて柔軟に設定する。
F.39	F.38 に記載したように、当該 CKMS を構成するデバイスとして HSM 及び CA サーバを対象に、ソフトウェア部品の更新と脆弱性情報への対応を主とした定期的なセキュリティレビューを実施する。
F.40	定期的なセキュリティレビューの実施内容は F.38 に記載したものである。
F.41	当該 CKMS における HSM 及び CA サーバにパッチプログラムを適用する場合、CKMS が提供する機能への影響を確認するため、検証環境を用いた機能テストを実施する。この機能テストでは F.35 に準拠した手順に基づき、以下を検証する。 ● システム全体が正常に動作していること ● 暗号鍵管理や署名生成などの CKMS の主要機能が影響を受けていないこと ● パッチ適用後もセキュリティ要件が満たされていること この機能テストの結果は記録として保管し、問題が発生した場合には適切な修正を速やかに実施する。

③ 追加のセキュリティアセスメントで実行される要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.42	FR11.14	CKMS 設計は、追加のセキュリティアセスメントが実施されるべき状況を明記しなければならない。	11.3 節
F.43	FR11.15	CKMS 設計は、追加のセキュリティアセスメントのスコープを明記しなければならない。	11.3 節

解説・考慮点

システムに著しい変更が加えられたとき、以下の範囲での変更箇所への追加のセキュリティアセスメントを実行すべきである。なお、累積的なシステム変更が著しい場合には、完全セキュリティアセスメントを実施すべきである。

- 前回のセキュリティアセスメント以降の第三者認証されたデバイスへの変更
- システム設計変更後のアーキテクチャレビュー
- CKMS の機能テスト及びセキュリティテスト

CKMS の設計にあたって、項目 F.42 及び F.43 は、追加のセキュリティアセスメントで実行される要求事項を明確化することを求めたものである。F.42 はアセスメントの実施条件、F.43 はアセスメントの範囲及び内容に関する要求事項がそれぞれ対象である。

上記のように、CKMS に大きなシステム変更があった場合には、追加のセキュリティアセスメントを実施すべきである。追加アセスメントの項目は、完全なセキュリティアセスメントの項目に基づいてシステム変更の影響に応じて選択する。「大きなシステム変更」とは、以下のようなケースを指す。

- CKMS 全体、または CKMS を構成する中核デバイスに関わる機能変更（例：暗号鍵生成、署名、認証機能などの改修や追加）
- 中核デバイスのハードウェアやソフトウェアの置き換え
- CKMS の性能や機能に関わる増強（例：新しいデバイスの統合やシステム拡張）

これらの変更が、CKMS のセキュリティ要件や運用ポリシーに及ぼす影響を評価するために、適切なスコープでのセキュリティアセスメントを実施することが求められる。

なお、上記のような「大きなシステム変更」が重要な機能等に及んだ場合、完全なセキュリティアセスメントを実施すべきである。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.42	当該 CKMS において、追加のセキュリティアセスメントは次の状況で実施する。 ● HSM、CA ソフトウェア、及び CA サーバの OS におけるメジャーなソフトウェアバージョンの更新時（機能変更を伴う大規模な更新を含む） ● OS セキュリティを支援するソフトウェアの新規導入、ベンダ変更、及びメジャーなバージョンの更新時 ● HSM のリプレイス時 ● CA サーバのリプレイス時
F.43	F.42 に記載した追加セキュリティアセスメントでは次の 3 つを実施する。 ● HSM に関わる第三者認証のレビュー ● CKMS 全体のアーキテクチャレビュー ● CKMS 提供機能に関わる機能テスト及びセキュリティテスト HSM のリプレイス時には、FIPS 140-2/-3 レベル 3 の認証が有効であることを確認する。

④ セキュリティメンテナンスで実行される要求事項を決めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.44	FR11.16	CKMS 設計は、セキュリティを維持するために、実行することが必要な堅牢化アクティビティをリスト化しなければならない。	11.4 節

解説・考慮点

当初は特定のセキュリティレベルを実現していた CKMS であっても、設定が変更されたり新しい脅威が発見されたりすることで、セキュリティレベルが低下することがある。そのため、セキュリティアセスメントとは別に、CKMS のセキュリティを維持・強化するために、堅牢化ガイドラインに従って適切に CKMS のメンテナンスを実施し、必要に応じてアップグレードすることが必要である。セキュリティメンテナンスには、以下の対策例が含まれる。

- CKMS を最新のセキュリティパッチで更新する
- 堅牢化ガイドラインに従ってシステム設定を定期的にレビューする
- 堅牢化ガイドラインに従って CKMS を定期的にテストする
- 更新された堅牢化ガイドラインを適用する
- 定期的なペネトレーションテストを行う

項目 F.44 は、CKMS の設計にあたって、セキュリティメンテナンスで実行される要求事項を明確化することを求めたものである。

本節②の定期的セキュリティアセスメントに記載したように、CKMS 運用中にセキュリティを維持するための日常的な活動に以下のものがある。

a) CKMS 内のデバイスやコンポーネント向けにベンダから提供されたアップデートパッチの適用可否を検討し、必要なパッチの適用や対策技術の導入を検討する
   - パッチ適用時には、CKMS の機能やセキュリティ要件に与える影響を評価し、運用後の動作確認を実施する
   - パッチ適用されない場合、該当する脆弱性に対する対策技術の導入を検討する

b) CKMS 内で取得した実行要求や操作ログを分析して、不正操作や異常な要求がないかを検出・対応する
   - 異常が検出された場合、速やかに原因分析を行い、必要な対策を実施する

c) CKMS エンティティとして権限を有する担当者の変更管理を行い、ACL の設定を正確かつ最新の状態に維持する
   - 担当者の異動やアクセス権変更要求が発生した際に更新を行い、定期的なレビューを実施することで適切なアクセス制御を維持する

これらの活動は、CKMS のセキュリティメンテナンスとして位置づけられ、システムの安全性と運用の堅牢性を確保するために継続的に実施する。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.44	当該 CKMS ではセキュリティメンテナンスとして以下の内容を実施する。 a) HSM、CA ソフトウェア、及び CA サーバの OS に対するセキュリティパッチの適用判断及び適用状況のチェック

	● ベンダからパッチがリリースされた時点でパッチの適用可否を判断し、適用後の動作確認を行う b) CA サーバ及び HSM における操作ログの確認と分析 ● 不正操作や異常な要求がないかを毎日または定期的に確認し、必要に応じて対応を実施する c) CA サーバ及び HSM に登録されたエンティティ及び ACL の最新化チェック ● アクセス権の変更要求が発生した時点や定期的なレビュー時に設定の正確性及び最新性を確認する 上記項目については、帳票を用いるなど棚卸し管理を実現できる仕組みを合わせて検討する。

4.4 CKMS へのアクセスコントロールの危殆化時の BCP 対策

解説・考慮点

本節は、SP 800-130 の 6.8 節に記載されている事項について解説したものである。なお、SP 800-130 の 6.8 節には 8 つの小節があるが、「設計指針」では内容に依存してそれらを 5.3 節、5.7 節、8.1 節、9.4 節に分離して記載してある[15]。

アクセスコントロールには、セキュリティ境界において、認可されたエンティティのみがセキュリティ境界内部に入れるようにするための門番としての役割がある。逆に言えば、アクセスコントロールが危殆化することは、直ちにセキュリティ危殆化につながるリスクがある。本節では、アクセスコントロールが危殆化した場合の対策を取り扱う。

アクセスコントロールの危殆化が検知された場合、鍵情報の危殆化と同様、次のステップを参考に、適切な当事者に危殆化を警告し、望ましくない影響を軽減し、最後にセキュアな状態に復帰することが必要である。

a) その原因及び範囲を決定するために危殆化を評価
b) 鍵情報（暗号鍵やメタデータ）の露出を最小化するために危殆化軽減手段を実行
c) 危殆化の再発を防止するために適切な是正手段を実施
d) CKMS をセキュアな運用状態に復帰させる

本節は一般的な情報システムにおけるアクセスコントロールが危殆化した場合の BCP 対策と共通する部分が多い。一方、CKMS 固有の要素としては管理する鍵情報の危殆化にまで影響が及んだかどうかをアセスメントし、その可能性がある場合には暗号モジュールや鍵情報の危殆化時の対応が必要となることが挙げられる。また、CKMS が管理する鍵を利用する情報システムの重

[15] 暗号鍵管理ガイダンスではそれぞれ以下の節に対応する。設計指針の 5.3 節と 5.7 節はガイダンス Part 1 の 2.3 節と 2.7 節、設計指針の 8.1 節と 9.4 節はガイダンス Part 2 の 3.1 節と 4.4 節である。

要性によってアクセスコントロールの強靭性に関わる要求レベルも様々である。このようなCKMS固有の観点も考慮して、情報システム全般におけるアクセスコントロールに責任を持つチーム（例えば、社内情報システムに関わるインシデントレスポンスチーム：CSIRT）とCKMS管理チームが連携して本節のBCP対策を実施することが望ましい。

① 物理セキュリティの危殆化に対するBCP対策を定めなければならない。

項目	FR番号	Framework Requirementsの内容	SP 800-130
F.45	FR6.123	CKMS設計は、あらゆるCKMSのコンポーネント又はデバイスへの物理セキュリティ侵害がCKMSによって検知されたときに、自動的に通知されるエンティティを明記しなければならない。	6.8.8節
F.46	FR6.122	CKMS設計は、CKMSがどのように暗号モジュール以外のコンポーネント及びデバイスへの認可されない（不正な）物理アクセスから回復するかを明記しなければならない。	6.8.8節
F.47	FR6.124	CKMS設計は、侵害された領域がどのようにセキュアな状態に再確立できるかを明記しなければならない。	6.8.8節

解説・考慮点

CKMSの物理セキュリティ侵害は、暗号鍵又は暗号モジュールの危殆化とは別の危殆化をもたらす可能性がある。一旦セキュリティが侵害されると、侵害された領域全体の完全性が疑わしくなるうえ、新しい暗号鍵及び機微な情報をまた将来危殆化させられるように、領域内のロジックを改ざんしている可能性がある。つまり、暗号鍵又は暗号モジュールの危殆化に対するBCP対策だけでは不十分であるかもしれない。

CKMSの設計にあたって、項目F.45は、物理セキュリティの侵害を検知した時の対応や手続きを明確化することを、F.46及びF.47はBCP対策として物理セキュリティの危殆化からの復旧を行うための手続きや要求事項を明確化することを求めたものである。
なお、物理セキュリティの危殆化に伴う暗号鍵又は暗号モジュールの危殆化の場合には、最初に物理セキュリティの危殆化に対するBCP対策を実施しなければならない。

本ガイダンスの4.1節で定めた物理セキュリティコントロールが危殆化して、設定したエンティティ以外の主体による物理アクセスが検知された場合（あるいは疑われる場合）、通報先や回復の手続きを定めることを要求している。

上記のように、別に定める暗号モジュールや鍵情報への危殆化対策（本ガイダンスの3.1.2節及びガイダンスPart 1の2.7節）と整合をとりながら、これらを補完する対策が求められる。不正な物理アクセスの波及範囲によっては、暗号モジュールや鍵情報の危殆化対策の実行も求められる。

CKMS の物理セキュリティコントロールの回復時には、同様の侵害を発生させないようなコントロール手段の構築と共に、侵害によって保護エリア内にバックドアが仕込まれていないかの検査が求められる。バックドアとしては物理的な仕掛け（例えば、盗聴器やリモートコントロール可能な装置）以外に、CKMS を構成するデバイスにおけるセキュリティコントロールの迂回となる設定変更やマルウェア設置なども考えられる。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。CKMS の物理アクセスコントロールとして、CKMS の設置エリア以外にその外側の建物や敷地へのコントロールもとられる。ここでは、CKMS の設置エリア内への不審者侵入を対象とする。建物や敷地への侵入に関わる検知や物理的アクセスコントロールの回復手順については組織の運用マニュアルに従った対応がされるものとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.45	当該 CKMS において、不審な物理アクセスの検知メカニズムは 4.1 節、F.04 に記載したとおりである。検知後の通報は CKMS 責任者及び CKMS 管理担当に行われる。
F.46	当該 CKMS において、CKMS 設置エリア内への不正な物理アクセスが検知あるいは疑われる場合、侵入経路と侵入範囲の特定、CA サーバや HSM への不正アクセスの痕跡調査、エリア内への不審な装置の有無などを調査する。調査結果に基づいて、侵入経路や侵入範囲に対する物理アクセスコントロールの強化対策を実施して機能回復を進める。その過程で暫定的な措置として、従来の物理アクセスコントロールの代替手段を講じる場合もある。回復時は侵入対策の刷新後に CKMS の再立ち上げを実施する。その際、CA サーバ及び HSM に対して初期化と再インストール、もしくはバックアップからの復元を実施する。
F.47	F.46 に記載したように、物理アクセスコントロールの機能回復及び機能強化に加えて、侵入対策の刷新後に CKMS の再立ち上げを実施する。また、CA サーバや HSM などの暗号モジュール、及びそれらデバイス内部の鍵情報の危殆化対策も影響範囲の調査結果に基づいて実施する。

② コンピュータシステムの危殆化に対する BCP 対策を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.48	FR6.114	CKMS 設計は、CKMS システムハードウェア、ソフトウェア、及びデータに対する認可されない改変を検出するために利用されるメカニズムを明記しなければならない。	6.8.5 節
F.49	FR6.115	CKMS 設計は、CKMS システムハードウェア、ソフトウェア、及びデータに対する認可されない改変からどのように CKMS が回復するのかを明記しなければならない。	6.8.5 節

解説・考慮点

重要なファイルへの改ざんが監視ユーティリティによって検出又はイベントログに表示された場合、当該ファイルは、正当でセキュアであると分かっているセキュアなストレージに保管されたバックアップファイルを使って置き換えるべきである。また、広範囲にわたってソフトウェアが改ざんされた場合、当該ソフトウェアは後述する障害・災害発生時の BCP 対策に記載された手順を準用すべきである。 CKMS の設計にあたって、項目 F.48 は、ハードウェア、ソフトウェア、及びデータに対する改ざんを検知するための要求事項を明確化することを、F.49 は BCP 対策としてハードウェア、ソフトウェア、及びデータに対する改ざんからの復旧を行うための手続きや要求事項を明確化することを求めたものである。

CKMS におけるコンピュータシステムのセキュリティコントロールは 4.1.2 節で定めている。4.1.2 節で定めた各種のセキュリティコントロールが迂回されたり不正な変更がされたりしないように権限の管理や設定変更に関わる監視を行う。主に 4.1.2 ④の監査機能によって、操作ログを元に監視が行われるのが一般的であるが、ログ自体が改ざんされないようにすることが前提となる。

コンピュータシステムにおける危殆化の検出と回復の手順を定めることが要求されている。多層防御のどこまでが無効化されたか危殆化事象の評価を行い、影響範囲に応じた回復手順を定めることが重要である。また、回復手順において再発防止策を検討し、多層防御の強化を図ることも重要である。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.48	当該 CKMS において、コンピュータシステムの危殆化の検知は、CA サーバにおいてセキュアブートにおけるソフトウェアのインテグリティチェックや操作ログの確認による。操作ログは CKMS 管理担当者によって異常な操作ログの有無が定期的に確認される。
F.49	当該 CKMS において、コンピュータシステムの危殆化や障害発生時、災害発生時の復旧対策として、重要な鍵情報や設定情報などは定期的にバックアップを取得しておく。機能回復時にはバックアップデータを利用して再設定を行う。また、危殆化事象の評価を行い、適切な再発防止策をとり、コンピュータシステムのセキュリティコントロールを強化する。なお、CA サーバや HSM などの暗号モジュールの危殆化や内部の鍵情報の危殆化が疑われる場合は、それぞれの危殆化時の対応策を実施する。

③ ネットワークセキュリティコントロールの危殆化に対する BCP 対策を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.50	FR6.116	CKMS 設計は、システムによって使用されるネットワークセキュリティコントロールの危殆化からどのように回復するかを明記しなければならない。特に、 a) CKMS 設計は、それぞれのネットワークセキュリティコントロールデバイスに対して考えられる危殆化シナリオを明記しなければならない。 b) CKMS 設計は、それぞれの想定される危殆化シナリオに対して、この節に記載されたどの軽減技術が採用されるかを明記しなければならない。 c) CKMS 設計は、採用されるあらゆる追加又は代替の軽減技術を明記しなければならない。	6.8.6 節

解説・考慮点

ネットワークセキュリティコントロールの危殆化は CKMS 自体の危殆化につながり得る。以下が危殆化の例である。

- ネットワークセキュリティコントロールデバイスの物理的危殆化
- ネットワークセキュリティコントロールデバイスで使用されるひとつ以上の暗号鍵の危殆化
- ネットワークセキュリティコントロールデバイスを管理するために使用されるひとつ以上の暗号鍵の危殆化
- 危殆化につながるネットワークアーキテクチャの変更（例えば、誰かが VPN 接続されたワークステーションをセキュアでないネットワークに接続し、VPN ワークステーションがイントラネットを攻撃するために使用される）
- 特権ユーザのパスワード（例えば、システム管理者のパスワード）の危殆化
- プラットフォーム OS の危殆化
- ネットワークセキュリティアプリケーション（例えば、ファイアウォール、IDS 等）の危殆化
- プロトコルへの新しい攻撃による危殆化

ネットワークセキュリティコントロールの危殆化の状況によって、取るべき是正措置（軽減措置や回復手段）が異なる。このため、全ての状況において、インシデントを完全に調査し、ネットワークセキュリティコントロールの危殆化に起因して他のシステム及び暗号鍵のどれが危殆化した可能性があるのかを特定する必要がある。

BCP 対策としての復旧策も、個々の危殆化のシナリオごとに用意する必要がある。具体的な対策については SP 800-130、6.8.6 節に記載されている。

項目 F.50 は、CKMS の設計にあたって、BCP 対策としてネットワークセキュリティコントロールの危殆化からの復旧を行うための手続きや要求事項を個々の危殆化のシナリオごとに明確化することを求めたものである。

CKMS におけるネットワークセキュリティコントロールは 4.1.3 節で定めている。これらはネットワーク境界の防御やネットワーク経由での侵入の検知や防御を行うメカニズムであり、上記のように、様々な危殆化シナリオが考えられる。SP 800-130、6.8.6 節に上記の危殆化シナリオに応じた危殆化対応の概要が記載されているので、これを参考に危殆化時の回復手順を定めることが推奨される。

多層防御のどこまでが無効化されたか危殆化事象の評価を行い、影響範囲に応じた回復手順を定めることが重要である。また、回復手順において再発防止策を検討し、多層防御の強化を図ることも重要である。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.50	当該 CKMS におけるネットワークセキュリティコントロールについて、4.1.3 節、F.17 に記載したように、フィルタリング機能を搭載したルータ、ホスト搭載型の IDS、外部からのシステム管理の目的で設置した VPN がある。 a) 危殆化シナリオとして、以下の 3 つを想定する。 ● ルータの物理的危殆化 ● ルータや VPN アクセスポイントを管理する特権ユーザのパスワードの危殆化 ● IDS の危殆化 b) 上記の危殆化シナリオに対する是正措置として、以下を定める。 ● ルータの物理的危殆化の是正措置としては、ルータ製品の置き換えを実施する。フィルタリング機能や処理性能、特権ユーザの認証機能については従来機と同等以上の製品を採用する。 ● ルータや VPN アクセスポイントにおける特権ユーザのパスワード危殆化の是正措置としては、パスワードの置き換えとパスワードエントロピーのチェックを実施する。また、より強固な新たな認証方式があれば是正措置の候補とする（例えば 2 要素認証など）。 ● IDS 危殆化の是正措置としては、セキュリティパッチの適用、及び、同等以上の攻撃検出性能を有するアプリケーション製品への置き換えを検討する。 また、上記是正措置に加えて、CKMS のマルウェアスキャン、インテグリティチェック、ログの確認を含む危殆化事象の評価を行い、CA サーバや HSM などの暗号モジュールの危殆化や内部の鍵情報の危殆化が疑われる場合は、それぞれの危殆化時の対応策を実施する。 c) 上記 a)b)に記載した事項以外に実施する緩和策として、以下を定める。

	● フィルタリングルールや侵入検知シグネチャの設定や更新状況の確認 ● ルータや IDS における通信ログの監視と分析 ● パスワード管理など情報セキュリティの教育及びインシデント対応訓練の実施

4.5 CKMS 設備への障害・災害発生時の BCP 対策

解説・考慮点

本節は、SP 800-130 の 10.1 節から 10.5 節に記載されている事項について解説したものである。
CKMS の障害は情報へのアクセスの停止につながる可能性がある。障害が発生する原因としては、システム故障のほか、災害等による物理的損害や公共サービスの供給停止などがある。本節では、災害等を含めたあらゆる事象発生時にどのように運用継続性を達成するのかについて取り扱う。

なお、本節では鍵情報の喪失・破損からの復旧を想定しており、各検討項目の内容が検討項目 B.71 及び B.72 の内容（「設計指針」の 5.6 節[16]）に矛盾しないようにすべきである。また、流出や暴露などの鍵情報の危殆化からの復旧は想定していないことに注意されたい。鍵情報の危殆化からの復旧に関しては、「設計指針」における 5.7 節及び 9.4 節[17]を参照して定める必要がある。

① CKMS 設備への物理的損害発生時における BCP 対策を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.51	FR10.1	CKMS 設計は、必要な環境的、火災、及び物理的なアクセスコントロール保護メカニズム、及び損害からの基幹及び全てのバックアップ設備への回復手続きを明記しなければならない。	10.1 節

解説・考慮点

設備への物理的損害発生を想定したバックアップ及び回復設備は、保護されるデータ及び CKMS 運用の価値と機微度にふさわしいレベルで設計、実装及び運用されるべきである。例えば、風水害や地震は環境リスクであり、火災は環境リスク及び設備設計に依存したリスクの両方に該当する。

[16] 暗号鍵管理ガイダンスでは Part 1 の 2.6 節に対応する。
[17] 暗号鍵管理ガイダンスではそれぞれ Part 1 の 2.7 節と Part 2 の 4.4 節に対応する。

項目 F.51 は、CKMS の設計にあたって、CKMS 設備への物理的損害発生を想定した BCP 対策として準備すべき要求事項、及び、復旧を行うための手続きや要求事項を明確化することを求めたものである。なお、物理的損害には、システム故障だけでなく、風水害や地震、火災などのあらゆるリスクに起因するものも含む。

上記のように、地震及び水害などの自然災害、並びに火災及び事故などの人為災害が発生して CKMS を収容した施設やエリアに対して損害が生じ、CKMS の運用が困難となる、物理的アクセスコントロールに影響が及ぶなどの事態が考えられる。予めそうした状況を想定して予防措置や回復の手順を定めておくなどの BCP 対策を検討することを求めている。

CKMS が管理する鍵情報を利用する情報システムや対象製品におけるサービス継続の重要性や必要性を踏まえて、BCP 対策を検討することが重要である。BCP に関わる一般的な予防措置としては、冗長系を用意すること、内部の重要な情報をバックアップすること、及び保険に加入することなどがある。冗長系への切り替え及びバックアップ情報からの復旧などの訓練を定期的に行うことも重要である。また、本節の災害対応については、事前に施設レベルで準拠する耐震基準、火災・風水害の防災基準の準備状況、及び非常用電源の有無などを確認しておくことも具体的な対応項目として挙げられる。

施設への災害に関わる BCP 対策について、以下のような文献が参考になる。

- 「データセンター　セキュリティ　ガイドブック」、日本データセンター協会
- 「金融機関等コンピュータシステムの安全対策基準・解説書」、金融情報システムセンター

また、保護されるデータ及び CKMS 運用の価値と機微度にふさわしいレベルの決定にあたっては、FIPS 199 及び FIPS 200 が参考になる。

CKMS の施設レベルの災害発生に伴って、物理アクセスコントロールが無効となった結果、正規エンティティ以外が管理エリアにアクセス可能となるなど、各種のセキュリティコントロールが危殆化した状態と同様の状況も生じ得る。CKMS の復旧においては、このような点も踏まえた手順を検討することが望ましい。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.51	CKMS が提供する機能のうち、IoT 製品向け証明書の発行については、IoT 製品の製造設備の構成要素の一つと捉えることができる。また、証明書の失効機能については、CKMS が機能を停止している間は新たな失効処理を実行できないが、機能停止が短期間であれば IoT 製品の運用に与える影響は限定的である。以上の理由から、当該 CKMS の災害対策は IoT 製品の製造設備と同程度の基準とする。なお、IoT 製品の生産においては同一の機能を持つ製造ラインが、複数拠点に存在する。ある拠点が稼働できない場合は、他の拠点にて生産を補うことは可能であり、それぞれの製造拠点が代替の製造拠点を構成しているため、以下の方針とする。 1. 基本方針 災害復旧対策を目的としたプライベート CA システム単位の冗長化は行わない。ただし、製造工場において障害対策として以下を実施する。CA サーバにおけるディスクの冗長化、CA サーバ及びルータについて予備デバイスの準備、HSM に対して障害発生時に代替機が提供される保守契約の締結。 2. バックアップの取得と保管 CA サーバの設定、ルータの設定、並びに HSM 内部の鍵情報及び設定のバックアップを、各デバイスの設定及び鍵情報の変更前後に取得する。バックアップは、製造工場のバックアップ設備と同じ基準を満たす場所（例えば、幾つかの製造拠点）にも保管する。 3. 物理セキュリティ危殆化時の対策 地震・火災等により物理セキュリティが危殆化した際は、鍵情報及び CKMS へのアクセスコントロールが危殆化したものとして対応する。具体的な対策は暗号鍵管理ガイダンス Part1「2.7 鍵情報の危殆化時の BCP 対策」及び本ガイダンス「4.4 CKMS へのアクセスコントロールの危殆化時の BCP 対策」を参照する。 4. 復旧手順の準備 災害が発生した製造拠点における復旧手順を準備する。 ● 予備デバイス及び代替機への必要なパッチ及びアップデートの適用 ● 予備デバイス及び代替機への初期状態からのソフトウェアのセットアップ ● 予備デバイス及び代替機へのバックアップからの設定及び鍵情報のリカバリー ● 鍵情報及び CKMS へのアクセスコントロール危殆化からの回復 ● CKMS の運用再開 5. 復旧訓練の実施 災害シナリオを想定した復旧訓練を年 1 回実施する。

② 公共サービス（電気、水道、下水道、空調等）の停止時における BCP 対策を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.52	FR10.2	CKMS 設計は、基幹及び全てのバックアップ設備に対する、電気、水道、衛生、暖房、冷房、及び空気清浄に関する推奨要求値だけでなく最小要求値についても明記しなければならない。	10.2 節

解説・考慮点

CKMS の継続的な可用性を保証するためには、通常運用時及び非常時において、全ての CKMS デバイスの要求を満たすのに十分な電力が基幹及び全てのバックアップ CKMS 設備で利用可能であるように準備されておく必要がある。例えば、同じ影響を受けないようにするため、基幹設備とバックアップ設備とは別系統の独立した電力線から電力供給を受けることが必要である。

項目 F.52 は、CKMS の設計にあたって、公共サービス（電気、水道、下水道、空調等）の停止を想定した BCP 対策として準備すべき代替手段への要求事項を明確化することを求めたものである。最小要求値とは、公共サービスの停止に伴って代替手段が切り替わった時に、CKMS の継続的な可用性を保証するために最低限確保することが必要な電力や水道、空調などの容量のことである。

上記のように、CKMS の運用に必要な電気及び水道などの公共サービスの停止を想定した BCP 対策に関わる項目である。公共サービスの停止によって空調の稼働にも影響が及ぶ。本節①で述べたように、災害が原因となり公共サービスが停止する場合もある。電力については、自家発電設備や UPS など蓄電設備を設置する対策が考えられる。また、CKMS への電力供給の推奨要求値や最小要求値を明確にすることで、準備すべきバックアップ電源の選定基準を定める。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。以下に記載した最小要求値は仮想的な数値であることに注意されたい。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.52	F.51 と同様に、当該 CKMS は IoT 製品の製造設備と同程度の対策を必要とする。公共サービス停止時の対策は、製造工場の対策を前提とし、製造工場がその最小保証値を維持できない場合には安全にシステムを停止し、公共サービス復旧後に CKMS を再稼働する。ただし、CKMS 単体の対策として、製造設備全体での対策である自家発電設備に加え、短時間の電力供給停止及び電源異常（過電圧・過負荷・雷など）への対応として UPS を設置する。UPS は CA サーバ、HSM、ルータへの短時間の電力供給が可能であればよいため、以下の方針とする。 ● 1500VA、30 分の電力供給が可能な UPS を選定する

③ 通信及び計算機能の機能停止時における BCP 対策を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.53	FR10.3	CKMS 設計は、ユーザ、エンタープライズ、及び CKMS アプリケーションによる予測されるニーズに見合うサービスの運用継続を保証するために、設計内に存在し、かつ運用中に利用可能であることを要求される通信及び計算機能の冗長性を明記しなければならない。	10.3 節

解説・考慮点

CKMS の高可用性を保証するためには、必要な機能を実行しユーザが要求するサービスを提供するために十分な通信及び計算能力を必要とする。このため、もともと CKMS には冗長な通信設備等がバックアップとして設置されることも多い。一方、非常時にはこのバックアップ手段が代替手段として活用できる。

項目 F.53 は、CKMS の設計にあたって、ユーザニーズの増大への対応の他、通信及び計算機能の機能停止を想定した BCP 対策としても利用可能な代替手段への要求事項を明確化することを求めたものである。

上記のように、CKMS の運用に必要な通信及び計算機能の可用性に関わる項目である。通信に関しては、障害に備えて冗長性を備えたネットワーク構成や回線の二重化といった対策が採用される。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.53	当該 CKMS の通信は製造工場内の LAN 及びインターネット接続回線を使用する。また、当該 CKMS の計算機能の冗長性については CA サーバと HSM、ルータが対象となる。以下の方針とする。 1. 通信の冗長化 LAN へは平時には有線ケーブルを使用して接続し、バックアップとして無線アクセスポイントを用意する。有事に備え、無線アクセスポイントへの切り替え手順を定める。 インターネット接続回線は、施設全体で通信事業者と回線二重化の障害対策がとられている。そのため、CKMS 独自の対策は実施しない。 2. 計算機能の冗長化

	CA サーバ及びルータの予備デバイスを用意し、製造工場内に保管する。HSM は障害発生時に代替機が提供される保守契約を結ぶ。 3. バックアップの取得と保管 CA サーバ、ルータの設定、並びに HSM 内部の鍵情報及び設定のバックアップを、設定及び鍵情報の変更前後に取得する。バックアップは、製造工場内に保管する。 4. 復旧手順の準備 製造工場において以下の内容を実施するための復旧手順を準備する。 ● 予備デバイス及び代替機への必要なパッチ及びアップデートの適用 ● 予備デバイス及び代替機への初期状態からのソフトウェアのセットアップ ● 予備デバイス及び代替機へのバックアップからの設定及び鍵情報のリカバリー ● CKMS の運用再開 5. 復旧訓練の実施 障害シナリオを想定した復旧訓練を年 1 回実施する

④ ハードウェア障害発生時における BCP 対策を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.54	FR10.4	CKMS 設計は、バックアップの方策、及びハードウェアコンポーネント及びデバイスの障害からの回復のための方策を明記しなければならない。	10.4 節

解説・考慮点

CKMS は情報管理システムのセキュアな運用にとって極めて重要であるため、CKMS コンポーネント及びデバイスのハードウェア障害の影響は最小限に抑えることが望ましい。そのためには、例えば、同じ故障を引き起こすことがないようにするため、基幹システムからの独立性を持っているバックアップシステムを常時スタンバイしておくといった対策がある。
ただし、ハードウェア障害からの回復が容易でありスピードもある対策は一般にコストがかかるものであり、CKMS の設計において冗長性と対策コストとの間の最適なトレードオフを見出すことが必要である。

項目 F.54 は、CKMS の設計にあたって、システムハードウェア障害発生を想定した BCP 対策としても準備すべき要求事項、及び、復旧を行うための手続きや要求事項を明確化することを求めたものである。

CKMS を構成するデバイスにハードウェア障害が発生した場合、その影響を最小限に抑えるための項目である。冗長性を持たせることを基本方針とする。メインのハードウェアとバックアップとなるハードウェアを常時稼働させ、障害発生時に速やかにバックアップ系に移行する方法と、バックアップ系を常時稼働させず、メイン系の状態を定期的に同期させる方法がある。バックアップ系はメイン系から独立性を持つことが不可欠であり、同一の障害がメイン系とバックア

ップ系の両方に影響しないように設計することが重要である。なお、障害発生時の影響を最小限に抑えようとすれば、その分コストが高くなるというトレードオフがある。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.54	F.53 に記載のとおり、当該 CKMS では CA サーバとルータは製造工場内に予備デバイスを準備するが、HSM は障害発生時に代替機が提供される保守契約を結ぶものとする。バックアップ取得と復旧の手順は次のようになる。 ● CA サーバ及びルータ ソフトウェア及び設定情報を変更する都度、バックアップを取得し、予備機へ同期する。復旧の際は予備機を起動して予備機に切り替える。 ● HSM 鍵情報、設定情報及びソフトウェアを変更する都度、バックアップを取得する。復旧の際は代替機を取り寄せ、バックアップからリカバリーする。

⑤ ソフトウェア障害発生時における BCP 対策を定めなければならない。

項目	FR 番号	Framework Requirements の内容	SP 800-130
F.55	FR10.5	CKMS 設計は、システムソフトウェアの正しさを検証するために、CKMS によって提供されている全ての技術を明記しなければならない。	10.5 節
F.56	FR10.6	CKMS 設計は、ソフトウェアがメモリにロードされたときにソフトウェアの改変又は破損を検知するために、CKMS によって提供される全ての技術を明記しなければならない。	10.5 節
F.57	FR10.7	CKMS 設計は、バックアップ及び重大なソフトウェア障害からの回復のための方策を明記しなければならない。	10.5 節

解説・考慮点

ソフトウェア障害の原因としては、「（製造時の）ソフトウェアバグ」と「（実行時の）予期せぬソフトウェアの破損」がある。前者のような多くのソフトウェア障害は、良好な確立されたプログラミング実践を使用してコードを書くことで防ぐことができる。一方、後者のような、コードが破損する障害は可能な限り速やかに検知されるべきである。これには、マルウェア感染も含まれる。

ソフトウェア障害はいずれ起きるとの仮定の下で運用すべきであり、その対策として、完全なセキュア状態のシステムバックアップが定期的に作成され、最新の CKMS のセキュア状態がリロードされて修復され、CKMS が運用可能な状態に復旧できるようにすることが推奨される。

CKMS の設計にあたって、項目 F.55 はソフトウェア障害を発生させないための対策としての要求事項を明確化することを、F.56 はソフトウェア障害の原因となるリスクを検知するための要求事項を明確化することを、F.57 はソフトウェア障害発生を想定した BCP 対策としても準備すべき要求事項、及び、復旧を行うための手続きや要求事項を明確化することを求めたものである。

上記のように、本節は、ソフトウェア障害に関わる緩和策、及びソフトウェア障害を起因とする障害対策に関わる要件である。ソフトウェアの開発工程におけるバグや脆弱性の作り込みを低減する手法については、製品セキュリティと呼ばれる領域でベストプラクティスがある。セキュリティに配慮したコーディング規約（セキュアコーディング）の適用やソースコードの静的解析などが該当する。また、既知脆弱性の検査やファジングテストを実施することも推奨される。例えば、「脆弱性対処に向けた製品開発者向けガイド（IPA）」に解説がある。

ソフトウェアの破損を検知する具体的な手法としては、ソフトウェアのインテグリティチェックや既知解テストなどが挙げられる。

ソフトウェアのバグや脆弱性を完全に除去することや、ソフトウェア破損を実行前に完全に検出することは困難である。そのため、ソフトウェア障害を起因とした障害への対策を検討しておくことが重要である。障害発生時に発生前の正常な状態にリカバリーできるように備える必要がある。そのためには、ソフトウェアや設定情報が正常かつ安全な状態にあるときに取得したバックアップを保存しておくことが有効な緩和策となる。

《トイモデルと記載例》

本節のトイモデルも 1 章に示した、IoT 製品（家電想定）向けプライベート CA システムとする。

IoT 製品（家電想定）向けプライベート CA システムにおける記載例

F.55	自社で開発するソフトウェアについては、バグや脆弱性の作り込みを低減するセキュアコーディング手法を適用し開発する。CKMS の本格的な運用開始前にテストを実施し、機能が正しく動作するか、不正な変更や故障が発生していないか確かめる。また、使用するソフトウェア（CA サーバ、ルータ、HSM にて動作する OS やアプリケーションなど）について、バグや脆弱性に起因したパッチ及びソフトウェアアップデートが提供された場合には、リリースノートを確認する等して修復が保証されていることを確認し、パッチ及びソフトウェアアップデートを適用する。適用後には、運用開始前に CKMS の運用に悪影響がないことをテストし確認する。
F.56	CA サーバはセキュアブートによりソフトウェアの改変及び破損を検知する。また、HSM は適宜自己テストを実行することによりソフトウェアの改変及び破損を検知する。

F.57	ソフトウェアや設定情報が正常かつ安全な状態にあるときにバックアップを取得する。 バックアップから CKMS を回復した後、CKMS の運用を再開する前に、CKMS の運用に悪影響がないことをテストし確認する。 なお、重大なソフトウェア障害から回復する場合、そのソフトウェア障害は分析され修復が保証されたものである必要がある。

Appendix 参考資料一覧

本文に記載した参考資料、規格、制度、資格、技術論文等を一覧としてまとめる。概ね本文での記載順に、章を単位に記載した。章の間で重複するものもある。

■ 1 章　はじめに

- 「政府機関のサイバーセキュリティ対策のための統一基準（令和 5 年度版）」、NISC
  https://www.nisc.go.jp/pdf/policy/general/kijyunr5.pdf
- 「CRYPTREC 暗号リスト（電子政府推奨暗号リスト）」、CRYPTREC
  https://www.cryptrec.go.jp/list.html
- 「暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準」、CRYPTREC
  https://www.cryptrec.go.jp/list.html
- 「暗号鍵設定ガイダンス」、CRYPTREC
  https://www.cryptrec.go.jp/op_guidelines.html
- 「暗号鍵管理システム設計指針（基本編）」、CRYPTREC
  https://www.cryptrec.go.jp/op_guidelines.html
- 「暗号鍵管理ガイダンス Part 1（2023 年 5 月発行）」、CRYPTREC
  https://www.cryptrec.go.jp/op_guidelines.html
- NIST SP 800-130（A Framework for Designing Cryptographic Key Management Systems）
  https://csrc.nist.gov/pubs/sp/800/130/final

■ 2 章　暗号鍵管理システム（CKMS）の設計原理と運用ポリシー

- RFC 3647（Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework）
  https://www.rfc-editor.org/rfc/rfc3647
- NIST SP 800-57 Part 2（Recommendation for Key Management: Part 2 – Best Practices for Key Management Organizations）
  https://csrc.nist.gov/pubs/sp/800/57/pt2/r1/final
- 政府認証基盤 GPKI（Government Public Key Infrastructure）
  https://www.gpki.go.jp/
- PKCS #11（Cryptographic Token Interface Base Specification）
  https://www.oasis-open.org/2023/08/10/two-pkcs-11-oasis-standards-published/
- PKCS #10（Certification Request Syntax Specification）
  https://www.rfc-editor.org/rfc/rfc2986
- RFC 5280（Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile）
  https://www.rfc-editor.org/rfc/rfc5280
- FIPS 140-2/-3
  https://csrc.nist.gov/pubs/fips/140-2/upd2/final

https://csrc.nist.gov/pubs/fips/140-3/final

- ISO/IEC 15408
  https://www.iso.org/standard/72891.html
  https://www.ipa.go.jp/security/jisec/about/kijun.html
- CMVP 認証
  https://csrc.nist.gov/Projects/Cryptographic-Module-Validation-Program
- 政府情報システムのためのセキュリティ評価制度（クラウドサービスの評価認証制度）ISMAP
  https://www.ismap.go.jp/csm?id=csm_ismap_index
- セキュリティ要件適合評価及びラベリング制度（JC-STAR）
  https://www.ipa.go.jp/security/jc-star/index.html
- サイバーレジリエンス法、欧州
  https://www.cyberresilienceact.eu/
- サイバーセキュリティ法、中国
  http://www.npc.gov.cn/zgrdw/npc/xinwen/2016-11/07/content_2001605.htm
- GDPR（General Data Protection Regulation）、欧州
  https://eur-lex.europa.eu/eli/reg/2016/679/oj
- 電子署名法
  https://www.digital.go.jp/policies/digitalsign
- 「暗号強度要件（アルゴリズム及び鍵長）設定基準」、CRYPTREC
  https://www.cryptrec.go.jp/list.html
- 「注意喚起情報」、CRYPTREC
  https://www.cryptrec.go.jp/er.html

■ 3 章　暗号鍵管理デバイスへのセキュリティ対策

- FIPS 140-2/-3
  https://csrc.nist.gov/pubs/fips/140-2/upd2/final
  https://csrc.nist.gov/pubs/fips/140-3/final
- ISO/IEC 15408
  https://www.iso.org/standard/72891.html
  https://www.ipa.go.jp/security/jisec/about/kijun.html
- PKCS #11（Cryptographic Token Interface Base Specification）
  https://www.oasis-open.org/2023/08/10/two-pkcs-11-oasis-standards-published/
- Shamir の秘密分散法
  https://dl.acm.org/doi/10.1145/359168.359176
- CMVP 認証
  https://csrc.nist.gov/Projects/Cryptographic-Module-Validation-Program
- CAVP 認証
  https://csrc.nist.gov/Projects/cryptographic-algorithm-validation-program

■ 4 章　暗号鍵管理システム（CKMS）のオペレーション対策

- 「データセンター セキュリティ ガイドブック」、日本データセンター協会
  https://www.jdcc.or.jp/topics/127/
- 「金融機関等コンピュータシステムの安全対策基準・解説書」、金融情報システムセンター
  https://www.fisc.or.jp/publication/guideline_pdf.php
- FIPS 140-2/-3
  https://csrc.nist.gov/pubs/fips/140-2/upd2/final
  https://csrc.nist.gov/pubs/fips/140-3/final
- CMVP 認証
  https://csrc.nist.gov/Projects/Cryptographic-Module-Validation-Program
- 「セキュリティ設定共通化手順 SCAP（Security Content Automation Protocol）概説」、IPA
  https://www.ipa.go.jp/security/vuln/scap/scap.html
- NIST SP 800-207（Zero Trust Architecture)
  https://csrc.nist.gov/pubs/sp/800/207/final
- 「脆弱性対処に向けた製品開発者向けガイド」、IPA
  https://www.ipa.go.jp/security/guide/vuln/forvendor.html
- 「ソフトウェア管理に向けた SBOM の導入に関する手引」、経済産業省
  https://www.meti.go.jp/press/2023/07/20230728004/20230728004.html
  https://www.meti.go.jp/policy/netsecurity/vendor.html#id05
- CISSP (Certified Information Systems Security Professional)
  https://japan.isc2.org/cissp_about.html
- 情報処理安全確保支援士（Registered Information Security Specialist, RISS）
  https://www.ipa.go.jp/jinzai/riss/index.html
- CISM (Certified Information Security Manager)
  https://www.isaca.org/credentialing/cism
- FIPS 199（Standards for Security Categorization of Federal Information and Information Systems）
  https://csrc.nist.gov/pubs/fips/199/final
- FIPS 200（Minimum Security Requirements for Federal Information and Information Systems）
  https://csrc.nist.gov/pubs/fips/200/final

発行日　2025 年 x 月 x 日　　第 1 版発行

発行者

・〒113-6591
東京都文京区本駒込二丁目 28 番 8 号
独立行政法人 情報処理推進機構
（セキュリティセンター　技術評価部　暗号グループ）
INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN
2-28-8 HONKOMAGOME, BUNKYO-KU
TOKYO, 113-6591 JAPAN

・〒184-8795
東京都小金井市貫井北町四丁目 2 番 1 号
国立研究開発法人 情報通信研究機構
（サイバーセキュリティ研究所　セキュリティ基盤研究室）
NATIONAL INSTITUTE OF
INFORMATION AND COMMUNICATIONS TECHNOLOGY
4-2-1 NUKUI-KITAMACHI, KOGANEI
TOKYO, 184-8795 JAPAN

CRYPTREC
Cryptography Research and Evaluation Committees

耐量子計算機暗号（PQC）への対応について

2025年3月25日

CRYPTREC事務局
（デジタル庁、総務省、経済産業省）

耐量子計算機暗号（PQC）への対応について

- 2020年度に「量子コンピュータ時代に向けた暗号の在り方検討タスクフォース」でCRYPTREC暗号リストへのPQC掲載を見据えて検討したが、「PQCは、多数の方式が提案され安全性を検討している段階で、利用実績等に言及できる段階ではない」ことから、CRYPTREC暗号リストに組み込まず、別途ガイドラインを作成することとした。
- 上記の検討から約4年が経過。政策的な観点（各国取組との調和性、国内における議論の高まりなど）を踏まえれば、上記の方針を見直すべき時期にあるのではないか。
- 今後、安全性等が確認されたPQCを推奨候補リストに順次掲載できるよう準備を始めてはどうか。
  - 米欧をはじめ、複数の国においてPQCへの移行に関する方針や推奨アルゴリズムに関する情報が発出されている。我が国政府におけるPQC移行の旗振り・総合調整役は定まっておらず、移行方針もないが、CRYPTRECリスト掲載に向けたPQCの技術的検討は、移行方針の検討と両輪で進めるべきもの。サイバー空間における経済安全保障の観点からも、PQCのリスト掲載を遅滞なく行うことがCRYPTRECに求められている。
  - 機動的なスケジュールを前提とすれば、まずは諸外国において多くの専門家による検証を経て決定された方式（例えばFIPS 203 (ML-KEM)、FIPS204（ML-DSA）、FIPS 205（SLH-DSA）など）の安全性評価・実装性能評価を先行し、その後、国産PQCを含めた他のアルゴリズムの取扱を順次検討し、追加の評価を実施してはどうか。なお、CRYPTRECとしてPQCの公募を行うことも考えられるが、人的・予算的リソースを勘案すると直ちに実施することは困難であるため、今後、公募のメリット・デメリットを精査しつつ引き続き検討する。

資料６

2025 年度暗号技術評価委員会活動計画(案)

1. 活動目的

CRYPTREC 暗号リストに掲載されている暗号技術や電子政府システム等で利用される暗号技術の安全性維持及び信頼性確保のために安全性及び実装に係る監視及び評価を行う。

2. 活動概要

(1) 暗号技術の安全性及び実装に係る監視及び評価

以下の通り、暗号技術の安全性に係る監視・評価 及び 実装に係る技術の監視・評価を実施する。

① CRYPTREC 暗号リストの監視

国際会議等で発表される CRYPTREC 暗号リストの安全性及び実装に係る技術（暗号モジュールに対する攻撃とその対策も含む）に関する監視を行い、会議や ML を通して報告する。

② 電子政府推奨暗号リストからの運用監視暗号リストへの降格、並びに、推奨候補暗号リスト及び運用監視暗号リストからの危殆化が進んだ暗号の削除に係る検討

CRYPTREC 暗号リストの安全性に係る監視活動を継続的に行い、急速に危殆化が進んだ暗号技術やその予兆のある暗号技術の安全性について評価を行う。また、リストからの降格や削除、注釈の改訂が必要か検討を行う。

③ CRYPTREC 注意喚起レポートの発行

CRYPTREC 暗号リストの安全性及び実装に係る技術の監視活動を通じて、国際会議等で発表された攻撃等の概要や想定される影響範囲、対処方法について早期に公開することが望ましいと判断された場合、注意喚起レポートを発行する。

④ 推奨候補暗号リストへの新規暗号（事務局選出）の追加に係る検討

標準化動向に鑑み電子政府システム等での利用が見込まれると判断される暗号技術の追加を検討する。

⑤ 新技術等に関する調査及び評価

将来的に有用になると考えられる技術やリストに関わる技術について、安全性・性能評価を行う。必要に応じて、暗号技術調査ワーキンググループによる調査・評価、または、外部評価による安全性・性能評価などを行う。

- 2024 年度第 2 回の暗号技術評価委員会以後に、NIST の PQC 標準化において第 4 ラウンドにより HQC が選定された。さらに、署名方式については、NIST での選定

が続いている。これらの状況を考慮し、2024 年度の暗号技術調査ワーキンググループ（耐量子計算機暗号）の委員からも、2025 年度以降もワーキンググループを設置する意見が出ていることから、引き続き、暗号技術調査ワーキンググループ（耐量子計算機暗号）を設置して、耐量子計算機暗号に関する最新動向を把握する。

- 耐量子計算機暗号の社会的動向を踏まえ、NIST 標準として公開された FIPS-203,204,205 について、安全性評価・実装性能評価関連の活動を開始する。
- 「素因数分解の困難性に関する計算量評価」、「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新についても当該ワーキンググループで検討し、更新を行う。

(2) 暗号技術の安全な利用方法に関する調査（技術ガイドラインの整備、学術的な安全性の調査・公表等）

暗号技術を利用する際の技術面での注意点に関する調査、新技術の安全性・性能に関する調査・評価を行う。

3. 活動スケジュール

暗号技術評価委員会は年 2 回の開催を予定する。

回	開催日	議案
2025 年度 第 1 回	2025 年 6 月中旬 ～7 月上旬	● 暗号技術評価委員会活動計画の具体的な進め方に関する審議 ● 暗号技術調査ワーキンググループの活動計画(案)の審議 ● 外部調査・評価に関する審議
2025 年度 第 2 回	2026 年 2 月中旬 ～3 月上旬	● 暗号技術評価委員会活動報告(案)についての審議 ● 暗号技術調査ワーキンググループの活動報告(案)の審議、及び、ガイドライン(案)に関する審議 ● 外部調査・報告に関する審議

以上

資料７

2025 年度 暗号技術活用委員会活動計画（案）

1. 活動目的

暗号技術活用委員会では、情報システム全般のセキュリティ確保に寄与することを目的として、暗号の取り扱いに関する観点から、運用ガイドライン／ガイダンスの作成を行う。

2. 活動概要

（1） クラウドにおける鍵管理ガイダンスの作成

暗号鍵管理ガイドラインの拡充活動の一環として、クラウドサービスを利用したシステム構築を対象とした暗号鍵管理ガイダンスの作成を目標に、クラウド鍵管理ガイダンス WG を設置する。作成するガイダンスの位置づけや内容を確定し、遅くとも 2026 年度末までにガイダンスの完成を目標とする。

（2） 暗号鍵管理ガイドライン／ガイダンスの修正

暗号鍵管理ガイドライン／ガイダンスの利用価値向上を目指して、暗号鍵管理ガイダンスの合冊化検討、及び暗号鍵管理システム設計指針（基本編）の修正を行う。

（3） 暗号利活用のための新たなガイドライン／ガイダンスの検討

暗号技術活用委員会で発行した過去のガイドライン／ガイダンスの更新を含めて、新たなガイドライン／ガイダンスの作成について検討する。

（4） その他

その他、暗号技術の活用に係る状況の変化に応じ、暗号技術検討会で必要と位置づけられた活動の実施を検討する。

ただし、2025 年 3 月の暗号技術検討会の審議次第では、以下の活動項目を追加する。

- **耐量子計算機暗号の扱いに係る検討**

  昨今の耐量子計算機暗号をめぐる社会的動向を踏まえ、耐量子計算機暗号の取扱い基準や運用ガイドライン／ガイダンスにおける耐量子計算機暗号の位置づけ・記載内容等についての検討を開始する。

3. 活動スケジュール

暗号技術活用委員会の開催日程・議題については、以下のとおり、年 2 回の委員会開催を予定する。また、必要に応じて追加の委員会開催やメール審議を実施する。

回	開催日	議案（予定）
第1回	2025年6月下旬 ～7月上旬	■ 2025年度暗号技術活用委員会活動計画の確認 ■ クラウド鍵管理ガイダンスWG活動計画の審議 ■ 暗号鍵管理ガイドライン／ガイダンス修正についての検討 ■ （耐量子計算機暗号の扱いに関する検討）
第2回	2026年2月下旬 ～3月上旬	■ クラウド鍵管理ガイダンスWG活動成果の審議 ■ 暗号鍵管理ガイドライン／ガイダンス修正案の審議 ■ （耐量子計算機暗号の扱いに関する検討） ■ 2025年度暗号技術活用委員会活動報告案について

以上

資料８

暗号技術検討会
２０２４年度　報告書

２０２５年３月

目次

１．はじめに

情報通信技術の急速な発展により、自動車、家電、医療、農業、工場など様々な分野で、あらゆるモノがネットワークに繋がるIoT社会が到来し、サイバー空間と実空間の高度な融合により、多様なニーズにきめ細やかに対応したモノやサービスを提供できる社会への産業構造の変化が進みつつある。一方で、IoT機器の普及に伴うサイバー攻撃の起点の増加や、サイバー攻撃自体の巧妙化・複雑化が続く中で、サイバー攻撃の影響が実空間にまで到達するリスクも増していくと考えられる。このような産業構造、社会の変化に伴うサイバー攻撃の脅威の増大に対応したセキュリティ確保が求められる中、暗号技術は既に様々な製品やサービスに組み込まれ、セキュリティを支える基盤技術として欠かせないものであるが、IoT機器から得られる大量のデータの流通・連携を支える上でも、その重要性は一層増すと考えられる。

このような社会の変化に伴い、CRYPTRECにおける、暗号アルゴリズムの安全性の評価及び監視を通じたセキュリティ確保、そして情報システム全体のセキュリティ確保に向けた暗号技術の利活用のための情報提供等の取組が果たすべき役割も大きくなっている。

2024年度は、暗号技術検討会の活動として、耐量子計算機暗号（PQC）への対応についての承認等を行った。そして、各委員会の活動として、暗号技術評価委員会では、同委員会の下に設置した暗号技術調査WG（耐量子計算機暗号）において、耐量子計算機暗号に関する調査報告書及びガイドライン(いずれも2024年度版)並びに量子コンピュータによる共通鍵暗号の安全性への影響に関する調査報告書（2024年度版）を作成したとともに、「素因数分解の困難性に関する計算量評価」及び「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図を更新した。暗号技術活用委員会では、同委員会の下に設置した暗号鍵管理ガイダンスWGにおいて、2022年度に発行した「暗号鍵管理ガイダンス」の追補版として「暗号鍵管理ガイダンスPart 2」を作成したとともに、クラウドにおける鍵管理ガイダンスの検討を行った。これらの2024年度の活動の詳細については、国立研究開発法人情報通信研究機構及び独立行政法人情報処理推進機構が取りまとめる「CRYPTREC Report 2024」を参照いただきたい。

今後も暗号技術を用いた情報システム及び情報社会全体のセキュリティ確保のために、成果物の検討や情報発信等を行っていく所存である。

末筆ではあるが、暗号技術検討会及び関係委員会等に御協力いただいた構成員、オブザーバの方々をはじめ、関係者の皆様に心から謝意を表する次第である。

2025年3月

暗号技術検討会
座長　松本　勉

２．暗号技術検討会開催の背景及び開催状況

２．１．暗号技術検討会開催の背景

暗号技術は情報セキュリティの基盤技術であり、その安全性を暗号技術の専門家により技術的、専門的な見地から客観的に評価することが重要である。電子政府のセキュリティ確保のためには、安全性に優れた暗号技術を利用することが不可欠である。

このため、総務省及び経済産業省は、客観的な評価により安全性及び実装性に優れると判断される暗号技術をリスト化し、各府省に対してその利用を推奨することにより、高度な安全性と信頼性に支えられ、国民が安心して利用できる電子政府の構築に貢献することを目指し、2001年5月に最初の暗号技術検討会を開催した。

その後、2021年のデジタル庁発足に伴いデジタル庁が加わり、デジタル庁、総務省及び経済産業省は、電子政府推奨暗号等の安全性を評価・監視し、暗号技術の更なる普及促進を行うため、継続的に暗号技術検討会を開催している。

暗号技術検討会での検討を経て、2003年2月に策定された電子政府推奨暗号リストは、2013年3月にCRYPTREC暗号リストとして改定され、2023年3月に再改定された（2024年5月に更新）。

２．２．CRYPTRECの体制

CRYPTRECとは、Cryptography Research and Evaluation Committeesの略であり、デジタル庁、総務省及び経済産業省が共同で開催する暗号技術検討会（座長：松本勉国立研究開発法人産業技術総合研究所フェロー、横浜国立大学上席特別教授）と、国立研究開発法人情報通信研究機構（NICT）及び独立行政法人情報処理推進機構（IPA）が共同で開催する委員会から構成される暗号技術評価プロジェクトをいう。

2024年度は、暗号技術検討会では、耐量子計算機暗号（PQC）への対応についての承認等を行った。暗号技術評価委員会では、同委員会の下に設置された暗号技術調査WG（耐量子計算機暗号）において、耐量子計算機暗号に関する調査報告書とガイドライン(それぞれ2024年度版)並びに量子コンピュータによる共通鍵暗号の安全性への影響に関する調査報告書（2024年度版）を作成したとともに、「素因数分解の困難性に関する計算量評価」、「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図を更新した。暗号技術活用委員会では、同委員会の下に設置された暗号鍵管理ガイダンスWGにおいて、2022年度に発行した「暗号鍵管理ガイダンス」の追補版として「暗号鍵管理ガイダンスPart 2」を作成したとともに、クラウドにおける鍵管理ガイダンスの検討を行った。

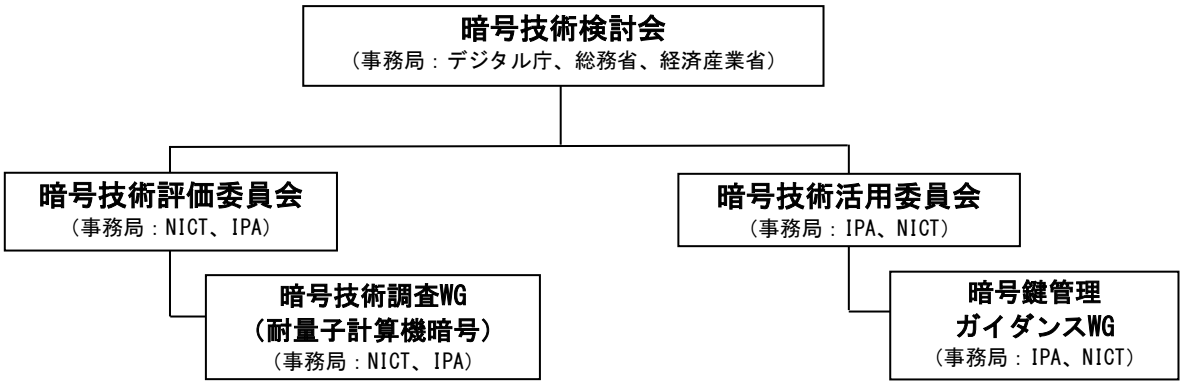


図2.2-1　CRYPTREC体制図（2024年度）

２．３．暗号技術検討会の開催実績

2024年度、暗号技術検討会は、下記内容について検討を行うため１回開催した。

【第 1 回】2025年3月25日（火）9:00～11:00

（主な議題）

・2024年度暗号技術評価委員会 活動報告について【報告】
・CRYPTREC暗号リスト仕様書の参照先変更について【報告】
・耐量子計算機暗号ガイドライン／調査報告書の更新について【承認】
・「量子コンピュータが共通鍵暗号の安全性に及ぼす影響」に関する外部評価報告書（案）について【承認】
・2024年度暗号技術活用委員会 活動報告について【報告】
・暗号鍵管理ガイダンス（Part 2）について【承認】
・耐量子計算機暗号（PQC）への対応について【承認】
・2025年度暗号技術評価委員会活動計画（案）について【承認】
・2025年度暗号技術活用委員会活動計画（案）について【承認】
・暗号技術検討会 2024年度 報告書（案）について【承認】

（概要）

・暗号技術評価委員会についてNICTより2024年度の活動報告が行われた。
・CRYPTREC暗号リスト仕様書の参照先変更について報告が行われた。
・耐量子計算機暗号ガイドライン／調査報告書の更新についてNICTより説明が行われ、原案のとおり承認された。
・「量子コンピュータが共通鍵暗号の安全性に及ぼす影響」に関する外部評価報告書（案）についてNICTより説明が行われ、原案のとおり承認された。
・暗号技術活用委員会についてIPAより2024年度の活動報告が行われた。
・暗号鍵管理ガイダンス（Part 2）についてIPAより説明が行われ、原案のとおり承認された。
・耐量子計算機暗号（PQC）への対応について事務局より説明が行われ、原案のとおり承認された。
・2025年度暗号技術評価委員会活動計画（案）についてNICTより説明が行われ、原案のとおり承認された。
・2025年度暗号技術活用委員会活動計画（案）についてIPAより説明が行われ、原案のとおり承認された。
・暗号技術検討会 2024年度 報告書（案）について事務局より説明が行われ、議論結果を反映することとした上で承認された。

３．各委員会の活動報告

３．１．暗号技術評価委員会

３．１．１．活動の概要

暗号技術評価委員会は、CRYPTREC暗号リストに掲載されている暗号技術や電子政府システム等で利用される暗号技術の安全性維持及び信頼性確保のために安全性及び実装に係る監視及び評価を行う。主要な検討課題は以下のとおりである。

- 暗号技術の安全性及び実装に係る監視及び評価
- 暗号技術の電子政府推奨暗号リストからの降格
- 暗号技術に関する注意喚起レポートのCRYPTRECホームページでの公表
- 推奨候補暗号リストへの新規暗号（事務局選出）の追加
- 新技術暗号等に係る調査

また、CRYPTREC暗号リストとは別の文書として、「暗号技術ガイドライン（耐量子計算機暗号）2024年度版」「耐量子計算機暗号の研究動向調査報告書2024年度版」を作成した。基本方針は以下のとおりである。

- 耐量子計算機暗号に関するガイドライン（2024年度版）、研究動向調査報告書（2024年度版）を作成するため、2023-2024年度に、耐量子計算機暗号に関するワーキンググループを設置した。
- 2023年度までに実施した調査と、2024年度の調査を含め、2024年3月にガイドライン2022年度版および研究動向調査報告書2022年度版を更新し、2024年度版とした。

さらに、調査報告書「量子コンピュータによる共通鍵暗号の安全性への影響に関する調査及び評価2024年度版」を作成した。基本方針は以下のとおりである。

- 量子コンピュータによる共通鍵暗号の安全性への影響に関する調査及び評価2024年度版については、2019年度に作成した「量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価」の更新のため、動向調査を行った。そして、実施した調査に基づき、2024年3月に調査報告書2019年度版を更新し、2024年度版とした。

これらの課題について2024年度に行った具体的な検討内容を、以下のとおり報告する。

３．１．２．暗号技術の安全性及び実装に係る監視及び評価

学会等での情報収集に基づくCRYPTREC暗号等の監視活動を行った。監視報告の詳細については、CRYPTREC Report 2024（暗号技術評価委員会報告）に掲載する。

３．１．３．暗号技術調査ワーキンググループ（耐量子計算機暗号）

大規模な量子コンピュータが実用化され、その量子コンピュータを用いた攻撃に対しても安全

性を担保することが期待される暗号（耐量子計算機暗号:PQC）の研究開発及び標準化などが各国で進められている。そこで、2020年度第2回暗号技術検討会において、耐量子計算機暗号ガイドラインを作成するために暗号技術調査ワーキンググループ(耐量子計算機暗号)（以下：PQC WG）を設置することが承認された。また、「素因数分解の困難性に関する計算量評価」、「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新を PQC WG で実施することが承認された。2024年度についても、2024年度第一回暗号技術評価委員会において、PQC WGが設置されること、および、2024年度のPQC WGの活動として以下の2点を実施する活動計画が承認された。

- 耐量子計算機暗号に関し、NISTのPQC標準化において第4ラウンドが進行中であることをはじめ技術開発、標準化活動が引き続き世界的に活発であることから、動向を2024年度末までに調査・把握し、調査報告書・ガイドラインの改定を行う。
- 「素因数分解の困難性に関する計算量評価」、「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図を更新する。

これらの成果（３．１．３．１～３．１．３．２節）は2024年度第二回暗号技術評価委員会にて報告され、了承された。

３．１．３．１．暗号技術調査ワーキンググループ（耐量子計算機暗号）

2022年度に耐量子計算機暗号に関する調査報告書とガイドライン（それぞれ2022年度版）を作成・公開したが、その後もNISTをはじめとする世界各国の機関において耐量子計算機暗号の選定・標準化活動が継続されており、情勢が流動的であることを鑑み、2023-2024年度の2年間で再度、耐量子計算機暗号に関する調査報告書とガイドライン（それぞれ2024年度版）を作成することが承認された。そして、2023-2024年度のPQC WGの活動により、2022年度版が出版された以降の研究技術動向に関して調査を行い、ガイドライン・調査報告書2024年度版を作成した。調査報告書・ガイドライン執筆方針の基本的な部分は2022年度版調査報告書・ガイドラインを踏襲している。

- 耐量子計算機暗号のスコープ

  公開鍵暗号を中心にまとめる。

- 耐量子計算機暗号に関する現状調査

  ガイドライン及び調査報告書に記載する耐量子計算機暗号を5分類とする。2022年度版とは異なり、耐量子計算機暗号の活用方法を調査報告書・ガイドラインの両方に記載。導入の章の内容について2022年度版では同じものであったが、ガイドラインでは技術的な詳細を省き簡略化する。これらの項目に関する情報を調査した。

耐量子計算機暗号調査報告書・ガイドライン

- 耐量子計算機暗号に関する調査報告書・ガイドラインの作成方針

- 2024年度版の内容は、2022年度版の調査報告書・ガイドラインをベースとし、技術の進展に伴う部分を追記・修正する。なお、著者の著作権の関係から調査報告書・ガイドラインともに改定ではなく新規の扱いとし、過去の版と区別する必要がある際には（2024年度版）のように年度を明示する。

- 耐量子計算機暗号ガイドラインは、暗号理論に精通していない利用者を対象とし、耐量子計算機暗号に関する調査報告書は、暗号理論の研究者や技術者を対象とする。基本的には耐量子計算機暗号ガイドラインは調査報告書から技術的詳細を省き、その一部を抜粋したものとする。2022年度版とは異なり、耐量子計算機暗号の活用方法を調査報告書・ガイドラインの両方に記載する。導入の章の内容について2022年度版では同じものであったが、ガイドラインでは技術的な詳細を省き簡略化する。

表3.1-1　ガイドラインの章立て

章	タイトル
1	はじめに
２	PQCの活用方法
３	格子に基づく暗号技術
４	符号に基づく暗号技術
５	多変数多項式に基づく暗号技術
６	同種写像に基づく暗号技術
７	ハッシュ関数に基づく署名技術
3章以降の構成（A章の場合：Aは３，４，５，６，７を表す）	
A.1.	安全性の根拠となる問題（例：LWE問題、シンドローム復号問題）
A.2.	代表的な暗号方式（例：Regev暗号、McEliece暗号）
A.3. A.3.1. A.3.2. A.3.3. ・・・	主要な暗号方式 暗号方式1（例：CRYSTALS-KYBER, Classic McEliece） 暗号方式２ 暗号方式３ ・・・
A.4.	まとめ

- 耐量子計算機暗号ガイドライン及び調査報告書に記載する暗号方式の選定基準
  公開鍵暗号方式である主要な耐量子計算機暗号（NIST PQC 標準化への提案方式等）を記載するが、対象となる暗号方式は PQC WG によって承認されたものである。

３．１．３．２．「素因数分解の困難性に関する計算量評価」、「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新

「素因数分解の困難性に関する計算量評価」、「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図（以下単に「予測図」という。）は公開鍵暗号方式のセキュリティパラメータの選択について検討を行うため、2006年度に設置された暗号技術調査WG（公開鍵暗号）において作成された。2019年度暗号技術評価委員会において、今後の予測図の取扱いについて審議し対応方針（「今後の予測図の取扱い」「今後の公開鍵暗号のパラメータ選択」）を決定した。2024年度において、対応方針は以下のとおりとなっている。

予測図の取扱い対応方針

〈今後の予測図の取扱い〉

(1) いわゆるムーアの法則を仮定して外挿線を年度末からプラス20年後まで従来通り直線で引き、評価に大きな変動がないと考えられる限りにおいては、安全サイドに倒した評価として予測図を当面の間更新していく。

〈今後の公開鍵暗号のパラメータ選択〉

(2) 公開鍵暗号のパラメータ選択に関する対応方針については、安全性以外にも相互接続性など、運用上の観点もあるため、暗号技術評価委員会だけではなく、暗号技術検討会、暗号技術活用委員会や関係各所などを含めて検討する。

※予測図は各国・国際標準化機関等により示されている主要な暗号技術の安全性基準と比較すると、より現状に則した評価であり、危殆化時期がそれらよりも先に延びるものとなっている。

予測図の更新について

素因数分解問題の困難性および楕円曲線上の離散対数問題の困難性に関する計算量評価に大幅な進展はなかったため、TOP500.orgにおける2024年6月と11月のベンチマーク結果を追加して予測図の更新を行った（図3.1-1及び図3.1-2）。

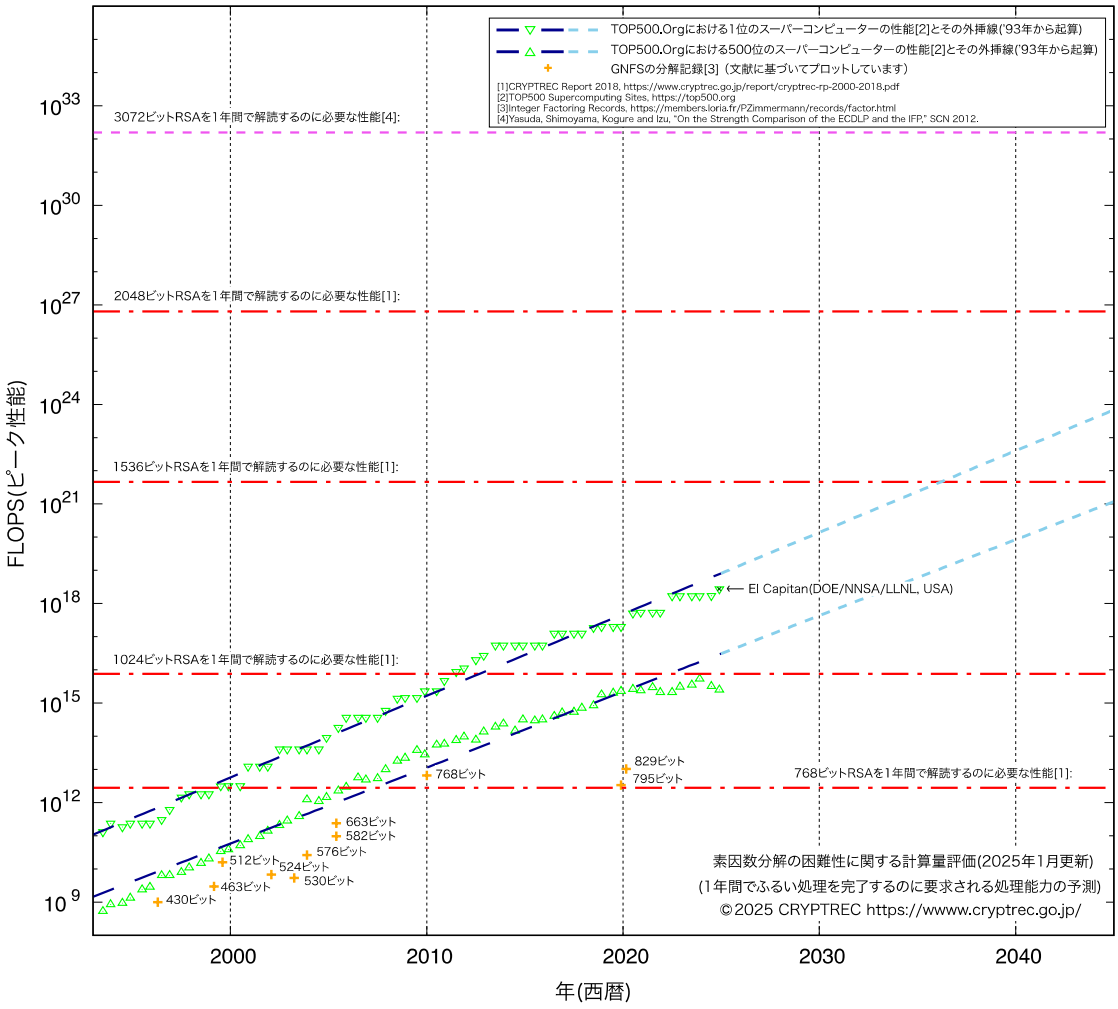


図3.1-1：素因数分解の困難性に関する計算量評価（2025年1月更新）[1]

[1] スーパーコンピューターの性能の伸びに関する外挿線は僅かであるが鈍化してきている。

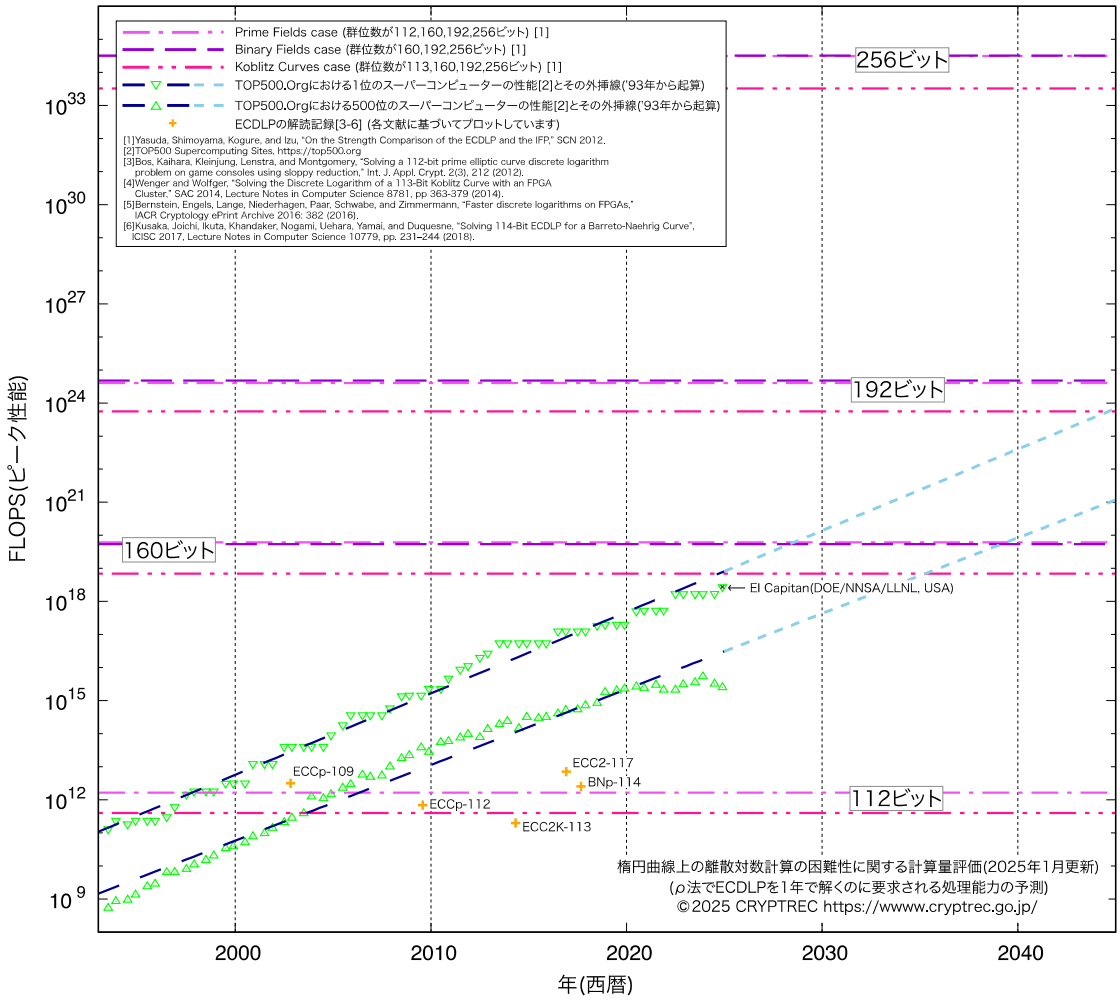


図3.1-2：楕円曲線上の離散対数計算の困難性に関する計算量評価（2025年1月更新）[2]

３．１．４．外部評価「量子コンピュータが共通鍵暗号の安全性に及ぼす影響」

３．１．４．１．背景

（１）2019年度、暗号技術調査ワーキンググループ（暗号解析評価）における活動として「量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価」を外部評価により実施し、本報告書（以下、「2019年度外部評価報告書（CRYPTREC EX-2901-2019)」という）をCRYPTRECの技術調査報告書として公開した。

（２）2022年度、PQC WGにおける活動として「CRYPTREC暗号技術ガイドライン（耐量子計算機暗号）（CRYPTREC GL-2004-2022）」と「耐量子計算機暗号の研究動向調査報告書（CRYPTREC TR-2001-2022）」（以下、「PQCガイドライン等」という）を作成した。

（３）2019年度外部評価報告書が公開されていることを踏まえ、PQCガイドライン等ではPQCとして共通鍵暗号を含まず、公開鍵暗号のみを示す言葉としている。つまり、PQCガイドライン等では

[2] スーパーコンピューターの性能の伸びに関する外挿線は僅かであるが鈍化してきている。

共通鍵暗号の耐量子安全性については触れていない。

（４）2023年度第2回暗号技術評価委員会で共通鍵暗号の耐量子安全性に関する議論が行われた。具体的には、共通鍵暗号の耐量子安全性に関する技術動向調査を実施し、2019年度外部評価報告書に調査内容を反映させる形で更新することはどうか、ということについて議論が行われた。本件について、事務局で対応を検討することが確認された。

（５）2024年度第1回暗号技術評価委員会において、量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価を外部評価で実施し、本結果を2019年度外部評価報告書に反映させる形で更新することが承認された。

３．１．４．２．評価・調査実施概要

細山田 光倫 様（日本電信電話株式会社）に外部評価を依頼した。選出理由と依頼内容は以下のとおりである。

（１） 選出理由

共通鍵暗号の耐量子安全性に関する広い知見をお持ちであり、当該分野に関する数多くの実績をお持ちであるとともに、2019年度外部評価報告書の執筆者であるため。

（２）依頼内容

量子コンピュータが共通鍵暗号の安全性に及ぼす影響について、公開されている解析手法やその影響範囲などについてまとめ、考察などを行い、2019年度外部評価報告書に最新の情報を反映させて、報告書（以下、「2024年度外部評価報告書」という）を作成する。

３．１．４．３．外部評価報告書の概要

（１）目次

2024年度外部評価報告書の目次は表3.1-2のとおり。表内において、2019年度版外部評価報告書から大きく加筆・修正した箇所を青字、追加された箇所を赤字で示す。

表3.1-2　2024年度外部評価報告書の目次

章	章タイトル	概要
1	はじめに	導入、2019年度版との差異
2	準備	1. Grover のアルゴリズム 2. Simon のアルゴリズム
3	攻撃のモデル： 古典クエリと量子クエリ	1. 古典攻撃モデル 2. Q1 モデル（古典クエリ攻撃モデル） 3. Q2 モデル（量子クエリ攻撃モデル） 4. Q1 モデルと Q2 モデルの比較 5. ハッシュ関数への攻撃のモデル
4	攻撃コスト評価方法に関する議論	1. 古典的衝突探索と誕生日のパラドクス 2. 最初の量子衝突探索アルゴリズム：BHT 3. BHT のアルゴリズムの効率性をめぐる議論 4. 量子ビット数の観点で効率的なアルゴリズム：CNS 5. その他の議論
5	汎用量子攻撃	1. Grover のアルゴリズムによる鍵回復・原像攻撃 2. 衝突探索および関連する問題

		3. 多重原像探索 4. タイムメモリトレードオフとレインボーテーブル 5. ノストラダムス攻撃 6. 汎用量子攻撃の具体的なコスト
6	量子クエリ攻撃（Q2）	1. Even-Mansour（EM）暗号への鍵回復攻撃 2. Feistel 暗号（Luby-Rackoff 構成）への識別攻撃 3. Crypto 2016 における Kaplan らの結果 4. Grover のアルゴリズムと Simon のアルゴリズムの組み合わせ 5. 隠れシフト問題と Kuperberg のアルゴリズム 6. 線形化攻撃 7. その他の古典攻撃の高速化
7	古典クエリ攻撃（Q1）	1. 桑門・森井による EM 暗号への鍵回復攻撃 2. オンライン-オフライン中間一致攻撃 3. 量子クエリ無しでの Simon のアルゴリズムの応用 4. 古典的に$2k$ビット安全ならkビット耐量子安全か？ 5. その他の古典攻撃の高速化 6. 古典安全性証明の結果が Q1 モデルへ持ち上がる場合
8	ハッシュ関数への（汎用でない）攻撃	1. 衝突攻撃 2. 原像攻撃
9	考察とまとめ	CRYPTREC暗号リスト、NIST LWC最終選考方式Asconの耐量子安全性に関する考察

（２）調査結果の概要

調査結果について、主に新規追加事項（表3.1-2の赤字箇所）を概説する。

① 準備：攻撃モデル（3章）

攻撃者は量子計算機を持っており、秘密鍵が埋め込まれた攻撃対象のオラクル（暗号化／復号／認証タグ生成オラクル）へクエリ可能である。

・古典クエリ攻撃（Q1）モデル：オラクルへのクエリが古典情報

・量子クエリ攻撃（Q2）モデル：オラクルへのクエリが量子重ね合わせ状態

なお、ハッシュ関数への攻撃を考える場合はオラクルを使用する必要がないため、本資料では単に量子攻撃モデルと記載する。

② 汎用量子攻撃：タイムメモリトレードオフとレインボーテーブル（5.4節）

ランダムな関数$H:\{0,1\}^n \to \{0,1\}^n$の原像探索にかかるオンライン計算量$T$は、使用可能なメモリサイズ$S$によって変動することが知られている。最も有名な手法には、Hellmanのタイムメモリトレードオフ攻撃とOechslinのレインボーテーブルがある。いずれも、時間とメモリのトレードオフ$T = O((2^n/S)^2)$が与えられる。

2024年にDunkelmanらは、量子計算機を用いることで時間とメモリのトレードオフを$T = O((2^n/S)^{1.5})$まで改善できることを示した。攻撃アイデアの根幹はHellmanのタイムメモリトレードオフ攻撃とOechslinのレインボーテーブルと同じである。

量子計算リソースは、多項式サイズの小さい計算用量子プロセッサと指数的に大きなサイズのQRAMがある、と仮定している。

③ 汎用量子攻撃：ノストラダムス攻撃（5.5節）

Merkle-Damgard構造のハッシュ関数Hに対する汎用量子攻撃である。具体的には、攻撃者は以下の問題を解く。

Step 1. 攻撃者は何らかの値yを事前に計算する。

Step 2. Xが選ばれ、攻撃者に与えられる。

Step 3. 攻撃者は$H(X\|R)=y$を満たすRを求める。

古典攻撃モデルでは、nビットハッシュ関数に対し、$O(2^{2n/3})$回の圧縮関数評価で攻撃が実行可能であると知られている。2022年にBenediktらは、量子攻撃モデルにおいて評価回数を$O(2^{3n/7})$回まで削減できることを示した。

④ Q2モデルにおける量子クエリ攻撃：線形化攻撃（6.6節）

Q2モデルにおける量子クエリ攻撃のアイデアは、周期関数を作ってSimonのアルゴリズムを適用する、ということが大部分を占めている。例えば、EM暗号、Feistel暗号、GCMやCBC-MACを含むブロック暗号利用モード、などへの攻撃がある（報告書6.1-6.3節）。ブロック暗号利用モードへの攻撃は2016年のKaplanらによって報告されたが、ISO標準のLightMACを含むいくつかのブロック暗号利用モードには同様のアイデアを適用できないという問題があった。

2021年にBonnetainらは、量子線形化攻撃を提案してこの問題を解決した。攻撃手法の詳細は省略するが、既存のアイデアと同様、攻撃対象の内部構造を詳細に分析して周期関数を作り、Simonのアルゴリズムを適用することにより、多項式時間での識別攻撃を可能にした。

⑤ 古典攻撃モデルにて$2k$ビット安全であればkビット耐量子安全か？（7.4節）

Groverのアルゴリズムにより、kビット鍵の全数探索に必要な計算量が2^kから$2^{k/2}$まで落ちることが知られている。量子計算機の実用化後に共通鍵暗号の安全性を現在と同程度に保つためには鍵長を2倍以上にする必要がある、と言われるのはこのためである。

一方、この逆の「古典攻撃モデルにて$2k$ビット安全であればQ1/Q2モデルにてkビット安全である」という主張について考察すると、必ずしもこの主張が成り立つとは限らない。例えば、Q2モデルではEM暗号に対する多項式時間攻撃（6.1節）があり、Q1モデルでもFX構成の拡張版である2XOR構成と呼ばれる構造のブロック暗号に対し、上記の主張を破る攻撃が2022年にBonnetainらによって報告された。

⑥ 古典的な安全性証明の結果がQ1モデルでもそのまま成り立つ場合（7.6節）

古典的な安全性証明がランダムオラクルモデルなどのプリミティブを理想化した条件で与えられるのではなく、反証可能な標準的仮定（CTRモードであればブロック暗号が擬似ランダム置換という仮定）のみに依存している場合、古典的な安全性証明の結果がQ1モデルでの安全性証明としてそのまま成り立つ。

⑦ ハッシュ関数に対する（汎用的でない）量子攻撃（8章）

衝突攻撃の攻撃可能段数に関して、古典攻撃モデルよりも量子攻撃モデルの方が優れている例がいくつか報告されている。例えば、電子政府推奨暗号リスト掲載のSHA-256、SHA-512、そしてSHA3-256が該当し、結果の詳細は表3.1-3の通りである。

表3.1-3　古典・量子攻撃モデルでのSHA2とSHA3に対する衝突攻撃の比較

対象	出力長	段数	攻撃可能段数	
			古典	量子
SHA-256	256	64	31	38
SHA-512	512	80	31	39
SHA3-224	224	24	5	6
SHA3-256	256	24	5	6

一方、原像攻撃の攻撃可能段数に関して、現状において古典攻撃モデルよりも量子攻撃モデルの方が優れている例は報告されていない。

（３）考察結果の概要

Q2モデルでの攻撃、Q1モデルでの攻撃、ハッシュ関数への攻撃に分け、種々の主要な方式の安全性への影響を考察する。より具体的には、CRYPTREC電子政府推奨暗号リスト掲載方式、そしてNIST標準軽量暗号Asconに焦点を当てる。

①　Q2モデルでの攻撃

古典攻撃モデルにおいて安全性が保証されている共通鍵暗号技術（GCM、CBC-MAC、など）に対して多項式時間で実行可能な攻撃が存在するが、Q2モデルでは攻撃対象が量子回路上に実装されている必要がある。これは非常に特殊な状況であり、現状では既存の共通鍵暗号技術にQ2モデルでの攻撃の影響が及ぶことは無いと考えられる。特に、CRYPTREC電子政府推奨暗号リスト掲載方式やNIST標準軽量暗号Asconの安全性を評価する上でQ2モデルでの攻撃を考慮する必要はない。

②　Q1モデルでの攻撃

Q2モデルでの攻撃とは異なり、古典攻撃モデルにおいて安全性が保証されている共通鍵暗号技術に対して多項式時間で実行可能な攻撃は存在しない。ただし、古典攻撃モデルにて$2k$ビット以上の安全性があったとしても、Q1モデルでの安全性がkビット以下になる例も示されているため、暗号技術ごとに確認が必要である。

Q1モデルにおいて、ハッシュ関数を除くCRYPTREC電子政府推奨暗号リスト掲載方式やNIST標準軽量認証暗号のAscon-AEAD/Ascon-80pqの安全性に量子計算機が与える影響は、“Groverのアルゴリズムを用いるとkビット鍵の全数探索が$O(2^{k/2})$で実行できるため、長期的に保護したいデータには鍵長が192ビットや256ビットの暗号技術を使用した方が賢明である”と考えられる。これらの暗号技術について、Q1モデルで安全性が期待できる範囲を表3.1-4でまとめる。

③　ハッシュ関数への攻撃

SHA-256、SHA-512、SHA3-256では、量子計算機が使用可能になると衝突攻撃の攻撃可能段数が古典攻撃モデルと比べて伸びることが知られている。表3.1-3で示すように、安全性マージンは十分に確保されているものの、今後の動向を注視する必要がある。その他、CRYPTREC電子政府推奨暗号リスト掲載のハッシュ関数やNIST標準軽量ハッシュ関数のAscon-Hash256/Ascon-XOF128の安全性に量子計算機が及ぼす影響は、汎用量子攻撃（特に、BHTのア

ルゴリズム）のみを考慮すれば十分である。これらの暗号技術について、BHTのアルゴリズムを適用するのに必要な計算時間と量子メモリの概算値を表3.1-5でまとめる。

表3.1-4　ハッシュ関数を除く電子政府推奨暗号リスト掲載方式、Ascon-AEAD、そしてAscon-80pqについて、Q1モデルで安全性が期待できる範囲（Groverのアルゴリズム：$\leq 2^{k/2}$）

技術分類	方式名	鍵長k（ビット）	安全性が期待できる範囲
ブロック暗号	AES Camellia	128 192 256	時間 $\leq 2^{64}$ 時間 $\leq 2^{96}$ 時間 $\leq 2^{128}$
ストリーム暗号	KCipher-2	128	時間 $\leq 2^{64}$
秘匿モード	CBC, CFB, CTR, OFB, XTS	k	時間 $\leq 2^{k/2}$ かつ 古典的に安全性が保証される範囲
認証付き 秘匿モード	CCM, GCM	k	時間 $\leq 2^{k/2}$ かつ 古典的に安全性が保証される範囲
メッセージ 認証コード	CMAC, HMAC	k	時間 $\leq 2^{k/2}$ かつ 古典的に安全性が保証される範囲
認証暗号	ChaCha20- Poly1305	256	時間 $\leq 2^{128}$ かつ 古典的に安全性が保証される範囲
	Ascon- AEAD128	128	時間 $\leq 2^{64}$ かつ 古典的に安全性が保証される範囲
	Ascon-80pq	160	時間 $\leq 2^{80}$ かつ 古典的に安全性が保証される範囲

表3.1-5　電子政府推奨暗号リスト掲載のハッシュ関数、Ascon-Hash256、Ascon-XOF128に対してBHTのアルゴリズムを適用するのに必要な計算時間と量子メモリの概算値： $\min(2^{c/3}, 2^{h/3})$

方式名	キャパシティc	出力長h	計算時間	量子メモリ
SHA-256 SHA-512/256 SHA3-256	- - 512	256	$2^{85.3}$	$2^{85.3}$
SHA-384 SHA3-384	- 768	384	2^{128}	2^{128}
SHA-512	-	512	$2^{170.7}$	$2^{170.7}$

SHA3-512	1024			
SHAKE128	256	$\ell \geq 256$	$\min(2^{85.3}, 2^{\ell/3})$	$\min(2^{85.3}, 2^{\ell/3})$
SHAKE256	512	$\ell \geq 256$	$\min(2^{170.7}, 2^{\ell/3})$	$\min(2^{170.7}, 2^{\ell/3})$
Ascon-Hash256	256	256	$2^{85.3}$	$2^{85.3}$
Ascon-XOF128	256	$\ell > 0$	$\min(2^{85.3}, 2^{\ell/3})$	$\min(2^{85.3}, 2^{\ell/3})$

（４）2019年度外部評価報告書における結論との差異

具体的な方式の実用面での安全性評価について、2019年度外部評価執筆時からの大きな差異は、ハッシュ関数の衝突攻撃可能段数が古典攻撃モデルの場合に比べて量子攻撃モデルの場合に伸びることが明らかになってきたということである。

このような状況の変化に応じ、2019年度版外部評価報告書の結論を表3.1-6で示すように変更した。その他の結論部分について大きな差異はない。

表3.1-6　ハッシュ関数に関する技術動向の変化

2019年度外部評価報告書執筆時	現在
古典的に128ビット安全性のあるハッシュ関数の安全性に量子攻撃が現実的な脅威を直接及ぼすとは現状考えづらい。	重要な用途に供するハッシュ関数の出力長（スポンジ構造の場合は出力長に加えてキャパシティ長）はBHTのアルゴリズムの計算量$O(2^{n/3})$を基準にして384ビットや512　ビットのものを用いた方が無難であると考えられる。

３．１．４．４．外部評価報告書に対する暗号技術評価委員会の見解

2024年度外部評価報告書から、CRYPTREC電子政府推奨暗号リスト掲載方式とNIST標準軽量暗号Asconの安全性に量子計算機が及ぼす影響は、汎用量子アルゴリズム（特に、Groverのアルゴリズムと BHTのアルゴリズム）のみを考慮すれば十分であるという結論を得た。

以上より、2024年度外部評価報告書（案）を2024年度外部評価報告書とすることが了承された。

３．１．５．暗号技術評価委員会の開催実績

2024年度、暗号技術評価委員会は計2回開催した。各回会合の概要は表3.1-7のとおりである。

表3.1-7　暗号技術評価委員会の開催状況

回	開催日	議案
第1回	2024年7月9日	■　暗号技術評価委員会活動計画の具体的な進め方についての審議 ■　PQC WGの活動計画案の審議 ■　「量子コンピュータが共通鍵暗号の安全性に及ぼす影響」につい

		て外部評価を行うことの審議 ■ 監視状況報告
第2回	2025月3月3日	■ PQC WGの活動内容の報告 ■ 「量子コンピュータが共通鍵暗号の安全性に及ぼす影響」について外部評価についての報告と、本外部評価の公開に関する審議 ■ 監視状況報告 ■ CRYPTREC Report 2024作成について ■ CRYPTRECシンポジウム開催について

また、PQC WGは計2回開催した。さらに、メールによる審議を実施した。2023年度から2024年度のPQC WG各回、および、メール審議の概要は表3.1-8のとおりである。

表3.1-8 PQC WGの開催状況

年度	回	耐量子計算機暗号ガイドラインの議論・決定・報告
2023年度	第1回 2023/9/13	✓ 追記・改定の方針について議論 ✓ 執筆担当者を議論
	第2回 2024/1/19	✓ 追記・改定すべき項目及びその章立ての決定 ✓ 調査の中間報告
2024年度	第1回 2024/7/26	✓ 中間報告、追加及び削除すべき暗号方式があれば議論
	第2回 2025/2/3	✓ 内容の確定
	メール審議 2025/2/4～2/12	✓ エディトリアルな部分の審議 ・ 「PQC」という単語の示す範囲 ・ 句読点の利用方針

３．２．暗号技術活用委員会

３．２．１．活動の概要

2024年度の活動概要は以下の通りである。詳細については、CRYPTREC Report 2024暗号技術活用委員会報告[3]を参照されたい。

(1) 暗号鍵管理ガイダンスの拡充

暗号鍵管理ガイドラインの拡充を目的として進めていた暗号鍵管理ガイダンスについて、2021年度から2023年度に引き続いて暗号鍵管理ガイダンスWGを設置し、2023年5月に発行したガイダンスでは記載を見送った部分の拡充を行う。2023年5月版の内容見直しも含め、2024年度完成を目標とする。

(2) 暗号利活用のための新たなガイダンスの作成

「クラウドにおける鍵管理ガイダンス」をテーマとする新たなガイダンスの作成に着手する。おおむね2年程度での完成を想定して執筆作業を行う。クラウド利用者が留意すべき鍵管理を解説することを目的とする。

３．２．２．暗号鍵管理ガイダンスの拡充

2024年度は、2023年5月発行のガイダンスにおいて記載を見送った部分について追補版のガイダンスを作成した。分冊構成としたため、2023年5月発行のガイダンスを「暗号鍵管理ガイダンスPart 1」、今年度執筆した追補版を「暗号鍵管理ガイダンスPart 2」と呼ぶこととした。

表3.2-1　暗号鍵管理ガイダンスの章構成

暗号鍵管理システム設計指針（基本編）	暗号鍵管理ガイダンスPart 1（2023年5月発行）	暗号鍵管理ガイダンスPart 2（2024年度執筆）
1. はじめに	1. はじめに	1. はじめに
2. 暗号鍵管理の在り方	（1章に集約）	（1章に集約）
3. 本設計指針の活用方法	（1章に集約）	（1章に集約）
4. 暗号鍵管理システムの設計原理と運用ポリシー		2. 暗号鍵管理システムの設計原理と運用ポリシー
5. 暗号アルゴリズム運用のための暗号鍵管理オペレーション対策	2. 暗号アルゴリズム運用のための暗号鍵管理オペレーション対策	
6. 暗号アルゴリズムの選択	3. 暗号アルゴリズムの選択	

[3] CRYPTREC Report 2024 暗号技術活用委員会報告, https://www.cryptrec.go.jp/promo_cmte.html

7. 暗号アルゴリズム運用に必要な鍵情報の管理	4. 暗号アルゴリズム運用に必要な鍵情報の管理	
8. 暗号鍵管理デバイスへのセキュリティ対策		3. 暗号鍵管理デバイスへのセキュリティ対策
9. 暗号鍵管理システムのオペレーション対策		4. 暗号鍵管理システムのオペレーション対策

ガイダンスPart 1（2023年5月発行）では、「設計指針」の5章から7章に該当する項目に関して、項目の概説及びその記載例を提供している。これらの項目は、暗号鍵管理システム(CKMS)の利用環境に関わらず検討する必要がある共通項目であり、「狭義」の意味での暗号鍵管理に相当する。

ガイダンスPart 2（2024年度執筆）では、「設計指針」の4章、8章、9章に該当する項目に関して、項目の概説及びその記載例を提供している。Part 2の2章はCKMSの全体方針を定める項目である。また、Part 2の3章はCKMSに利用するデバイス管理を含む場合に検討すべき項目であり、Part 2の4章はCKMSのシステム管理を含む場合に検討すべき項目である。これらのPart 2における3章や4章までを含む場合、「広義」の意味での暗号鍵管理に相当する内容となる。

ガイダンスPart 2の各章の記載概要は以下のとおりである。

1. はじめに

   イントロダクションとして、ガイダンスPart 2の位置づけについて、「設計指針」やガイダンスPart 1との関係を含めて記載した。また、ガイダンスPart 2におけるトイモデルとして設定した「IoT機器（家電製品）向けに公開鍵証明書を発行するプライベートCAシステム」の概要を説明した。

2. 暗号鍵管理システムの設計原理と運用ポリシー

   「設計指針」での「暗号鍵管理システムの設計原理と運用ポリシー」における検討項目について解説・考慮点を記載した。CKMSのセキュリティポリシー、CKMSに関わるエンティティの定義、CKMSを構成するデバイスやコンポーネントの一覧、CKMSでの実現目標、CKMSに関わる法規制や標準化技術、将来的な移行対策などの検討項目から構成される。トイモデルとしてプライベートCAのセキュリティポリシーや運用の想定例を設定して、各検討項目の対応例を説明した。

3. 暗号鍵管理デバイスへのセキュリティ対策

   「設計指針」での「暗号鍵管理デバイスへのセキュリティ対策」における検討項目について解説・考慮点を記載した。CKMSにおけるセキュアな暗号鍵管理・保管の中核となる暗号鍵管理デバイスへのアクセスコントロールに対する検討事項、暗号鍵管理デバイス内の暗号モジュールに対する検討項目、暗号鍵管理デバイス及びCKMSのセキュリティ評価・試験に関する検討項目、暗号鍵管理デバイスにおける障害発生時のBCP対策に関わる検討項目などで構成される。トイモデルとして、プライベートCAで用いるハードウェア・セキュリティモジュールに関わる具体例として各検討項目の対応例を説明した。

4. 暗号鍵管理システムのオペレーション対策

   「設計指針」での「暗号鍵管理システムのオペレーション対策」における検討項目について解

説・考慮点を記載した。CKMSにおける物理的セキュリティコントロール及びコンピュータシステムやネットワークにおけるセキュリティコントロールとそれらが危殆化した場合のBCP対策、システム及びデバイスの開発プロセスやセキュリティメンテナンスに関わる検討項目、セキュリティアセスメントに関わる検討項目、CKMS全体に関わる災害時のBCP対策などの検討項目で構成される。トイモデルとして、プライベートCAの設置環境や入退室管理、プライベートCAを構成するサーバシステムでのセキュリティコントロール、セキュリティアセスメントにおける実施項目、プライベートCAの災害復旧対策を想定して各検討項目の対応例を説明した。

３．２．３．暗号利活用のための新たなガイダンスの作成

新たなガイダンスとして「クラウドにおける鍵管理ガイダンス（仮称）」を設定し、作成方針を議論した。委員会での議論を経た検討結果を以下の観点で説明する。

① 目的・想定読者・スコープについて
② 記載内容のポイントについて
③ 作成スケジュールおよび検討体制について

① 目的・想定読者・スコープについて

本ガイダンス執筆の目的については以下のように整理した。

クラウドサービスを活用して効率的に情報システムを構築することは一般的になっている。一方で、クラウドサービスの活用には、クラウドサービスに預けた情報が漏洩すること、設定不備やクラウドサービスにおける障害波及のリスクがあること、等の懸念事項も生じる。クラウドサービスにおける暗号鍵管理システムを適切に選択・構築・運用することによって、そうした懸念事項に対処できる部分がある。クラウドサービスにおける暗号鍵管理の仕組みや注意事項を解説したガイダンスを作成し、クラウド環境で安全に暗号を運用するための一つのガイダンスとする。

本ガイダンスの想定読者については以下とした。

クラウドサービスを利用した情報システムの構築者（SI 事業者）、運用者、利用者。

本ガイダンスのスコープは以下のように整理した。

IaaS や PaaS のクラウドサービスを利用して、情報システムのプラットフォーム構築を行うケースを対象に、どのような鍵管理サービスを提供すべきかをターゲットとする。暗号機能による保護の対象はクラウドサービスに預けたデータ及び鍵情報の機密性と完全性の確保、並びに暗号化消去とする。

② 記載内容のポイントについて

本ガイダンス記載内容のポイントとなる事項について、以下の点を設定した。ただし、詳細はガイダンス作成の過程で再度議論する。

- クラウド鍵管理サービスの分類
  クラウドサービスプロバイダ(CSP)が提供する鍵管理サービスを体系化し、ユースケースに応じてどのような鍵管理サービスを利用すべきかを判断できる情報を提供する。

- クラウド鍵管理サービスに関わる責任分界について
  クラウドサービス利用時の鍵管理システムに関わる CSP との責任分界について、クラウド鍵管理サービスの種類に応じて原則となる考え方を整理する。その際、クラウドサービスモデルに依存する部分があるかについても現状を整理する。

- 暗号鍵管理ガイダンス（NIST SP 800-130）の Framework Requirement との関係
  暗号鍵管理ガイダンスにおいて解説している検討項目(Framework Requirement)について、クラウドサービス利用時はどのように検討されるべきかを記載する。現在の暗号鍵管理ガイダンスでは、オンプレミスに構築した CKMS をトイモデルに設定して検討項目への対応例を記載しているため、クラウドサービスを利用した場合に重点的に検討すべき項目や対応例がどのように変わるかを説明する。

③ 作成スケジュールおよび検討体制

本ガイダンスの作成にあたって、2025 年度より WG を新しく設置することとなった。WG 委員として CSP 事業者、SI 事業者・SaaS 事業者、クラウド HSM ベンダ、クラウドサービス利用者、大学や関連団体などの有識者にそれぞれ参画いただき、事務局を中心に委員の知見をまとめる形で作成を進める。

作成スケジュールについては、WG の任期である 2 年間で遅くとも本ガイダンスの作成を行う計画である。ただし、技術やサービスの進展が早い領域でもあるため、計画は柔軟に捉えることとする。

３．２．４．暗号技術活用委員会の開催状況

2024年度の暗号技術活用委員会での審議概要は表3.2-2の通りである。

表3.2-2　暗号技術活用委員会の開催状況

回	開催日	議案
メール	2024年6月	● 2024 年度暗号鍵管理ガイダンス WG 活動計画の審議
第一回	2024年10月28日	● 2024 年度暗号技術活用委員会活動計画の確認 ● 2024 年度暗号鍵管理ガイダンス WG 活動計画の確認 ● 暗号鍵管理ガイダンス WG 進捗報告 ● クラウドにおける鍵管理ガイダンスについて
第二回	2025年3月4日	● 2024 年度暗号鍵管理ガイダンス WG 活動報告及びガイダンス案の審議 ● クラウドにおける鍵管理ガイダンスについて ● 2024 年度暗号技術活用委員会活動報告案について

４．今後のCRYPTRECの活動について

CRYPTRECでは、2025年度も、電子政府推奨暗号等の安全性を評価・監視するとともに、暗号技術の更なる普及促進を行うべく検討を進める。

暗号技術検討会においては、CRYPTREC暗号リストの更新等について必要に応じて検討を行う予定である。

暗号技術評価委員会においては、NISTをはじめとする世界各国の機関において耐量子計算機暗号の選定・標準化活動が継続されており、情勢が流動的であることに鑑み、引き続き暗号技術調査ワーキンググループ（耐量子計算機暗号）を設置して、耐量子計算機暗号に関する最新動向を把握するとともに、社会的動向を踏まえてNIST標準として公開されたFIPS-203,204,205について安全性評価・実装性能評価関連の活動を開始する予定である。

暗号技術活用委員会においては、新たなガイダンスとして作成方針を検討した「クラウドにおける鍵管理ガイダンス（仮称)」について、新たなWGを設置して同ガイダンスの作成を本格的に開始する予定である。また、暗号技術検討会に応じて、耐量子計算機暗号をめぐる社会的動向を踏まえ、耐量子計算機暗号の取扱い基準や運用ガイドライン／ガイダンスにおける耐量子計算機暗号の位置づけ・記載内容等についての検討を開始する予定である。

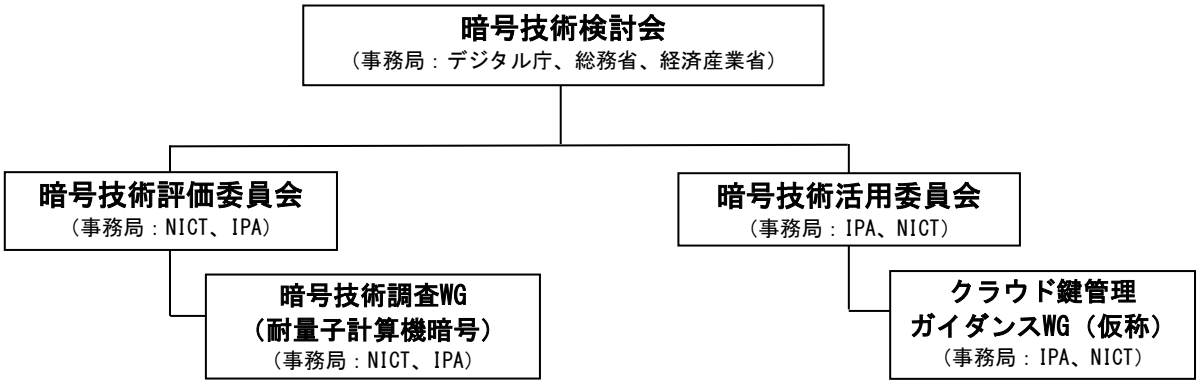


図4-1 CRYPTREC体制図（2025年度）（予定）