資料１

2022年度 第２回 暗号技術検討会

令 和 ５ 年 ３ 月 ３ ０ 日
９ ： ０ ０ ～
オ ン ラ イ ン 開 催

議 事 次 第

１．開会

２．議事
(1) 2022年度暗号技術評価委員会 活動報告について【報告】
(2) CRYPTREC暗号技術ガイドライン（耐量子計算機暗号）及び CRYPTREC暗号技術ガイドライン（高機能暗号）について【承認】
(3) 2022年度暗号技術活用委員会 活動報告について【報告】
(4) 暗号鍵管理ガイダンスについて【承認】
(5) 「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)」（案）に対する意見募集に寄せられたご意見に対するデジタル庁、総務省及び経済産業省の考え方（暗号技術検討会事務局案）並びにCRYPTREC暗号リストの改定版（暗号技術検討会事務局案）について【承認】
(6) 2023年度暗号技術評価委員会活動計画（案）について【承認】
(7) 2023年度暗号技術活用委員会活動計画（案）について【承認】
(8) 暗号技術検討会 2022年度 報告書（案）について【承認】
(9) 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	2022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	2022年度暗号技術評価委員会　電子メールによる審議とその結果
資料３－４	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）活動報告
資料３－５	CRYPTREC暗号技術ガイドライン（耐量子計算機暗号）
資料３－６	2022年度暗号技術調査ワーキンググループ（高機能暗号）活動報告
資料３－７	CRYPTREC暗号技術ガイドライン（高機能暗号）
資料３－８	軽量暗号に関する技術動向調査報告（外部評価）
資料３－９	CRYPTREC暗号技術ガイドライン（耐量子計算機暗号）及びCRYPTREC暗号技術ガイドライン（高機能暗号）について
資料４－１	2022年度 暗号技術活用委員会 活動報告
資料４－２	暗号鍵管理ガイダンスワーキンググループ活動報告
資料４－３	暗号鍵管理ガイダンス概要について
資料４－４	暗号鍵管理ガイダンス
資料５－１	CRYPTREC暗号リストの改定について
資料５－２	「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)」（案）に対する意見募集に寄せられたご意見並びにそれらに対するデジタル庁、総務省及び経済産業省の考え方（暗号技術検討会事務局案）

資料５－３　CRYPTREC暗号リスト（改定版）（暗号技術検討会事務局案）
資料５－４　CRYPTREC暗号リスト（現行版）
資料６　2023年度暗号技術評価委員会活動計画（案）
資料７　2023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 2022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

2023.3.30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　上席特別研究員
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　教授
太田　和夫	国立大学法人電気通信大学　名誉教授
高木　　剛	国立大学法人東京大学大学院　情報理工学系研究科　教授
田村　裕子	日本銀行　金融研究所　企画役
近澤　　武	三菱電機株式会社　情報技術総合研究所　開発戦略部　担当部長
手塚　　悟	慶應義塾大学　環境情報学部　教授
本間　尚文	国立大学法人東北大学　電気通信研究所　教授
松井　　充	三菱電機株式会社　開発本部　役員技監
松浦　幹太	国立大学法人東京大学 生産技術研究所　教授
松本　　勉	国立大学法人横浜国立大学大学院　環境情報研究院　教授
松本　　泰	セコム株式会社　IS研究所　顧問
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 セキュリティ保証スキーム研究チーム 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（政府機関総合対策担当）
個人情報保護委員会事務局 参事官
警察庁 情報通信局 情報管理課 情報セキュリティ対策官
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 産業技術環境局 国際電気標準課長
防衛省 整備計画局 情報通信課 AI・サイバーセキュリティ推進室長
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 高機能暗号研究チーム長
独立行政法人情報処理推進機構　　技術本部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

2022 年度暗号技術評価委員会活動報告

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

下記の通り、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 暗号等の監視

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。

- 今年度時点では、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 2022 で報告する。
  （資料３－２参照）
- ECDSA 及び ECDH については、現状通り、当該暗号技術を電子政府推奨暗号リストに記載しておくことは妥当であると判断した。SC2000 については、応募社の判断を尊重し、取り下げを認め、当該暗号技術を推奨候補暗号リストから削除することは妥当であると判断した。
  （資料３－３参照）

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

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。

- 降格や削除が必要となる暗号技術は無かった。

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。

- 注意喚起レポートの発行を要する事案は発生しなかった。

④ 推奨候補暗号リストへの新規暗号（事務局選出）の追加

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。

- 2021 年度から継続して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、耐量子計算機暗号に関するガイドラインを作成し、「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。
  - 暗号技術調査ワーキンググループ（耐量子計算機暗号）を開催し、2021 年度に決定した執筆方針及び目次案に沿ってガイドラインの案及び調査報告書の案を作成した。特に、耐量子計算機暗号の中で今後の利用が見込まれる暗号方式について議論し、それらを選択してガイドラインに記載した。また、近年に脆弱性が発見された世界標準の有力候補の暗号方式とその脆弱性については調査報告書の案にまとめた。
  - 同ワーキンググループを開催し、2020 年度に決定した「今後の予測図の取り扱い」の説明を読みやすく修正し、これに基づいて「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。

    （資料３－４）、（資料３－５）
- 高機能暗号に関するガイドラインを作成するため、2021 年度に引き続き、高機能暗号に関するワーキンググループを設置する。
  - 暗号技術調査ワーキンググループ（高機能暗号）を開催し 2021 年度に決定した目次案に沿って、高機能暗号ガイドラインの案を作成した。執筆に際し、専門性の高い新規委員に依頼し、技術、応用、標準化等の内容を精査した。また、高機能暗号の応用事例に関するヒアリングを日本電気株式会社、三菱電機株式会社に対して行い、その内容をガイドラインの

案に反映した。
（資料３－６）、（資料３－７）

- 2016 年度に NICT 及び IPA が策定・公表した「CRYPTREC 暗号技術ガイドライン(軽量暗号)」の更新のため、2022 年度は、NIST Lightweight コンペティションファイナリストを主な対象とした安全性および実装性能などに関わる調査・評価を行う。

  - NIST Lightweight Cryptography Project のファイナリスト 10 方式を対象とした安全性評価及び実装性能評価を外部評価により実施した。なお、安全性評価に関しては、上記 10 方式に加え、ISO/IEC 標準規格として 29192 シリーズで規格化された軽量メッセージ認証コードの 1 つである Tsudik's keymode も対象とした。また、軽量暗号に関わる NIST 公開文書や ISO/IEC などの標準化動向調査も外部評価により実施した。
    （資料３－８参照）
    なお、来年度は、2021 年度第 2 回暗号技術評価委員会において承認された更新方針に従い、事務局によりガイドラインの更新案を編集し、ドラフト版について外部有識者にガイドラインとして掲載内容の適切性や情報の過不足などについてレビュー頂き、完成版を 2023 年度第 2 回暗号技術評価委員会にて審議する予定である。

(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、下の表のとおり 2 回開催した。

回	開催日	議案
メール審議	2022 年 6 月 1−7 日	● 暗号技術調査ワーキンググループ（高機能暗号）の活動計画案の審議
第 1 回	2022 年 7 月 26 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 ● 暗号技術調査ワーキンググループ（耐

		量子計算機暗号）の活動計画案の審議 ● 暗号技術調査ワーキンググループ（高機能暗号）の活動計画案の報告 ● 外部評価（軽量暗号に関するガイドライン係る技術動向調査）実施についての審議 ● 監視状況報告
メール審議	2023 年 1 月 30 日－2 月 10 日	● 自主取下げに係る電子メールによる審議
第 2 回	2023 月 2 月 27 日	● 自主取下げに係る電子メールによる審議内容と結果の報告 ● 暗号技術調査ワーキンググループ（耐量子計算機暗号）の活動内容の報告 ● 暗号技術調査ワーキンググループ（高機能暗号）の活動内容の報告 ● 軽量暗号ガイドラインに係る技術動向調査結果の報告 ● 監視状況報告 ● CRYPTREC Report 2022 作成について ● CRYPTREC シンポジウム開催について

以上

暗号技術評価委員会委員名簿

（五十音順、敬称略）

委員長	高木　剛	東京大学　教授
委員	青木 和麻呂	文教大学　准教授
委員	岩田　哲	名古屋大学　准教授
委員	上原　哲太郎	立命館大学 教授
委員	大東　俊博	東海大学　准教授
委員	國廣　昇	筑波大学　教授
委員	四方　順司	横浜国立大学　教授
委員	手塚　悟	慶應義塾大学　教授
委員	花岡　悟一郎	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研究センター　首席研究員
委員	藤﨑　英一郎	北陸先端科学技術大学院大学　教授
委員	本間　尚文	東北大学　教授
委員	松本　勉	横浜国立大学　教授
委員	松本　泰	セコム株式会社　ディビジョンマネージャー
委員	山村　明弘	秋田大学　教授

資料３－２

監視状況報告

1. 監視活動報告

2021 年度第二回暗号技術評価委員会（2022 年 2 月 26 日）から 2022 年度第二回暗号技術評価委員会（2023 年 2 月 27 日）までに、表 1 に示す国際会議に参加するとともに各種調査を行い、暗号解読技術等に関する研究動向を収集した。

表 1 国際会議への参加状況

学会名・会議名		開催国・都市	期間
Eurocrypt 2022	The 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques	(Virtual Conference)	2022年5月30日～6月3日
Crypto 2022	The 42nd Annual International Cryptology Conference	(Virtual Conference)	2022年8月13日～8月18日
PQCrypto 2022	Post-Quantum Cryptography The 13th International Workshop	(Virtual Conference)	2022年9月28日～9月30日
TCC2022	Theory of Cryptography The 20th International Conference	(in-person)	2022 年 11 月 7 日～11 月 10 日
NIST 4th Standardization	Fourth PQC Standardization Conference	(Virtual Conference)	2022 年 11 月 29 日～12 月 1 日
Asiacrypt 2022	The 28th International Conference on the Theory and Application of Cryptology and Information Security	(Virtual Conference)	2022 年 12 月 5 日～12 月 9 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。

AES については、AsiaCrypt 2022 にて、ブーメラン攻撃の計算量の更新が発表された。また同学会で、AES の量子回路実装の最適化がなされており、今後の量子計算機による AES 解析に応用が期待されている。

またストリーム暗号の ChaCha についての攻撃論文も 3 件報告された（CRYPTO2022、Eurocrypt 2022、Asiacrypt 2022)。特に Eurocrypt 2022 にて、6 ラウンドの ChaCha128 に対する攻撃について大きな更新が見られた。

RSA については、CRT-RSA に対して、部分的に鍵が公開された場合の素因数分解である、部分的鍵公開攻撃（Partial Key Exposure attack）が、Eurocrypt 2022 と AsiaCrypt 2022 で 2 件発表された。Eurocrypt 2022 では純粋な CRT-RSA に対して、AsiaCrypt 2022 では指数ブラインディングされた場合に対して、それぞれ新たな攻撃が報告された。

また、SHAKE128 の 6 ラウンドに対する衝突攻撃が、AsiaCrypt 2022 で発表された。同論文では、量子的な設定の下での衝突攻撃が、SHA3-224 と SHA3-256 の 6 ラウンドに対しても行われている。

いずれも、現実的な脅威となるには至っていないが、今後動向を注視すべきである。

2.1.1. 共通鍵暗号に関する解読技術

・Revisiting Related-Key Boomerang attacks on AES using computer-aided tool [Asiacrypt 2022]

Patrick Derbez, Marie Euler, Pierre-Alain Fouque, Phuong Hoa Nguyen

近年、ブロック暗号のブーメラン識別器やブーメラン攻撃を自動的に探索するために、いくつかの MILP モデルが導入されている。しかし、それらは鍵スケジュールが線形である場合にのみ使用可能である。ここでは、AES のような非線形の鍵スケジュールに対して、新しいモデルを導入する。このモデルはより複雑であり、網羅的な探索には時間がかかりすぎる。しかしながら、ソルバにいくつかのヒントを追加することで、時間計算量 2^{124}、データ計算量 2^{124}、メモリ計算量 $2^{79.8}$ で AES-192 に対する現在最高の関連鍵ブーメラン攻撃を発見した。これは、ASIACRYPT 2009 で Biryukov と Hovratovich による攻撃の計算量（それぞれ 2^{176}、2^{123}、2^{152}）より優れている。特に時間計算量とメモリ計算量において大きな改善を与え、暗号解読における MILP の威力が示された。

・Synthesizing Quantum Circuits of AES with Lower T-depth and Less Qubits [Asiacrypt 2022]

Zhenyu Huang, Siwei Sun

量子アルゴリズムによる暗号解読に必要な資源を正確に見積もるためには、量子アルゴリズムを基本的な量子ゲートで構成される量子回路に帰着する必要がある。本研究では、Grover と Simon のアルゴリズムに基づく量子攻撃でよく用いられる反復型共通鍵暗号の量子オラクルを実装する回路について、いくつかの汎用的な合成技術と最適化技術が提案された。まず、ブロック暗号のラウンド関数を in-place に実装するための一般的な構造を提

案する。次に、線形および非線形な暗号構成要素の効率的な量子回路を合成するための新しい技術が導入される。これらの技術を AES に適用し、深さ-幅のトレードオフ（depth-width tradeoff）の戦略が系統的に調べられる。その過程で、AES の S-box の古典的回路に関する新しい知見に基づいて、証明可能な最小の T-depth を持つ量子回路が導出されている。その結果、AES の量子回路の実装に必要な T-depth と幅（量子ビット数）が大幅に削減された。著者らの回路と EUROCRYPT 2020 で提案された回路と比較すると、幅を増やさずに T-depth を 60 から 40 に、あるいは幅をわずかに増やして 30 に減らすことに成功している。これらの回路は、Microsoft Q#で実装されており、ソースコードも公開されている。ASIACRYPT 2020 で提案された回路と比較すると、著者ら回路の 1 つは幅が 512 から 371 に減少し、同時に Toffoli-depth が 2016 から 1558 に減少していることが確認されている。また実際、深さを増やす代わりに、幅を 270 に減らすことができる。さらに、深さと幅のトレードオフの全範囲が提供され、AES の量子回路の合成と最適化における新記録も樹立されている。

・Revamped Differential-Linear Cryptanalysis on Reduced Round ChaCha [Eurocrypt 2022]

Sabyasachi Dey, Hirendra Kumar Garai, Santanu Sarkar, Nitin Kumar Sharma

本論文では、ChaCha に対する既存の差分線形攻撃に対するいくつかの改良点を提供する。ChaCha は 20 ラウンドを持つストリーム暗号である。CRYPTO 2020 において、Beierle らは正しいペアを選択した場合、3.5 ラウンド目に差分が発生することを観測している。彼らはこれを用いて改良した攻撃を行ったが、正しいペアを実現するためには平均で 2^5 回の繰り返しが必要であることを示した。

この方向で、リスト化（listing）の助けを借りて、正しいペアを見つける技術を提供する。また、PNB 構築の戦略的改善、計算量見積の修正、2 つの入出力ペアを用いた代替攻撃法も提供する。これらを用いて、時間計算量を改善し、Beierle らが 256 ビット版の ChaCha で報告した $2^{230.86}$ から $2^{221.95}$ に減少させることに成功した。また、10 年ぶりに 6 ラウンドの 128 ビット版 ChaCha に対する既存の計算量(Shi et al: ICISC 2012)を 1100 万倍以上改善し、6.5 ラウンド ChaCha128 に対する史上初の攻撃で時間計算量 $2^{123.04}$ を実現した。

・Rotational Differential-Linear Distinguishers of ARX Ciphers with Arbitrary Output Linear Masks [CRYPTO 2022]

Zhongfeng Niu, Siwei Sun, Yunwen Liu, Chao Li

EUROCRYPT 2021 で提案された回転差分線形攻撃は、差分線形攻撃の差分部分を回転差分に置き換えて一般化したものである。EUROCRYPT 2021 では、Liu らが Morawiecki らの手法（FSE 2013）に基づき、出力線形マスクが単位ベクトルである特殊なケースについて、回転微分線形相関を評価する方法を発表した。この手法により、Friet、Xoodoo、

Alzette に対して、出力線形マスクが単位ベクトルである強力な（回転）微分線形識別器がいくつか発見された。しかし、任意の出力マスクに対する回転微分線形相関をどのように計算するかは未解決であった。

本研究では、この未解決問題の一部を解決している。任意の出力線形マスクに対するモジュロ加算の（回転）微分線形相関を計算する効率的なアルゴリズムを提示し、それを基に ARX 暗号の(回転)微分線形相関を評価する手法が導出された。本技術を Alzette、SipHash、Chacha、Speck に適用した結果、決定論的なものを含め、大幅に改善された（回転）微分線形識別器が確認された。本研究の成果は全て実用的であり、実験的に検証され、手法の有効性が確認された。さらに、FSE2008、FSE2016、CRYPTO2020 で採用された ChaCha に対する実験的な識別器を説明することを試みている。予測された相関は実験的な相関に近いものであった。

・Latin Dances Reloaded: Improved Cryptanalysis against Salsa and ChaCha, and the proposal of Forró [Asiacrypt 2022]

Murilo Coutinho, Iago Passos, Juan Grados, Fábio de Mendonça, Rafael Timóteo, Fábio Borges

本論文では、ARX 暗号、特にストリーム暗号の Salsa/ChaCha ファミリに対する 4 つの主要な貢献を紹介する。

(a) ChaCha に対する差分-線形識別器を改善した。このために、アルゴリズムをより単純なサブラウンドの観点から見ることによって、線形近似の導出にアプローチする新しい方法が提案されている。このアイデアを用いて、3 つの単純なルールから、過去の研究から得られる全ての線形近似を導き出すことが可能であることを示した。さらに、もう 1 つのルールを追加することで、Eurocrypt 2021 で Coutinho と Souza が提案した線形近似を改善できることを示す。

(b) Salsa に対する攻撃を改善するため、双方向線形展開（Bidirectional Linear Expansions、BLE）と呼ばれる技術を提案する。これまでの研究では、ラウンドに前進する線形拡張のみを検討していたが、BLE では 1 ビットを前進と後退の両方向に拡張することが検討されている。

(c) これらの暗号の暗号解析の研究から得られた全ての知識を用いて、ラウンドごとの拡散と暗号解析への耐性を向上させるためのいくつかの修正を提案し、新しいストリーム暗号 Forró を完成させた。これにより、安全性を維持したままラウンド数を減らすことができ、多くのプラットフォーム、特に制約のあるデバイスにおいて、より高速な暗号を実現することができるとしている。

(d) さらに著者らは、GPU を用いた高いパフォーマンス環境で使用可能な、CryptDances という、Salsa、ChaCha、Forró の暗号解析ツールを開発している。CryptDances では、差分相関の計算、ChaCha の新しい線形近似の自動導出、PNB 攻撃の

計算量の見積の自動化などが可能になっている。CryptDances は https://github.com/MurCoutinho/cryptDances で利用可能である。

2.1.2. 公開鍵暗号に関する解読技術

・Approximate Divisor Multiples - Factoring with Only a Third of the Secret CRT-Exponents [Eurocrypt 2022]

Alexander May, Julian Nowakowski, Santanu Sarkar

本論文は、CRT-RSA の秘密指数d_p, d_qで、公開指数eが小さい場合の部分鍵公開攻撃について研究している。eが定数である場合、d_p, d_qのうち 1 つのビットの半分を知ることで、Coppersmith の有名な「factoring with a hint」の結果によって RSA 剰余Nを因子分解できることが知られている。この設定をeが定数でない場合に拡張する。

少し意外な結論として、eのサイズが $N^{1/12}$ である RSA が部分鍵公開攻撃に対して最も弱いということが、本論文の攻撃によって示された。これは、最上位ビット（MSB）または最下位ビット（LSB）のいずれかを含むd_p, d_qの両ビットの 3 分の 1 があれば、多項式時間でNを因数分解できるためである。

$ed_p = 1 + k(p-1)$, $ed_q = 1 + \ell(q-1)$とせよ。技術的には、著者らはNの素因数分解を二つの新しいアプローチで求めている。第一のステップでは、kとℓを多項式時間で復元する。これは、MSB の場合は完全に初等的に、LSB の場合は Coppersmith の格子に基づく方法を用いて実現される。これにより、求められたkを用いて、kpを法とする一変数多項式の根を計算することで、Nの素因数分解を得ることができる。これは、Howgrave-Graham の approximate divisor アルゴリズムの、Nの未知の約数pの既知の倍数kに対する approximate divisor multiples の場合への拡張と見なすことができる。approximate divisor multiples のポイントは、多項式時間で復元可能な未知数が、倍数kの大きさに対して線形に増加することである。

この部分鍵公開攻撃は、MSB がわかっている場合は厳密である一方、LSB の場合は標準的な Coppersmith タイプのヒューリスティックに依存する。このヒューリスティックを実験的に検証することにより、実際には小さな格子寸法で既に漸近的な境界値に到達することを示すことで、この部分鍵公開攻撃の実用性も示されている。

・A Third is All You Need: Extended Partial Key Exposure Attack on CRT-RSA with Additive Exponent Blinding [Asiacrypt 2022]

Yuanyuan Zhou, Joop van de Pol, Yu Yu, François-Xavier Standaert

Eurocrypt 2022 において、May らは CRT-RSA に対する部分鍵公開（PKE）攻撃を提案し、公開指数eのサイズが$N^{1/12}$の場合の、秘密指数d_pおよびd_qの最上位ビットら（MSBs）の 1/3、または最下位ビットら（LSBs）の 1/3 どちらかだけを知って、Nを効

率よく素因数分解した。実際には、PKE 攻撃はこれらの指数のサイドチャネル漏洩に依存している。CRT-RSA のサイドチャネル耐性実装では、未知のランダムなブラインド係数r_p, r_qにおける加法的なブラインド指数$d'_p = d_p + r_p(p-1)$, $d'_q = d_q + r_q(q-1)$を用いているため、PKE 攻撃はより困難なものになることが多い。

以上のことの背景に、本論文は、May らの PKE 攻撃を加法的なブラインド指数の CRT-RSA に拡張している。この場合、ブラインドされた CRT 指数 d_p^{π} と d_q^{π} の既知の MSB またはLSB のみを使用して秘密鍵全体を復元することができる。設定上では$r_p e \in (0, N^{1/4})$を許容するが、著者らが拡張した PKE 攻撃は、$r_p e \approx N^{1/12}$ の時理想的に働く。このケースでは、ブラインドされた CRT 指数 d'_p と d'_q の MSB もしくは LSB の 1/3 から全ての秘密鍵が復元可能である。著者らの拡張した PKE 攻撃は、May らによる以下の 2 ステップのアプローチに準拠している：第一ステップで鍵に依存する定数 k'（$ed'_p = 1 + k'(p-1)$, $ed'_q = 1 + l'(q-1)$ ）を計算し、第二ステップで $k'p$ を法とする一変数多項式の根を計算することでNを素因数分解する。本論文では、この手法を以下のように拡張している。MSB の場合では、第一ステップにおいて、$k'l'$の値を一つ推定した後k'を因数分解により計算する方法と、$k'l'_1, \ldots, k'l'_z$という複数の推定値からk'を GCD により確率的に計算するという、2 つのオプションを提案している。LSB の場合では、第二ステップで現れる差分一変数多項式を構成するアプローチが拡張されている。形式的な分析によって、本手法における LSB 攻撃は通常の Coppersmith タイプの仮定の下、多項式時間で機能することがわかっている。一方、本手法における MSB 攻撃は、簡約された入力サイズに対し劣指数時間を要する（問題が$e^2 r_p r_q \approx N^{1/6}$のサイズの素因数分解に帰着される）か、新しいヒューリスティックな仮定の下で確率的多項式時間で機能する。最も一般的な鍵サイズ（1024 ビット、2048 ビット、3072 ビット）とブラインド因数のサイズ（32 ビット、64 ビット、128 ビット）の設定の下で、著者らの実験を行い、この Coppersmith タイプの仮定と、彼らの新たなヒューリスティックな仮定の両方が尤もらしい事を確認している。

以上の攻撃は、128 ビットのブラインド指数が存在する場合の CRT-RSA に対して、初めて実験的な有効性が確認された PKE 攻撃であると著者らは主張している。さらに、Montgomery Ladder 指数 CRT 実装を対象としたリアルなサイドチャネル部分鍵漏洩に対して、提案された攻撃の応用実験も行っている。

2.1.3. ハッシュ関数に関する解読技術

・Exploring SAT for Cryptanalysis: (Quantum) Collision Attacks against 6-Round SHA-3 [Asiacrypt 2022]

Jian Guo, Guozhen Liu, Ling Song, Yi Tu

本研究では、古典的および量子的な設定における SHA-3 ハッシュ族のインスタンスに対する衝突攻撃に焦点を当てる。JoC 2020 で Guo らが提案した SHA3-256 および他の変種に対する 5 ラウンドの衝突攻撃以来、他に本質的な進展は発表されていない。徹底的な調査により、このような SHA-3 への衝突攻撃をより多くのラウンドに拡張することの課題は、差分トレイル探索の非効率性にあることが突き止められた。

この障害を克服するために、本論文の著者らは、SAT ベースの自動探索ツールキットを開発した。このツールは衝突攻撃の複数の中間ステップで使用され、その過程で遭遇する差分トレイル探索やその他の最適化問題において高い効率性を示しているとされる。その結果、6 ラウンド SHAKE128 に対する古典衝突攻撃、量子衝突攻撃、SHA3-224 と SHA3-256 に対する 6 ラウンド量子衝突攻撃が発表された。時間計算量はそれぞれ、$2^{123.5}$、$2^{67.25}/\sqrt{S}$、$2^{97.75}/\sqrt{S}$、$2^{104.25}/\sqrt{S}$である（Sは量子コンピュータのハードウェアリソースを表す)。6 ラウンドの SHA3-224 と SHA3-256 に古典衝突攻撃が適用されないことは、量子衝突攻撃のカバー率が高いことを示していて、CRYPTO 2021 で Hosoyamada と Sasaki が観測した SHA-2 に対するものと一致すると、著者らは指摘している。

2.2. その他の注視すべき技術動向

CRYPTREC 暗号リストの対象ではないが、特に NIST の耐量子計算機暗号の標準化に関して、いくつかの報告がなされた。

2023 年 2 月現在、耐量子計算機暗号として、CRYSTALS-KYBER、CRYSTALS-DILITHIUM、FALCON、SPHINCS+が標準化方式として既に選出されている。他候補を選別中の Round 4 において、同種写像を用いた KEM である SIKE が、セキュアではないということが、SIKE チームから発表された。これは、Castryck と Decru によりクリティカルな攻撃論文が、プレプリントとして発表されたためである。この攻撃によって、同種写像ベースの暗号形式である SIDH、SIKE、B-SIDH、SIOT はセキュアではないことが判明している。その一方、SIDH 署名、CSIDH、SeaSign、CSI-FiSH、OSIDH、SQISign はまだ攻撃が発表されてはいない。これについては、今後の情勢を注視する必要がある。

また、Rainbow に対して、NIST が設定している安全性レベル 1 のパラメータ設定に対して、2.5 GHz の PC で約 53 時間で攻撃が可能であるという報告が、CRYPTO2022、PQCrypto 2022 両方で発表された。

その他、共通鍵暗号などについても複数の攻撃が報告されている。より詳しくは、使用される機会が多いか今後多くなると予想される、共通鍵暗号、公開鍵暗号、ハッシュ関数、電子署名、それらに関連する暗号プリミティブ技術について、次の事項が発表された。

2.2.1. 共通鍵暗号に関する技術動向

・Key Guessing Strategies for Linear Key-Schedule Algorithms in Rectangle Attacks [Eurocrypt 2022]

Xiaoyang Dong, Lingyue Qin, Siwei Sun, Xiaoyun Wang

著者らは、線形鍵スケジュール暗号に対する矩形攻撃（rectangle attack）のための四分位を生成する際、鍵候補を示唆しうる正しい四分位は、いくつかの非線形な関係を満たす必要があることを見出した。しかし、生成される四分位の中には、常にこれらの関係に違反し、鍵候補を示唆しないものがある。著者らは、過去の矩形攻撃フレームワークからヒントを得て、四分位を生成する前に特定のキーセルを推測することで、無効な四分位法の数を減少させることができることを見出した。しかし、一度に多くのキーセルを推測すると、早期に中止する手法の利点が失われ、全体の計算量が増す可能性がある。そこで本論文では、より良いトレードオフを得るために、線形鍵スケジュール暗号に対する新しい矩形攻撃フレームワークを構築し、全体的な計算量を軽減するか、より多くのラウンドを攻撃することを目的としている。

このトレードオフモデルでは、全体の計算量に影響を与える多くのパラメータが存在しており、特に四分位生成前の鍵推測セルの数と位置の選択について重要となる。最適なパラメータを特定するため、SKINNY を例として、鍵回復フェーズに最適な矩形識別器、四分位生成前の鍵推測セルの数と位置、網羅的な探索ステップに影響を与える構築する鍵カウンタのサイズなど、均一な自動ツールを構築する。この自動化ツールを用いて、SKINNY-128-384 に対する、関連鍵設定における 32 ラウンドの鍵回復攻撃を確認し、これまでの攻撃を 2 ラウンド拡張した。また、n-2n や n-3n の他のバージョンについても、従来よりも 1 ラウンド多く達成することができた。 さらに、これまでの矩形識別器を用いて、ラウンド数を削減した ForkSkinny、Deoxys-BC-384、GIFT-64 に対してより良い攻撃を行っている。最後に、著者らの矩形暗号の枠組みを関連鍵設定から単一鍵設定に変換し、10 ラウンドの Serpent に対する新しい単一鍵矩形暗号攻撃を与えることを議論する。

・A Correlation Attack on Full SNOW-V and SNOW-Vi [Eurocrypt 2022]

Zhen Shi, Chenhui Jin, Jiyan Zhang, Ting Cui, Lin Ding, Yu Jin

本論文では、線形フィードバックシフトレジスタ（Linear Feedback Shift Register、以下 LFSR）のバイナリストリームと SNOW-V および SNOW-Vi のキーストリーム間の相関を合成関数に近似する手法に基づいて探索する方法を示す。有限状態機械（Finite State Machine、以下 FSM）に入力される LFSR の 4 つのタップが連続する 3 クロックの間の線形関係を利用して、SAT/SMT 手法に基づく自動探索モデルを示し、高い相関を持つ一連の線形近似トレイルを探し出す。中間マスクを使い切ることで、相関が $-2^{-47.76}$ の 2 値線形近似を見つけることができる。この近似を用い、SNOW-V に対する相関攻撃を、予想時間計算量 $2^{246.53}$、メモリ計算量 $2^{238.77}$, 同じ鍵および初期ベクトル（IV）で生成された $2^{237.5}$ の鍵ストリームで提案

する。SNOW-Vi については、同じ相関を持つバイナリ線形近似を提供し、SNOW-V と同じ複雑さで相関攻撃を実装する。本論文の著者らの知る限りでは、これは完全な SNOW-V および SNOW-Vi に対する、網羅的な鍵探索を上回る効率的な攻撃として初めて知られるものである。その結果、鍵と IV の単一ペアの鍵流の最大長が 2^{64} 未満であるという設計制約を無視すれば、SNOW-V と SNOW-Vi のいずれも 256 ビットのセキュリティレベルを保証できないことが示された。

・Refined Cryptanalysis of the GPRS Ciphers GEA-1 and GEA-2 [Eurocrypt 2022]

Itai Dinur, Dor Amzaleg

EUROCRYPT 2021 で、Beierle らは GPRS 暗号 GEA-1 と GEA-2 の最初の公開解析を発表した。彼らは、GEA-1 は 64 ビットのセッション鍵を使用しているが、65 ビットのキーストリームを知るだけで、44 GiB のメモリを使用して時間計算量 2^{40} で復元可能であることを示した。この攻撃は、暗号の初期化処理にある弱点を利用したもので、設計者が意図的に隠して安全性を低下させたと推測される。

GEA-2 ではそのような弱点は見つかっていなかったが、Beierle らはこの暗号に対して、約 2^{45} の時間複雑性を持つ攻撃を発表している。主な実用上の障害は、GPRS フレームを完全に暗号化するために使用される 12800 ビットのキーストリームを知る必要があることである。この攻撃のバリエーションは、連続するキーストリームのビットが少ない場合や、(連続する長いブロックがない) 利用可能なキーストリームが断片化されている場合にも、よりコストはかかるものの、適用可能である。

本論文は、GEA-1 および GEA-2 について、これまでの解析を改善し、補完するものである。GEA-1 では、メモリ使用量を 44 GiB から $2^{13} = 8192$ 倍、時間使用量を 2^{40} 倍に削減した攻撃を考案した。実装では、最新のラップトップで GEA-1 のセッションキーを平均 2.5〜時間で回復させることができる。GEA-2 に関しては、Beierle らの解析を補完する 2 つの攻撃が説明される。最初の攻撃は、攻撃者が利用できる連続したキーストリームビット数 (ℓで示される) と時間計算量の間の線形トレードオフを得ることができる。これは、おおよそ$\ell \leq 7000$ の範囲において、従来の攻撃より改善される。具体的には、 $\ell = 1100$ の場合、本攻撃の計算量は約 2^{54} である (従来の攻撃はブルートフォースの計算量 2^{64} より高速ではない)。利用可能なキーストリームが断片化されている場合、第二の攻撃は、時間計算量のコスト増加なしに、前攻撃のメモリ計算量を 32 GiB から 64 MiB に 512 倍減少させることができる。

本攻撃は、ストリーム暗号の解析技術と他の文脈で用いられるアルゴリズム技術 (例えば k-XOR 問題の解決など) の新しい組み合わせに基づくものである。

・A Greater GIFT: Strengthening GIFT against Statistical Cryptanalysis [Eurocrypt 2022]

Ling Sun, Bart Preneel, Wei Wang, Meiqin Wang

GIFT-64 は PRESENT より軽量な 128 ビット鍵による 64 ビットブロック暗号である。本論文では、差分攻撃と線形攻撃に対する GIFT-64 の詳細な分析を行う。著者らの研究は、最適な差分および線形特性の自動探索手法を、注意深い手動分析で補完している。差分攻撃の設定では、ラウンドごとに 2 つのアクティブな S-box を持つ差分特性の存在を理論的に説明し、これらの特性のいくつかの新しい性質を導いている。さらに、7 ラウンド以上をカバーする GIFT-64 の最適な差分特性はすべて、1 ラウンドにつき 2 つの S-box をアクティブにしなければならないことを証明する。すべての最適な特性は手作業で構成することができる。差分的な設定における作業と並行して、線形的な設定においても同様の分析を行う。しかし、差分設定での明確な見解とは異なり、GIFT-64 の最適な線形特性は、少なくとも 1 ラウンドが 1 つの S-box のみを活性化させるものでなければならない。さらに、自動探索手法の助けを借りて、線形攻撃に対して同等の安全性を維持しながら、差分攻撃に対してより優れた耐性を持つ 24 種類の GIFT-64 の亜種を見出した。これらの亜種は統計的暗号解析に対して GIFT-64 を強化されたため、ラウンド数を 28 から 26 に減らすことができると著者らは主張した。この観測により、GIFT-64 よりもエネルギー消費量の少ない暗号を作ることが可能になった。GIFT-64 の場合と同様に、ラウンド数を削減した亜種についても、ほとんどのアプリケーションに関係がないため、関連鍵の安全性を主張することはない。

・Beyond quadratic speedups in quantum attacks on symmetric schemes [Eurocrypt 2022]

Xavier Bonnetain, André Schrottenloher, Ferdinand Sibleyras

本論文では、古典的な問合せのみを用いてブロック暗号設計に対する初の量子鍵回復攻撃を行い、古典的な最良の攻撃と比較して 2 次以上の時間短縮を達成したことを報告する。

著者らは、EUROCRYPT 2012 での Gaži and Tessaro の 2XOR-Cascade 構成を研究している。これは、2n ビットの鍵を持つ n ビットブロック暗号から 5n/2 ビットの安全性を持つ n ビットブロック暗号を提供する鍵長拡張技術であり、理想モデルにおける安全性証明がなされている。ASIACRYPT 2019 での Bonnetain らの offline-Simon アルゴリズムが拡張され、特にこの構成を量子時間$\tilde{O}(2^n)$で攻撃できることを示し、最高の古典攻撃より 2.5 の量子速度向上を提供する。

一般に、共通鍵暗号のポスト量子安全性については、鍵のサイズを 2 倍にすれば十分であると考えられている。これは、Grover の量子探索アルゴリズムやその派生アルゴリズムが、最大でも 2 次関数的な高速化にしか到達できないためである。著者らの攻撃は、いくつかの対称構成の構造を利用することで、この限界を克服できることを示すものである。特に、2XOR-Cascade をブロック暗号強化として用いることは、量子敵対者に対してはそのブロック暗号自身と同じ安全性しか提供できないために、できない。

・Post-Quantum Security of the Even-Mansour Cipher [Eurocrypt 2022]

Gorjan Alagic, Chen Bai, Jonathan Katz, Christian Majenz

Even-Mansour 暗号は、公開されたランダム置換 $P;\{0,1\}^n \to \{0,1\}^n$ から、(鍵付きの) 疑似ランダム置換 E を構成する簡単な方法である。現在 NIST で標準化が検討されている軽量な暗号システムなど、さまざまな共通鍵暗号の中核をなしている。古典的な攻撃に対して安全であり、最適な攻撃には $q_P q_E \approx 2^n$ のオーダーで E への q_E 回の問合せと P への q_P 回問合せが必要となる。しかし、攻撃者に E と P の両方に対する量子アクセスが与えられた場合、この暗号は全く安全ではなく、 $q_P = q_E = O(n)$ 回のクエリを用いた攻撃が知られている。

しかし、もっともらしい設定では、量子攻撃者は、Pに対しては量子アクセス権を保持することができる一方で、信頼できるパーティにより実装された鍵付き並べ換えEについては、古典的なアクセス権しかもたない。この設定では、$q_P^2 q_E \approx 2^n$ の攻撃が知られており、純粋な古典的ケースと比較して安全性が低下することが示されているが、この自然な「ポスト量子」設定においても Even-Mansour 暗号の安全性がまだ証明できるかという疑問も残されている。

本論文では、この未解決の問題を解決し、このポスト量子設定における攻撃には$q_P^2 q_E + q_P q_E^2 \approx 2^n$ が必要であることを示す。著者らの結果は Even-Mansour の 2 鍵版と 1 鍵版の両方に適用される。また、量子問合せの下界に関する先行研究の結果を一般化したものでもあるため、その方面からも興味深いと思われる。

・Information-Combining Differential Fault Attacks on DEFAULT [Eurocrypt 2022]

Marcel Nageler, Christoph Dobraunig, Maria Eichlseder

差分故障解析 (DFA) は、共通鍵暗号の実装に対する非常に強力な攻撃である。 ほとんどの対策は実装レベルで適用される。ASIACRYPT 2021 で Baksi らは、線形構造を持つ S-box を使用することで、DFA に対して暗号レベルの耐性を内在させることを目的とした設計戦略を提案した。彼らは、彼らのインスタンスであるブロック暗号 DEFAULT において、DFA 敵対者は 128 鍵ビットのうち最大 64 ビットしか学習できないため、残りのブルートフォース計算量 2^{64} は現実的ではないと主張した。

本論文では、DFA 敵対者がラウンド間の情報を組み合わせて完全な鍵を復元できることを示し、その安全性の主張を無効とする。特にこのような暗号は、線形方程式を用いた正規化で効率的に表現できる等価な鍵の大きなクラスを示すことを観察する。 これを DEFAULT の強力な鍵スケジュールと組み合わせて利用することで、100 回以下の欠陥計算と無視できる時間計算量で鍵復元に成功している。さらに、独立したラウンド鍵を持つ DEFAULT の理想化バージョンでさえ、正規化された鍵に基づく情報結合攻撃に対して脆弱であることを示す。

・Superposition Meet-in-the-Middle Attacks: Updates on Fundamental Security of

AES-like Ciphers [CRYPTO 2022]

Zhenzhen Bao, Jian Guo, Danping Shi, Yi Tu

中間一致攻撃アプローチは最も強力な共通鍵暗号解読技術の一つとして良く知れらており、MD4, MD5, Tiger, HAVAL, Haraka-512 v2 ハッシュ関数全体に対する原像攻撃や、完全ブロック暗号 KTANTAN の鍵回復への応用で実証されている。攻撃が成功するには、プリミティブを 2 つの独立したチャンクに分離し、各アクティブセルが 1 つのチャンクのみを表現するために使用されるか、または混合されると使用不可能とみなされる状態にすることが重要である。本論文では、そのようなセルの一部が線形に混合され、独立したセルと同様に有用であることを観察する。これは、重ね合わせ状態の導入と付随する一連の技術につながり、EUROCRYPT 2021 で Bao らが、CRYPTO 2021 で Dong らが提案した MILP ベースの探索フレームワークに組み込まれ、AES ライクのハッシュ関数やブロック暗号に幅広く応用できることが分かる。

・Triangulating Rebound Attack on AES-like Hashing [CRYPTO 2022]

Xiaoyang Dong, Jian Guo, Shun Li, Phuong Pham

Rebound attack は FSE 2009 で Mendel らによって導入され、状態からの自由度を利用して、差分パスの重い中間ラウンドをフリーに実現することができる。ASIACRYPT 2009 で Lamberger ら、FSE 2010 で Gilbert と Peyrin が発明した Super-Sbox 技術により、インバウンドフェーズが 2 ラウンドに拡張された。ASIACRYPT 2010 では、Sasaki らが non-full-active Super-Sbox を導入し、メモリ使用量をさらに削減した。

本論文では、この研究成果を発展させ、複数の 1 ラウンドまたは (non-full-active) 2 ラウンドの Super-Sbox インバウンドフェーズを、状態と鍵の自由度を十分に活用し、かつ大容量メモリを使用せずに接続できる、Super-Inbound を紹介している。この技術を応用して、いくつかの AES ライクなハッシュ関数で古典的または量子衝突を発見し、AES-128 と Skinny ハッシュモード、Saturnin-hash、Grostl-512 などのターゲットで攻撃ラウンド数を 1〜5 向上させた。本攻撃の正しさを実証するため、古典的な設定において推定時間計算量 2^{24} の 6 ラウンド AES-128-MMO/MP に対する semi-free-start 衝突を実装し、標準的な PC で瞬時にペアのインスタンスを見つけることができた。

・Simplified MITM Modeling for Permutations: New (Quantum) Attacks [CRYPTO 2022]

André Schrottenloher, Marc Stevens

中間一致攻撃 (Meet-in-the-middle、MITM) とは、内部状態が 2 つの独立した経路 (「前方」と「後方」) に沿って計算され、それが一致する一般的なパラダイムである。近年では、EUROCRYPT 2021 で Bao らが開発した MILP モデルに代表されるように、汎用ソルバに詳細な攻撃モデルを用いて、改良された攻撃を自動的に探索することが行われてい

る。本論文では、汎用ソルバへの入力として大幅に削減された攻撃表現と、任意の解に対して詳細な攻撃の存在と計算量を証明する理論分析を組み合わせた、よりシンプルなMILP モデリングを研究している。このモデル化により、広範な暗号並べ換えに対する古典的攻撃と量子的攻撃の両方を発見することができる。第一に、Spongent ハッシュ関数の並べ替えを用いたPresent ライクな構成がある。識別器の中間一致攻撃ステップを最大3ラウンド改善された。第二に、AES のような設計がある。Bao らのモデルよりはるかに単純であるにもかかわらず、提案されたモデルは以前の最良の結果を回復することができる。唯一の制限は、鍵スケジュールから自由度を使用しないことである。第三に、このモデルはFeistel ネットワークのような、より多くの並べ替えをターゲットに拡張できることを示す。この文脈で、Simpira v2 や Sparkle の縮小版に対する新しい推測と決定の攻撃を行った。最後に、このモデルを用いて、古典的な攻撃と同じラウンド数で、新しい量子的な前像および擬似原像攻撃（例：Haraka v2, Simpira v2 ...）を見つけることができる。

・Simon's Algorithm and Symmetric Crypto: Generalizations and Automatized Applications [CRYPTO 2022]

Federico Canale, Gregor Leander, Lukas Stennes

本論文では、共通鍵暗号プリミティブを破るためにSimon のアルゴリズムを適用する方法についての理解が深められる。一方では、新しい攻撃の探索を自動化する。この方法を用いて、5ラウンドMISTY L-FK や5ラウンドFeistel-FK（内部並べ替えあり）などの構成に対する最初の効率的な鍵回復攻撃を Simon のアルゴリズムを用いて自動的に見つけることができた。また、Simon アルゴリズムの一般化として、非標準的なHadamard 行列を用いた研究を行い、周期以外の性質を持つ量子共通鍵暗号解読ツールの拡張を目指す。主な結果としては、いずれの一般化もそれを達成できないことであり、量子計算機で非標準的なHadamard 行列を利用して共通鍵暗号プリミティブを破るには、根本的に新しい攻撃が必要であると結論付けている。

・Quantum Attacks on Lai-Massey Structure [PQCrypto 2022]

Shuping Mao, Tingting Guo, Peng Wang, Lei Hu

Aaram Yun らは、Lai-Massey 構造がFeistel 構造と同等の安全性を有すると考えた。しかし、Luo らは、3ラウンドのLai-Massey 構造が、Feistel 構造とは異なるSimon アルゴリズムの量子攻撃に耐性があることを示した。本研究では、典型的な Lai-Massey 構造に対する量子攻撃を行う。その結果、3ラウンドLai-Massey 構造に対する量子CPA 識別器と、Feistel 構造と同じ4ラウンドLai-Massey 構造に対する量子CCA 識別器が存在することが示された。Lai-Massey 攻撃に対する攻撃を準Feistel 攻撃に拡張する。準Feistel 構造の結合（combiner）が線形である場合、3ラウンドbalanced 準Feistel

構造に対する量子 CPA 識別器と、4 ラウンド balanced 準 Feistel 構造に対する量子 CCA 識別器が存在することを示す。

・Algebraic Meet-in-the-Middle Attack on LowMC [Asiacrypt 2022]

Fukang Liu, Santanu Sarkar, Gaoli Wang, Willi Meier, Takanori Isobe

本論文では、部分的な非線形層の特徴を利用し、LowMC の安全性を解析するために代数的な中間一致攻撃（MITM）という新しい手法を提案している。これにより、単純な差分列挙攻撃のメモリ計算量を最先端のものより削減することが可能である。

LowMC の差分トレイルから完全な鍵を取り出す効率的な代数的手法が CRYPTO 2021 で提案されているが、その時間計算量は鍵サイズに対して指数関数的であることに変わりはなかった。本研究では、トレイルに十分な数のアクティブな S-box が存在する場合、これを定数時間にまで短縮する方法を示している。上記の新技法により、CRYPTO 2021 で発表された LowMC および LowMC-M に対する攻撃はさらに改善され、いくつかの LowMC インスタンスを初めて破ることができた。

・Mind the TWEAKEY Schedule: Cryptanalysis on SKINNYe-64-256 [Asiacrypt 2022]

Lingyue Qin, Xiaoyang Dong, Anyu Wang, Jialiang Hua, Xiaoyun Wang

近年、特定の用途に合わせた共通鍵暗号の設計が話題となっている。EUROCRYPT 2020 で、Naito、Sasaki、Sugawara は、認証付き暗号 PFB_Plus の要件を満たすために、閾値実装フレンドリー暗号 SKINNYe-64-256 を考案した。やがて Thomas Peyrin が、SKINNYe-64-256 は新しい tweakey スケジュールにより安全性の期待を失う可能性があると指摘した。SKINNYe-64-256 の安全性の問題はまだ不明だが、Naito らは対応策として、SKINNYe-64-256 v2 を導入することを決定した。

本論文では、SKINNYe-64-256 の新しい tweakey スケジュールについて形式的な暗号解析を行い、tweakey スケジュールにおける予想外の差分キャンセレーション（differential cancellation）を発見している。例えば、連続する 30 ラウンドのうち、キャンセレーションが起こる回数は最大 8 回であり、予想される 3 回のキャンセレーションより大幅に多いことを発見した。この性質は、線形代数による tweakey の更新関数（LFSR）の解析によって導き出される。さらに、矩形攻撃、MITM 攻撃、不可能差分攻撃（impossible differential attack）について新たな発見をし、対応する自動化ツールに著者らの発見による新たな制約を適応させた。最終的に、SKINNYe-64-256 に対する 41 ラウンドの関連 tweakey 矩形攻撃が発見された。残っているセキュリティマージンは 3 ラウンドのみである。STK は任意のサイズの tweakey を受け付ける一方、SKINNY と SKINNYe-64-256 v2 は最大 4n の tweakey サイズまでしかサポートしていない。本論文では、SKINNY-64 のために、サポートする tweakey のサイズをさらに拡張する新しい tweakey スケジュールを紹介し、このスケジュールが STK と SKINNY のセキュリティ要

件を継承していることをフォーマルに証明する。

・Enhancing Differential-Neural Cryptanalysis [Asiacrypt 2022]

Zhenzhen Bao, Jian Guo, Meicheng Liu, Li Ma, Yi Tu

CRYPTO 2019 で Gohr は、十分に訓練されたニューラルネットワークが、従来の差分識別器よりも優れた暗号解読の識別タスクを実行できることを示した。さらに、通常ではない鍵推測戦略を適用することで、最新のブロック暗号 Speck32/64 に対する 11 ラウンドの鍵回復攻撃を行い、公表された最先端の結果を改善した。このことから、次のような疑問が浮かぶ:機械学習(ML)は従来の手法に対してどの程度の優位性があるのか、また、現代暗号の暗号解読においてその優位性は一般的に存在するのか？

本論文では、最初の疑問に対する答えとして、より拡張されたラウンド数削減型の Speck32/64 に対して、ML を用いた鍵回復攻撃が考案されている。その結果、改良型 12 ラウンド攻撃と、初の実用的な 13 ラウンド攻撃が達成された。この成果において本質的な点は、ML ベースの攻撃における古典的な要素、すなわちニュートラルビットを強化することにある。2 番目の質問に対しては、ラウンド削減された Simon32/64 上で様々なニューラルネットワーク識別器を生成し、純粋な差分ベースの対応するものとの比較を行っている。

・Optimizing Rectangle Attacks: A Unified and Generic Framework for Key Recovery [Asiacrypt 2022]

Ling Song, Nana Zhang, Qianqian Yang, Danping Shi, Jiahao Zhao, Lei Hu, Jian Weng

矩形攻撃はブロック暗号に対して非常に強力な暗号解読法である。矩形識別器がある場合、鍵回復攻撃を可能な限り効率的に行うことが期待できる。文献上では、矩形鍵回復攻撃のアルゴリズムは 4 種類存在している。しかし、その性能はケースバイケースである。また、攻撃方法が最適化されていないアプリケーションも数多く存在する。

本論文では、矩形鍵回復攻撃について深く考察し、あらゆる攻撃パラメータに対応可能な統一的かつ汎用的な鍵回復アルゴリズムが提案される。特に、従来の 4 つの矩形鍵回復アルゴリズムをカバーするだけでなく、従来は見落とされていた 5 種類の新しい攻撃も明らかにされている。また、新しい鍵回復アルゴリズムとともに、最適な攻撃パラメータを自動的に見つけるフレームワークを提案し、新しいアルゴリズムを用いて矩形攻撃の時間複雑性を最小にする。新しい鍵回復アルゴリズムの効率性を実証するため、既存の識別器に基づく Serpent、CRAFT、SKINNY、Deoxys-BC-256 に適用し、一連の改良型矩形攻撃が得られた。

・Optimising Linear Key Recovery Attacks with Affine Walsh Transform Pruning

[Asiacrypt 2022]

Antonio Flórez-Gutiérrez

線形暗号解読は、ブロック暗号に対する鍵回復攻撃の主要な族の 1 つである。いくつかの論文では、高速な Walsh 変換を使用することで、その時間計算量を低減できる可能性に注目している。これらの先行研究は、鍵回復ラウンドの構造を無視して、任意のブール関数として扱っていた。本論文では、Walsh 変換のための新しいアフィン枝刈り技術を使用して、これらの関数の Walsh スペクトルのゼロを利用することによって、これらのアルゴリズムの時間とメモリの計算量を最適化した。これらの新しい最適化戦略は、DES に対する改良された攻撃と 29 ラウンドの PRESENT-128 に対する最初の既知の攻撃という 2 つの応用例で紹介される。

・Stretching Cube Attacks: Improved Methods to Recover Massive Superpolies [Asiacrypt 2022]

Jiahui He, Kai Hu, Meiqin Wang, Bart Preneel

キューブ攻撃は、共通鍵暗号の代数的性質を利用して、特殊な多項式である superpoly を復元し、その後に秘密鍵を復元するものである。対応するブール関数の ANF (algebraic normal form) が利用できない場合、分割特性に基づくアプローチにより、巧妙な方法で正確な Superpoly を復元することができる。しかし、Superpoly を復元するための計算コストは、暗号のラウンド数が増加するにつれて法外に大きくなる。例えば、ASIACRYPT2021 で提案された NMP (nested monomial prediction) は、 Trivium の 845 ラウンドまでの解析に留まっている。この NMP 技術のボトルネック、すなわち単項トレイルの数が多すぎて解けないモデルを緩和するために、著者らは superpoly に寄与する特定の中間ラウンドのいわゆる valuable term に焦点を当てている。2 つの新しい技術、すなわち、NBDP (Non-zero Bit-based Division Property) と CMP (Core Monomial Prediction) を導入し、MP の MILP モデルと比較して、より単純な MILP モデルを実現することが可能となる。CMP 技法は、valuable term を回復する計算量の点で、monomial prediction よりも大幅な改善をもたらすことが示されている。分割統治法とこれら 2 つの新しい技術を組み合わせることで、より効果的に valuable term を捕らえ、superpoly に何も貢献しない中間項に対する計算リソースの浪費を避けることが可能となる。 その結果、以前の攻撃の計算コストを大幅に削減することができ、846、847、848 ラウンドの Trivium、192 ラウンドの Grain、895 ラウンドの Kreyvium、776 ラウンドの Acorn が実用時間で復元された。さらに、Möbius 変換の内部的な性質を調べることで、鍵の全ビットを含む superpoly を用いた鍵回復方法を示し、対象となる暗号に対して最適な鍵回復攻撃を行うことができるようになった。

・On the Field-Based Division Property: Applications to MiMC, Feistel MiMC and GMiMC [Asiacrypt 2022]

Jiamin Cui, Kai Hu, Puwen Wei, Meiqin Wang

近年、MPC や ZKP などの高度な暗号プロトコルが実用化され、AO 暗号と呼ばれる有限体上の共通鍵暗号が開発されている。このような AO 暗号は、代数的攻撃、特に高階の差分攻撃に対して脆弱であることが指摘されている。そのため、代数的次数の増大を注意深く評価することに意義がある。しかし、AO 暗号の次数推定は、一般的で正確な方法がないため、暗号解析者の課題となっている。

本論文では、代数的次数の上界を求めるための最先端のフレームワークである division property を、バイナリ体から$\mathbb{F}_{2^n}$をスコープにいれた範囲に拡張する。

これは、AO 暗号の代数的次数を検出する汎用的な手法であり、Feistel 暗号にも適用可能である。この一般化された division property における著者らのアイデアは、ブロック暗号の多項式表現がある特定の単項式を含んでいるかどうかを評価することにある。演算の特徴を深く調べることで、SMT のビットベクトル理論を用いて効率的にモデル化できる体演算ベースの単項式の伝搬規則が紹介される。そして、代数的次数と単項式の指数との関係から、次数推定のための新しい検索ツールを構築することができる。

著者らは、この新しい枠組みを Feistel MiMC、GMiMC、MiMC などのいくつかの重要な AO 暗号に適用した。Feistel MiMC については、代数的次数の増大が本来の指数関数的な限界よりも大幅に遅くなることが示された。また、CRYPTO 2020 で提案された Feistel MiMC の 83 ラウンドの順列識別器よりもはるかに優れた、124 ラウンドまでの秘密鍵高次差分識別器が初めて提示された。また、データ計算量 2^{251} のフルラウンドゼロサム識別器も提示される。本手法は、より多くの分岐を持つ一般的な Feistel 構造に対してさらに拡張でき、GMiMC の実用的なインスタンスに対して、最大 50 ラウンドの高次の差分識別器を示すことが可能である。SP-network における MiMC については、本結果は Bouvier によって証明された正確な代数的次数と対応する。著者らはさらに、異なる指数をもつ MiMC のような方式に対する高階差分攻撃に対する安全性を保証するために、MiMC 仕様のラウンド数は十分ではないことを指摘した。

2.2.2. 公開鍵暗号に関する解読技術

・How to Backdoor (Classic) McEliece and How to Guard Against Backdoors [PQCrypto 2022]

Tobias Hemmert, Alexander May, Johannes Mittmann, Carl Richard Theodor Schneider

本論文では、McEliece 暗号をバックドア化し、バックドア化された公開鍵は通常の公開鍵と区別がつかないが、秘密鍵は効率的に取得できる方法を示す。暗号学上の理由から、McEliece は小さな乱数種$\boldsymbol{\delta}$を使用し、何らかの疑似乱数生成器 (PRG) により秘密鍵を決定

する乱数を生成する。本論文のバックドアメカニズムは、$\boldsymbol{\delta}$の暗号化を公開鍵にエンコードすることで機能する。$\boldsymbol{\delta}$を取り出すことで、(バックドアされた) 秘密鍵を効率的に復元することができる。興味深いことに、McEliece は$\boldsymbol{\delta}$を暗号化するために使用することができ、それにより著者らのバックドア機構を強力なポスト量子セキュリティ保証で保護することができる。この構成は、非圧縮秘密鍵に対する現在の古典的 McEliece NIST 標準提案に対しても有効であり、悪意を持ってバックドアを施した実装が広まる可能性がある。幸いなことに、このバックドアメカニズムは、Classic McEliece の提案で規定されているように、鍵生成後に$\boldsymbol{\delta}$を保存すれば、(バックドアされた) 秘密鍵の所有者によって検出することができる。したがって本結果は、実装者に対して、秘密鍵の内部に$\boldsymbol{\delta}$を格納し、バックドア機構から保護するために$\boldsymbol{\delta}$を使用するようにという強い助言を与えるものである。

・Improving Bounds on Elliptic Curve Hidden Number Problem for ECDH Key Exchange [Asiacrypt 2022]

Jun Xu, Santanu Sarkar, Huaxiong Wang, Lei Hu

Elliptic Curve Hidden Number Problem (EC-HNP) は、Boneh、Halevi、Howgrave-Graham によって Asiacrypt 2001 で初めて導入された。そして、ECDH のビット安全性を厳密に評価するため、Hidden Number Problem (HNP) の楕円曲線アナロジーとみなされる EC-HNP の Diffie-Hellman 亜種が PKC 2017 で発表された。この変種は、サイドチャネル攻撃の状況下で、ECDH 鍵交換の実用的な暗号解読にも利用できる。

本論文では、EC-HNP の Diffie-Hellman 亜種において、関係するモジュラー多変量多項式を解くための Coppersmith 法を再検討し、任意の正の整数d、与えられた十分に大きな素数p、$\mathbb{F}_p$上の楕円曲線において、ECDH 鍵のx座標の LSB を$1/(d+1)$出力するオラクルがあれば、全ビットを$\log_2 p$の多項式時間で計算するヒューリスティックアルゴリズムが与えられることが実証された。$d>1$のとき、ヒューリスティックな結果$1/(d+1)$は厳密な評価$5/6$とヒューリスティックな評価$1/2$の両方を著しく上回る。Coppersmith 法に含まれるヒューリスティックのため、固定された楕円曲線上での ECDH ビット安全性は得られない。しかし、小次元格子の NIST 曲線において、ヒューリスティックの有効性が実験的に検証する。

・Lattice sieving via quantum random walks [Asiacrypt 2021]

Johanna Loyer, André Chailloux

格子暗号は、ポスト量子暗号の有力な提案の一つである。最短ベクトル問題(Shortest Vector Problem、SVP) は格子暗号の暗号解析において最も重要な問題であり、多くの格子暗号はその困難性に基づく安全性を主張している。SVP に対する最良の量子アルゴリズムは Laarhoven によるもので、(ヒューリスティックに) 時間 $2^{0.2653d+o(d)}$ で実行される (dは格子の次元)。

本論文では、Laarhoven の結果を改良し、(ヒューリスティックに) 時間 $2^{0.2570\,d+o(d)}$

で実行されるアルゴリズムが提示される。また、本アルゴリズムの量子メモリと量子ランダムアクセスメモリの量を定量化し、時間とメモリの間のトレードオフを提示する。

・A Systematic Approach and Analysis of Key Mismatch Attacks on Lattice-Based NIST Candidate KEMs [Asiacrypt 2021]

Yue Qin, Chi Cheng, Xiaohan Zhang, Yanbin Pan, Lei Hu, Jintai Ding

格子ベースの KEM に対する鍵不一致攻撃の研究は、現在進行中の NIST のポスト量子暗号の標準化における暗号評価の重要な部分である。このような攻撃は数多く存在しているが、しかしこれらの KEM の鍵不一致攻撃に対する耐性を評価する統一的な手法はまだない。効率性の重要な指標は、そのような攻撃を成功させるために必要なクエリ数である。

本論文では、そのような攻撃に必要な最小平均クエリ数の下界を求める体系的アプローチの提案、開発が行われる。基本的な考え方は、クエリの下限を求める問題を最適なバイナリ復元木 (binary recovery tree、BRT) を見つけることに変換することである。この BRT において、下限の計算は、本質的には特定のシャノンエントロピーの計算となる。またこの最適 BRT のアプローチにより、いくつかの格子ベースの NIST 候補 KEM において、必要なクエリの数に関して、理論的な下限と実際の攻撃で観測された下限の間に大きなギャップがある理由を説明できる。さらにこれら既存攻撃に対する汎用的な改善方法が提案され、実験により確認された。提案された手法は、CCA 安全な NIST 候補 KEM に対するサイドチャネル攻撃を改善するために直接利用することができる。

2.2.3. ハッシュ関数に関する解読技術

・SwiftEC: Shallue--van de Woestijne Indifferentiable Function to Elliptic Curves [Asiacrypt 2022]

Jorge Chávez-Saab, Francisco Rodrı´guez-Henrı´quez, Mehdi Tibouchi

任意の値を楕円曲線上の点にハッシュ化することは、多くの暗号構成で必要とされるステップであり、長年にわたって多くの技術が提案されてきた。最初のものは Shallue と van de Woestijne によるもので、有限体上のすべての楕円曲線に適用できるという興味深い性質を持っていた(ANTS-VII)。しかしながら、基礎体に対するランダムオラクルと合成したときに、「ランダムオラクルと区別できない」という望ましい性質は持っていなかった。

この制限を克服するために、Brier らの基礎研究(CRYPTO 2011)を皮切りに、様々なアプローチが検討されている。例えば、$f:\mathbb{F}_q \to E(\mathbb{F}_q)$ を Shallue-van de Woestijne (SW) とし、HとH'が 2 つのランダムオラクルの場合、$m \mapsto f\big(H(m)\big) + f(H'(m))$はラン

ダムオラクルと区別できないことが示されている。この方法は、素直な、しかし区別不可能ではない関数$m \mapsto f(H(m))$に比べて計算コストが 2 倍かかるという欠点がある。これまでの他のほとんどの解決策も同じ問題で、fのような単純な符号化写像が 1 回の基礎体指数計算で済むのに対し、それらは少なくとも 2 回の指数計算と同じコストがかかっている。最近、Koshelev (DCC 2022)は、1 つの指数化のコストで無分別ハッシュの最初の構成を提供したが、それは曲線の非常に特定のクラス (j不変量 0 のものもある) に対してのみであり、より広く適用できそうにない技術を使っていた。

本研究では、この長年の未解決問題を再検討し、SW 写像を 1 パラメータ族$(f_u)_{u\in\mathbb{F}_q}$に拡張した上で、独立なランダムオラクルH, H'を用いた$F: m \mapsto f_{H'(m)}(H(m))$が、ランダムオラクルと区別不可能であることを確認した。さらに、非常に大きな曲線のクラスにおいて、1 パラメータ族は有理的にパラメトライズされ、それにより小さなfとほぼ同じコストでFを計算することができる。最終的に、ほとんどの曲線に対して単一の指数化で無差別的ハッシュを達成することに成功した。この新しいアプローチは、任意の楕円曲線の点を一様に近いランダムな文字列として表現する Tibouchi (FC 2014)の Elligator Squared 技術の改良型をもたらすものである。

2.2.4. 署名に関する解読技術

・Breaking Rainbow Takes a Weekend on a Laptop [CRYPTO 2022,PQCrypto 2022]

Ward Beullens

本研究で、NIST のポスト量子暗号標準化プロジェクトで最終選考に残った 3 つの署名方式の 1 つである Rainbow 署名方式に対する新しい鍵回復攻撃を導入された。この新しい攻撃は、NIST に提出されたすべてのパラメータセットに対して既知の攻撃を上回り、SL 1 パラメータに対する鍵回復を実用している。具体的には、第二ラウンドに提出された SL 1 パラメータの Rainbow 公開鍵が与えられた場合、提案された攻撃は標準的なノートパソコンで平均 53 時間（1 週末）の計算時間で対応する秘密鍵を返す。

・Improving Support-Minors rank attacks: applications to GeMSS and Rainbow [CRYPTO 2022]

Pierre Briaud, Javier Verbel, Daniel Smith--Tone, Ray Perlner, Daniel Cabarcas, John Baena

Support-Minors (SM) 法は、NIST のラウンド 3 候補である GeMSS と Rainbow に対する最近の攻撃に見られるように、これまで利用不可能だったランク特性を持つ多変量解析を攻撃する新しいルートを切り開いた。本論文では、この SM 方式をより深く研究し、この Support-Minors 方式に基づく GeMSS への攻撃を大幅に改善したものが提案されている。GeMSS はすでに最近の攻撃の影響を受けていたが、この攻撃はさらなる影響を与

え、単にパラメータのサイズを大きくする、最近の射影技術（projection technique）を適用することによっては、このスキームを修復することは不可能となった。例えば、GeMSS128 のパラメータセットに対する提案された攻撃は、推定時間計算量 2^{72} であり、射影技術により方式を修復した場合は、署名にかかる時間が非現実的なファクター 2^{14} で遅くなってしまう。もう一つの貢献は、大規模 SM システムにおいて、XL 戦略のメモリアクセスコストが懸念される場合に、Block-Wiedemann アルゴリズムをサブルーチンとして使用し、そのコストを削減できる最適化を提案することである。とあるメモリコストモデルにおいて、矩形 MinRank 攻撃は、実際にすべてのラウンド 3 の Rainbow パラメータセットのセキュリティをその目標セキュリティ強度以下に低下させ得ることを示し、同じメモリコストモデルを用いて、Rainbow 開発チームによって主張されていた下界と矛盾することが示される。

・A New Fault Attack on UOV Multivariate Signature Scheme [PQCrypto 2022]

Hiroki Furue, Yutaro Kiyomura, Tatsuya Nagasawa, Tsuyoshi Takagi

多変量署名方式の一つである unbalanced oil and vinegar 署名方式（UOV）は、量子攻撃に対する安全性が期待されている。暗号システムの安全性を実用的に実現するためには、計算誤差を発生させて安全性を破綻させる故障攻撃などの物理攻撃に対する安全性に対処する必要がある。

本研究では、秘密鍵に発生する故障を利用した UOV に対する新たな故障攻撃を提案する。提案攻撃では、まず秘密鍵に発生する故障を利用して秘密鍵の線形写像の一部を復元し、その後、公開鍵システムを変換する。その結果、提案攻撃は与えられた公開鍵システムを元のシステムよりも少ない変数で構成されるシステムに変換することができる。また、提案する攻撃を適用した後、既存の鍵回復攻撃を用いることで、元のシステムよりも少ない計算量で秘密鍵を回復することができる。シミュレーションの結果、100bit の安全性を満たす 2 つの実用的なパラメータセットに対して、提案する攻撃は約 80～90%の確率で 90bit の安全性しか持たないシステムに縮小できることを示した。また提案されている攻撃は、上記の場合よりも低い確率で、より小さなシステムを実現し、結果そのようなシステムをより効率的に破ることができることを示す。

・Breaking Category Five SPHINCS+ with SHA-256 [PQCrypto 2022, Best Paper Award]

Ray Perlner, John Kelsey, David Cooper

SPHINCS+は、NIST のポスト量子暗号（PQC）標準化プロセスの一環として標準化されたステートレスハッシュベースの署名方式である。その安全性の証明は、鍵付きハッシュ関数の DM-SPR（distinct-function multi-target second-preimage resistance）に依存している。SPHINCS+では、SHA-256 をベースにしたものなど、いくつかの鍵付きハッシュ関数のインスタンスが提案されている。PQC メーリングリストの Sydney Antonov による最近の観測では、

SHA-256 に基づく構成は、NIST に提出されたパラメータセットのいくつかについて、NIST カテゴリ 5 の DM-SPR を持たないことが実証されたが、この観測が偽造攻撃につながるかどうかは未解決のままだった。

本論文では、これらのパラメータセットの具体的な古典的安全性を約 40 ビット減少させる完全な偽造攻撃を与えることによって、この質問に肯定的に答える。提案された攻撃は、SPHINCS の WOTS+公開鍵に Antonov の技術を適用し、非常に限られたハッシュ値群に署名できる新しいワンタイムキーを導き出すことで機能する。この鍵から、元のハイパーツリー(hypertree)を少し改変したものを作成し、それを使って任意のメッセージに署名することで、有効であるかのように見える署名を得ることができる。

2.2.5. サイドチャネル攻撃の技術動向

・A Novel Completeness Test for Leakage Models and its Application to Side Channel Attacks and Responsibly Engineered Simulators [Eurocrypt 2022]

Si Gao, Elisabeth Oswald

今日のサイドチャネル攻撃のターゲットは、命令が並列に処理され、32 ビットのデータ語で動作する複雑なデバイスであることが多い。その結果、これらの最新デバイスでリークを発生させるのに関与する状態は大きく、不完全である可能性のある状態に基づいて評価（ワーストケース攻撃など)、シミュレータ、および(マスキング) 対策の仮定を建てると、結論が大幅に間違ってしまう可能性がある。

本論文では、想定される状態の「完全性」に対する新しい概念と、「崩壊モデル」に基づく効率的な統計的検定を提案する。この新しいテストは、グレーボックスの設定において、複数の 32 ビット変数を含む状態を回復するために使用することができる。さらにこの新しいテストがサイドチャネル攻撃分析に役立つことを説明し、既存の実装に対する新しい攻撃方向を明らかにする。また、統計的テストを適用することで、最新の漏洩シミュレータでさえ、それぞれのターゲットデバイスの利用可能な漏洩をすべて捕捉していないことを示す。

・Towards Micro-Architectural Leakage Simulators: Reverse Engineering Micro-Architectural Leakage Features is Practical [Eurocrypt 2022]

Si Gao, Elisabeth Oswald, Dan Page

リークシミュレータは、サイドチャネルリークを考慮したソフトウェアのテストを簡単かつ迅速に行えるという魅力的な機能を備えている。そのため、リークモデルの品質は非常に重要であり、これにはマイクロアーキテクチャのリークを忠実に含めることが含まれる。マイクロアーキテクチャのリークは、ARM Cortex M シリーズのようなローからミッドレン

ジの商用プロセッサでも現実的なものである。マイクロアーキテクチャの要素が公に知られていない場合、どのようにそれを記述すればよいのだろうか？という問に、グレーボックス状況で対応することは、当初は実行不可能と思われていた。

本論文では、最新のリークモデリング技術を用いて、商用プロセッサのマイクロアーキテクチャのリークの重要な要素をリバースエンジニアリングすることが可能であることを、初めて実証した。このアプローチでは、まず、パイプラインの各ステージのマイクロアーキテクチャのリークを復元し、グリッチを発生させることが知られている要素のリークを復元する。リバースエンジニアリングされたリーク機能を用いて、一般的なリークシミュレータ ELMO の拡張版を構築する。

・Private Circuits with Quasilinear Randomness [Eurocrypt 2022]

Vipul Goyal, Yuval Ishai, Yifan Song

関数fのためのt-private 回路とは、入力xのランダムな符号化を出力$f(x)$の符号化に対応付けるランダム化ブール回路 C であり、回路C内の任意のt本を探ってもxについて何も明らかにならないものである。t-private 回路は、サイドチャネル攻撃から組込み機器を保護するために使用することができる。このようなデバイスで新しいランダム性を生成するコストが高いことに動機づけられ、いくつかの研究がt-private 回路のランダム性計算量を最小化する問題を研究してきた。Coron ら(Eurocrypt 2020)による最もよく知られた上限は、sを回路のサイズとした時の、$O(t^2 c \log s)$-ビットである。本研究はこの上限を、入力エンコーダで使用されるランダムネスを含めて、$O(t \log s)$に改善した。さらにt-private 回路のステートフルな亜種にもこれを拡張する。この構成は、fの回路から無視できる故障確率でt-private 回路Cを生成する効率的なランダム化アルゴリズムが存在するという意味で、semi-explicit なものである。

・Efficiently Masking Polynomial Inversion at Arbitrary Order [PQCrypto 2022]

Markus Krausz, Georg Land, Jan Richter-Brockmann, Tim Güneysu

物理サイドチャネル解析は、組込み機器に実装されるポスト量子暗号方式にとって大きな脅威となる。しかし、多くの方式において安全な実装が見出されていないのが現状である。本論文では、ポスト量子暗号の鍵生成の主要な要素であるマスク多項式反転の効率的な解決策を提案している。このために、加法的マスキングとの効率的な任意次数変換を行う多項式-多倍長マスキング方式が導入されている。さらに、多項式の反転と乗算をマスキングスキームに統合する方法を示すことで、コストを大幅に削減している。また、Cortex-M4 上で 2 種類のポスト量子暗号方式に対する本アルゴリズムの性能を実証した。NTRU では、マスクなし反転と比較して、1 次マスク反転で 35%のオーバーヘッドを計測したが、BIKE では 11%とわずかなオーバーヘッドにとどまった。最後に、TVLA に基づくサイドチャネル解析の測定と実行により、1 次マスク反転に対する著者らのアルゴリズムの安全性を検証した。

・Efficiently Masking Polynomial Inversion at Arbitrary Order [PQCrypto 2022]

Thomas Schamberger, Lukas Holzbaur, Julian Renner, Antonia Wachter-Zeh, Georg Sigl

符号ベースのポスト量子アルゴリズム Hamming Quasi-Cyclic (HQC) は、NIST 標準化プロジェクトの第 4 ラウンド候補である。その第 3 ラウンド版以降、新しい誤り訂正符号の組み合わせ、すなわち Reed-Muller 符号と Reed-Solomon 符号の組み合わせが利用されており、そのため公表されている攻撃を適応する必要がある。

本論文の著者らは、CHES 2021 からの Uneo らによるパワーサイドチャンネル攻撃が、実装された Reed-Muller デコーダが固定された復号境界を持たないという事実を見逃しているため、実際には機能しないことを確認している。また、この研究では、再び攻撃を成功させることができる新しい攻撃戦略が提供されている。著者らの攻撃は、その成功を検証するためにシミュレーションに頼らず、HQC パラメータセットに対して高い確率で証明される。Guo らによるタイミングサイドチャンネル攻撃とは対照的に、著者らは必要な攻撃クエリを 12 分の 1 に減らし、タイミングオラクル固有の不確実性を排除することができる。ARM Cortex-M4 マイコン上の Reed-Solomon デコーダのパワーサイドチャンネルを利用した実用的な攻撃結果を示す。また、本攻撃手法が、関連研究のサイドチャネルをターゲットとした場合に、どのように利用可能であるかについて考察を行う。

最後に、情報セット復号を使用して、部分的に取得された秘密鍵に対する残りの攻撃複雑度を評価する。この研究は、HQC のすべての関連するビルディングブロックをサイドチャネルで安全に実装する必要性を再度強調するものである。

・A New Key Recovery Side-Channel Attack on HQC with Chosen Ciphertext [PQCrypto 2022]

Guillaume Goy, Antoine Loiseau, Philippe Gaborit

Hamming Quasi-Cyclic (HQC) は、NIST のポスト量子標準化手順のコードベース候補である。コードベース暗号の復号ステップはサイドチャネル攻撃に対して脆弱であることが知られており、HQC もその例外ではない。

本論文では、選択暗号文を用いた HQC に対する新しい鍵回復サイドチャネル攻撃が発表されている。本攻撃は、物理的にアクセス可能なマイコン上で静的秘密鍵が再利用されることを利用している。目標は、カプセル化解除の Reed-Muller 復号ステップ、より正確には Hadamard 変換をターゲットにして、静的な秘密鍵を回収することである。この関数はその拡散性で知られており、その性質をサイドチャネル解析により利用する。サイドチャンネルの情報は、Reed-Muller 符号のいくつかの復号パターンを区別するオラクルを構築するために使用される。このオラクルに対して、静的秘密鍵に関する完全な情報を与えるようなクエリを行う方法を示す。実験によれば、2 万回以下の電磁波攻撃痕跡で、復号に使用される静

的秘密鍵のすべてを取得することができる。最後に、本攻撃を阻止するためのマスキングに基づく対策を提示する。

2.2.6. その他に関する解読技術動向

・Quantum Algorithms for Variants of Average-Case Lattice Problems via Filtering [Eurocrypt 2022]

Yilei Chen, Qipeng Liu, Mark Zhandry

本論文では、以下の問題に対して、多項式時間量子アルゴリズムを示す。

・L^{∞}ノルムにおける、公開行列が非常に大きく、多項式的に大きい素数を法とする、法の数の半分から定数を引いた値でノルムが抑えられている場合の、最小整数解問題（Short Integer Solution、SIS)。

・あるパラメータにおける Extrapolated dihedral coset problem (EDCP)。

・多項式的に大きいサイズの法と、有界一様分布やラプラス分布といった特定のエラー分布における LWE ライクな量子状態が与えられたときの、誤差付き学習問題（Learning-With-Errors、LWE)。

標準的な形式の SIS、EDCP、LWE 問題は、最悪の場合、格子問題を解くのと同じぐらい難しい。しかし、著者らが解くことのできる亜種の問題は、ワーストケースでの格子問題を解くのと同じくらい難しいことが知られているパラメータ領域にはない。ただ、SIS と LWE については、古典的なアルゴリズムも量子多項式時間アルゴリズムも知られていなかった。EDCP については、著者らの量子アルゴリズムは Ivanyos ら (2018) の結果をわずかに拡張している。SIS と EDCP の変種に対する本アルゴリズムは、それらの問題から LWE への（より正確には LWE ライクな量子状態を与えられた LWE を解く問題への）既存の量子簡約を使用している。主な貢献は、フィルタリング技術を用いて、興味深いパラメータを持つ LWE 的量子状態が与えられた場合に、LWE を解くことである。

・Orientations and the supersingular endomorphism ring problem [Eurocrypt 2022]

Benjamin Wesolowski

本論文では、同種を基礎とする暗号における二つの重要な問題族と、それらがどう関連しているかを研究している。一つ目は超特異楕円曲線の自己準同型環の計算、二つ目は向きづけられた超特異曲線上の類群作用を用いた逆問題である。この 2 つの問題は、一般化リーマン仮説を仮定した多項式時間簡約によって、密接に関連していることが証明される。

著者は、本質的に等価な問題の 2 つのクラスを同定する。最初のクラスは、向きづけ

られた曲線の自己準同型環を計算する問題に対応する。CSIDH のような大規模な暗号システムの安全性は、劣指数時間で実行されるヒューリスティック量子アルゴリズムが知られている、このクラスに還元される。2 つ目のクラスは、向きづけ可能な曲線の自己準同型環の計算に対応する。本質的に全ての同種を基礎とする暗号系の安全性はこの第二のクラスに帰着されるが、このクラスでは、最もよく知られたアルゴリズムは未だに指数関数時間で実行されている。

本論文が提案する方式は、一般化だけでなく、既知の結果を強化するものもある。例えば、$\mathbb{F}_p$上で定義された曲線の場合、CSIDH の安全性は自己準同型環問題に劣指数時間で帰着できることが知られている。提案された方法は、CSIDH の安全性を多項式時間で自己準同型環問題に帰着する。また、そういった帰着はあり得ないと証明した議論にも反論している。

・Anonymity of NIST PQC Round 3 KEMs [Eurocrypt 2022]

Keita Xagawa

本論文では、NIST PQC Round 3 における全ての KEM の匿名性が調査される：Classic McEliece、Kyber、NTRU、Saber、BIKE、FrodoKEM、HQC、NTRU Prime (Streamlined NTRU Prime and NTRU LPRime)、SIKE が対象である。以下の結果が示されている。

・NTRU は決定論的 PKE が強く離散的シミュレーション可能であれば、量子ランダムオラクルモデル (QROM) において匿名性を持つ。NTRU は QROM において無衝突である。NTRU を KEM とし、適切な DEM から構成されるハイブリッド PKE 方式は匿名かつロバストである。BIKE, FrodoKEM, HQC, NTRU LPRime, SIKE についても同様の結果が、HQC の 3 つのパラメータセットのうち 1 つを除いて成り立つ。

・古典的な McEliece は、基礎となる PKE が強く離散的シミュレーション可能で、KEM と適切な DEM として構築されたハイブリッド PKE スキームが匿名であれば、QROM において匿名である。

・Grubbs, Maram, Paterson は、Kyber と Saber が QROM における現在の IND-CCA セキュリティ証明にギャップがあることを指摘した (EUROCRYPT 2022)。本論文では、Streamlined NTRU Prime にも QROM における IND-CCA セキュリティ証明のための技術的障害があることを示した。

これらは、Grubbs, Maram, and Paterson によって提起された NIST PQC Round 3 KEMs の匿名性と頑健性を調査する問題に答えるものである。以上の結果は、KEM の基礎となる PKE の強い disjoint-simulatability と、KEM の強い擬似ランダム性と、スムース性/スパース性を主なツールとして用いているが、これらは独立した興味を惹きうる。

・On the Impossibility of Key Agreements from Quantum Random Oracles [CRYPTO 2022]

Per Austrin, Hao Chung, Kai-Min Chung, Shiuan Fu, Yao-Ting Lin, Mohammad Mahmoody

本論文では、2018 年に Hosoyamada と Yamakawa によって公に提起された以下の問題を研究する：局所的な量子計算能力を持つパーティ A、B は、ランダムオラクルと古典通信によってのみ鍵合意できるか？（A、B は量子重ね合わせでランダムオラクルを問い合わせることができることに注意せよ）。上記の質問について最初の進展を行い、次のことを証明する。

・一方が古典、他方が量子の場合、合計d回のオラクル問い合わせを行い、確率 1 で鍵合意する限り、$O(d^2)$回の古典オラクル問い合わせで鍵合意を破る方法が常に存在する。

・両者がランダムオラクルに対して量子問合せを行える場合には、ある自然な予想が正しいと仮定した上で、ランダムオラクルに対する古典問合せが$\mathrm{poly}(d)$回で攻撃できることを示した。この予想を大まかにいえば、任意の 2 つの次数dの実係数多項式で、influence（多項式係数の大きさのようなもの）が$\delta = 1/\mathrm{poly}(d)$で抑えられているならば、その積はブール超立方体上における値が非零である、という主張である。著者らは、指数関数的に小さい影響に対しこの予想を証明し、任意の鍵合意プロトコルに対する（無条件の）古典的な$2^{O(md)}$回クエリを行う攻撃（mはランダムオラクルの出力長）を導き出す。

著者らの攻撃は古典的であるため、一般にそのような古典的な攻撃を見つけることが可能かどうかを問う。古典的なシミュレーションが可能であることを示す「シミュレーション予想」が偽であれば、古典的な敵によって破ることができない量子プロトコルが存在することになり、このアプローチに対する障壁を証明することができる。

・Some Easy Instances of Ideal-SVP and Implications to the Partial Vandermonde Knapsack Problem [CRYPTO 2022]

Katharina Boudgoust, Erell Gachon, Alice Pellet-Mary

本論文では、Pan ら（Eurocrypt'21）と Porter ら（ArXiv'21）の研究を一般化し、理想格子が最短ベクトル問題（SVP）の容易なインスタンスを定義する簡単な条件を提供する。すなわち、イデアルを固定する自己同型が多いほど、最短ベクトルを見つけるのが容易になることを示す。この観測は、ガロア体における素イデアルに対して既になされていたが、著者らはこれを任意の数体の任意のイデアルであって、その素分解が ramified でない場合に一般化する。この結果を暗号技術に応用し、部分的 Fourier 回復問題（partial Fourier recovery problem）としても知られる、部分的 Vandermonde ナップザック問題の特定のインスタンスが、多項式時間で解けることを示す。さらに本論文では、概念実証として、この攻撃を実装し、文献で提案されている具体的なパラメータ設定に対して、それらの特定のインスタンスを解くことに成功している。ランダム

なインスタンスに対して、無視できない確率で格子の次元を半分にすることに成功している。

・Accelerating the Delfs-Galbraith algorithm with fast subfield root detection [CRYPTO 2022]

Maria Corte-Real Santos, Craig Costello, Jia Shi

本論文では、与えられた超特異楕円曲線$E/\mathbb{F}_{p^2}$から部分体上の楕円曲線$E'/\mathbb{F}_p$への同種を求める新しいアルゴリズムを与える。これは、一般的な超特異曲線同種問題に対するDelfs-Galbraithアルゴリズムのボトルネックとなるステップである。本手法の核となるのは、高価な根探索アルゴリズムを避けつつ、多項式$f \in L[X]$が部分体$K \subset L$に根を持つかどうかを高速に決定する新しい方法である。多項式が超特異j-不変量で評価されるℓ番目のモジュラー多項式 $f = \Phi_{\ell,p}(X,j) \in \mathbb{F}_{p^2}[X]$ である特別な場合に、これは対応する楕円曲線と部分体上の曲線を結ぶℓ-同種が存在するかどうかを効率的に決定する方法を提供する。従来のDelfs-Galbraithウォークと合わせて、この方法で多くのℓ-同種な近傍を調べることで、単位時間あたりの超特異曲線のより大きな割合を探索することができる。この改良されたアルゴリズムの漸近的な計算量 $\tilde{O}(p^{1/2})$ は元の Delfs-Galbraithアルゴリズムと変わらないが、本論文で為された理論解析と実際の実装は共に部分領域探索の実行時間を大幅に短縮することを示す。これは、一般的な超特異性同型問題（同型暗号の基礎となる問題）の具体的な難しさに新しい光を当て、Delfs-Galbraithが最もよく知られた古典攻撃であるB-SIDHやSQISignなどのスキームのビットセキュリティに直接影響を与えるものである。

・Partial Key Exposure Attacks on BIKE, Rainbow and NTRU [CRYPTO 2022]

Andre Esser, Alexander May, Javier Verbel, Weiqiang Wen

いわゆる部分的鍵公開攻撃では、例えばサイドチャネルの漏洩によって秘密鍵に関する何らかの情報を得ることができる。この情報は秘密鍵のビット数の一部であったり（消去モデル）、秘密鍵の誤ったバージョンであったり（エラーモデル）する。目標は、漏洩した情報から秘密鍵を復元することである。RSA暗号などとは対照的に、ほとんどのポスト量子暗号は鍵の部分的な漏洩攻撃に対して通常耐性があるという通説がある。本論文は、コードベース、多変量、格子ベースの方式（BIKE、Rainbow、NTRU）に対する部分鍵公開攻撃を構築することで、この考え方に強く疑問を投げかけるものである。提案されている攻撃は、現代の耐量子計算機暗号が効率化のために本質的に使用している冗長性を利用する。理論面では、多項式時間の鍵回復攻撃を可能にする非自明な情報漏洩の境界を示す。例えば、全ての方式において、秘密鍵のビットの一定割合を知ることで、多項式時間で完全な鍵を復元することができる。多項式時間攻撃にこだわらなくとも、著者らの攻撃のほとんどは、大きな消去率やエラー率まで十分に拡張でき、実行可

能であり続ける。例えば、BIKE の場合、秘密鍵のビットの半分が消去された場合、または秘密鍵のビットの 4 分の 1 が欠陥のある場合に、60 ビット程度の攻撃複合度を得ることができた。この結果は、現代の耐量子計算機暗号システムにおいて、非常に誤りやすい鍵の漏洩であっても、秘密鍵の完全な復元が可能であることを示している。

・Improvement of algebraic attacks for superdetermined MinRank [PQCrypto 2022]

Magali Bardet, Manon Bertin

MinRank (MR) 問題は、多くの暗号アプリケーションで発生する計算問題である。PQCrypto 2019 にて Verbel らは、双線形 Kipnis-Shamir (KS) モデリングから始まる、MinRank 問題の超決定インスタンスを解く新しい方法を紹介した。彼らは、特定の Macaulay 行列の線形代数を用い、変数の 1 ブロックによる初期方程式の倍数、いわゆるカーネル変数のみを考慮する。その後、ASIACRYPT 2020 にて Bardet らは、カーネル変数に関連する Plücker 座標、すなわち、KS モデリングにおけるカーネル行列の最大小行列式を考慮した新しいサポートマイナーモデリング (Support Minors modeling, SM) を導入した。

本論文では、KS モデリングと SM モデリングの間のリンクについて、任意のインスタンスについて完全な代数的説明を与える。そして、超決定 MinRank インスタンスは SM モデリングの容易なインスタンスと見なすことができることを示す。特に、可能な限り小さい次数 (″first degree fall″) と可能な限り小さい変数数で計算を実行することが、必ずしも最善の戦略ではないことを示す。一般的なランダムインスタンスに対する攻撃の計算量の推定値を与える。

これらの結果を、NIST の標準化プロセスの第一ラウンドに提出された DAGS 暗号に適用した。Bardet らで改良された Barelli と Couvreur による代数的攻撃は、特定の superdetermined MinRank インスタンスであることを示す。ここでは、インスタンスは一般的なものではないが、DAGS から特定のインスタンスを解析し、特定のインスタンスを解くために最適なパラメータ (短縮位置の数) を選択する方法を提供することが可能であることを示している。

・Attack on SHealS and HealS: the Secdond Wave of GPST [PQCrypto 2022]

Steven D. Galbraith, Yi-Fu Lai

Asiacrypt 2021 の Fouotsa と Petit の同種に基づく公開鍵暗号化方式 SHealS と HealS、および鍵交換方式 HealSIDH を暗号解析した。

本論文では、効率的で安全な静的-静的 (static-static) 鍵交換プロトコルや、同種からの公開鍵暗号化 (PKE) を得ることをテーマとしている。静的-静的プロトコルは、参加者が公開鍵を随時変更することなく、目的のプリミティブを実行することを可能にする。これは CSIDH を用いて可能であり、CSIDH はいくつかの同種に基づく暗号プリミティブの構築に用

いられている。しかし(本論文が公開された当時は)、Kuperberg アルゴリズムに基づく CSIDH に対する亜指数関数的な攻撃のため、SIDH がより強固な基盤を提供していた。同種からくる頑健な基礎仮定を持つ効率的なプロトコルは、いまだ未解決の問題である。

SIDH 系が静的-静的特性を実現するための主なボトルネックは、適応的 GPST 攻撃に集約される。この攻撃は、悪意のある Bob が Alice の秘密鍵を各ハンドシェイクからビット単位で抽出することを可能にし、その逆もまた可能である。この攻撃への対策としては、ゼロ知識証明の埋め込みや k-SIDH 法が知られている。しかしこれらの対策では、必然的に複数の並列な同種計算が発生するため、導出された方式は実用的ではない。これを解決するために、Fouotsa と Petit(Asiacrypt'21) は、同種図式の可換性を利用した新しい鍵検証機構を持つ SIDH の変種を発表した。この方式は、素数長を 2 倍にした SIKE よりも少ない同種計算で、上述の他の既知の解決策よりもはるかに効率的である。さらに、適応的な攻撃に耐える静的鍵交換と PKE 解を同種から与えられるということが主張されている。

本研究では、Fouotsa と Petit により提示されたプロトコルに対する適応的な攻撃を提示することで、この主張に反論している。本論文の攻撃は、彼らの中核的な成果である鍵検証機構の欠陥に基づき、SHealS、HealS、HealSIDH を構築するものである。この攻撃は GPST 攻撃の単純な調整とみなすことができ、驚くべきことに、秘密鍵を適応的に回復するために、SIDH に対する GPST 攻撃と同じ数のオラクルクエリーを必要とする。言い換えれば、追加の鍵検証機構は元の SIDH 方式に対してプロトコルを遅くするだけでなく、適応的攻撃を防ぐ上でこの方式に利点を与えないということを著者らは主張している。

・Post-Quantum Insecurity from LWE [TCC 2022]

Alex Lombardi, Ethan Mook, Willy Quach, Daniel Wichs

本論文では、多くの基本的な暗号プリミティブについて、learning-with-errors (LWE) 仮定の下で古典的な安全性を証明しても、量子後の安全性が保証されないことを示す。これは、LWE がポスト量子的安全性を持つと広く信じられているにもかかわらず、証拠は与えられていないことを意味している。その代わりに、たとえ仮定がポスト量子的安全性であっても、暗号構成内部でポスト量子的安全性の欠如が生じうることを示している。具体的には、擬似乱数関数、CPA 暗号、メッセージ認証コード、署名、CCA 暗号など、ブラックボックス化による LWE で古典的に安全であることが証明されているが、ポスト量子安全性が証明されていない暗号を構築している。これらの暗号方式はすべてステートレスで非対話的だが、攻撃者が暗号方式に対してオラクルクエリーを行うことができる対話型ゲームによってその安全性が定義されている。多項式時間量子攻撃者は、暗号システムに対して数回の古典的なクエリを行うだけでこれらの方式を破ることができ、場合によっては 1 回のクエリで十分な場合もある。これまでは、ステートフルもしくは対話的なプロトコルに対して、ポスト量子仮定での安全性の欠如を示す例しかなかった。さらに、ブラックボックス安全性証明を持つステートレスもしくは非対話

的暗号システムについては、方式に対する量子攻撃は仮定に対する量子攻撃に変換されるはずだというフォークロアに対し、本研究はそうではないことを示している。主な手法は、上記のプリミティブの対話型セキュリティゲーム内に対話型プロトコルを注意深く埋め込むことである。また、古典的な送信者と受信者の間で、量子受信者が 3 ラウンド目に秘密メッセージを知るが、LWE を仮定すると古典的受信者は知らない、3 ラウンドの ″quantum disclosure of secrets (QDS) ″プロトコルを示したのも興味深い結果である。

・The Parallel Reversible Pebbling Game: Analyzing the Post-Quantum Security of iMHFs [TCC 2022]

Jeremiah Blocki, Blake Holman, Seunghoon Lee

古典的な（並列）石移動ゲーム（以下、pebbling ゲーム）は、静的なデータ依存グラフを持つ関数の評価に必要なリソース（空間、時空間、累積メモリ）を分析することができる、有用な抽象化である。暗号分野で特に興味深いのは、有向非循環グラフ (DAG) Gと暗号ハッシュ関数Hによって定義されるデータ非依存な memory-hard 関数$f_{G,H}$である。グラフGの pebbling 計算量は、$f_{G,H}$を複数回評価するコストと、関数$f_{G,H}$の固定された定義域に対するブルートフォース原像攻撃に対する総コストを特徴付けるものである。古典的な攻撃者は関数$f_{G,H}$を、少なくとも定義域の濃度（＝mとする）回評価する必要があるが、Grover のアルゴリズムを実行する量子攻撃者は関数$f_{G,H}$を評価する量子回路$C_{G,H}$への、ブラックボックス呼び出しを$O(\sqrt{m})$回行うだけで済む。したがって、量子攻撃のコストを分析するためには、量子回路$C_{G,H}$の時空間コスト（幅×深さ）を理解することが極めて重要である。

古典的な計算では、グラフGに対する効率的な pebbling 戦略は$f_{G,H}$を評価するアルゴリズムに相当するが、本研究ではまず、グラフGに対する pebbling 戦略は必ずしも同等の計算量を持つ量子回路の存在を意味しないことを指摘している。この観測を動機として、新しい並列な可逆 (reversible) ペブリングゲームが導入され、量子計算機における量子削除不可能定理 (No-Deletion Theorem) によって課される追加的な制限を捕らえる。この新しい可逆 pebbling ゲームを応用して、いくつかの重要なグラフの可逆時空間計算量が解析された：線グラフ、Argon2i-A、Argon2i-B、そして DRSample である。具体的には、以下が為されている

(1) サイズNの線グラフに対しては、可逆的な時空間計算量が最大 $O(N^{1+2/\sqrt{\log N}})$ であることを示す。
(2) 任意の (e,d)-簡約 DAG に対しては、可逆時空間計算量が $O(Ne+dN2^d)$ であることを示す。特に、Argon2i-A と Argon2i-B の可逆時空間計算量はそれぞれ最大 $O(N^2 \log\log N/\sqrt{\log N})$、$O(\frac{N^2}{(\log N)^{1/3}})$であることが示唆される。

(3) DRSample に対しては、可逆時空間計算量が最大$O(N^2 \log\log N/\log N)$であることを示す。また、Alwen and Blocki の深さ可変型グラフに対する(非可逆)pebbling 攻撃を拡張した可逆 pebbling の累積 pebbling コストについて研究する。

・Full Quantum Equivalence of Group Action DLog and CDH, and More [Asiacrypt 2022]

Hart Montgomery, Mark Zhandry

暗号学的群作用は、標準的な暗号群を緩和し、構造を少なくしたものである。構造を持たないことで、Shor のアルゴリズムがあるにもかかわらず、もっともらしく量子耐性を持ち、なおかつ多くの応用が可能である。群作用の最も有名な例は、楕円曲線の同種から構築されたものである。

本論文の主結果として、アーベル群作用に対する CDH は離散対数問題と量子的に等価であるということが証明された。Galbraith らは以前、CDH を完全に解くと量子的に離散対数問題と等価であることを示したが、この結果はどんな無視できない優位性(non-negligible advantage)に対しても有効である。また、群作用と同種プロトコルに関する他のいくつかの問題についても検討している。

・Cryptographic Primitives with Hinting Property [Asiacrypt 2022]

Navid Alamati, Sikhar Patranabis

Hinting PRG は、PRG のシードに関する循環安全性(circular security)の「決定論的」な形式を持つ PRG の(潜在的に)より強い亜種である(Koppula and Waters, CRYPTO 2019)。ヒンティング PRG は、多くの暗号アプリケーション、特に CCA 安全な公開鍵暗号とトラップドア関数を可能にする。本論文では、Hinting 性質を持つ暗号プリミティブを研究し、以下の結果を得ている。

・Hinting PRG の設計において、巡回群や同種ベースの群作用に対するある種の決定論的仮定から、より概念的に単純な新規のアプローチを提示することで、既存のアプローチと比較してより単純な安全性証明を可能にした。

・Hinting 性質を弱 PRF に自然に拡張した Hinting 弱 PRF を導入し、任意の Hinting 弱 PRF から循環もしくは KDM 安全な共通鍵暗号を実現する方法を示す。Hinting PRG を構築するための単純なアプローチは、同じ決定論的仮定の集合から Hinting 弱 PRF を実現するために拡張できることを実証する。

・Hinting 性質の強化版である機能的 Hinting 性質を提案し、秘密のシードや鍵の関数に関するヒントが存在する場合でも安全性を保証する。本論文では、平易な Hinting PRG および Hinting 弱 PRF を実現するための簡単な技術を基に、特定の関数(族)に対する機能的な Hinting PRG および Hinting 弱 PRF を実現する方法を示す。また、ある種の代数的特性を持つ帰納的 Hinting 弱 PRF が、ブラックボックス方式で KDM 安全な公開鍵暗号を実現する

上で適用可能であることを実証する。

・最後に、ランダムオラクルだけが与えられたこれらのプリミティブの単純な実現を使って、Hinting 弱 PRF（および Hinting PRG）を公開鍵暗号からブラックボックス的に分離できることを示す。

・A New Isogeny Representation and Applications to Cryptography [Asiacrypt 2022]

Antonin Leroux

本論文では、次数Dの巡回同種が存在する超特異楕円曲線E_1とE_2による三つ組(D, E_1, E_2)の集合におけるメンバーシップの知識として定義される、同種表現について研究している。このメンバーシップ証明は、電子署名の構築や暗号鍵の検証など、いくつかの基本的な暗号学的応用があることが知られている。本論文の前半では、同種についての既知の結果を言語と証明の枠組みで再解釈し、同種表現が NP に属することを Deuring 対応から自然に導く。

本論文の主な貢献は、(大きな) 素数の場合を対象とした新しい同種表現である suborder 表現を設計したことである。新手法の核心は、余定義域の自己準同型環から注意深く選ばれた suborder 内の smooth ノルムの自己準同型を明らかにすることである。著者らは、これらの新しいメンバーシップの証明は、SubOrder to Ideal Problem (SOIP)という新しい計算問題の困難性の仮定の下で、同系列に基づく暗号の興味深い展望を開くだろうと主張している。特にその応用として、suborder 表現に基づく新しい NIKE である pSIDH が紹介されている。またその過程で、ある quaternion order の新しい族の中でノルムの等式を解くいくつかのヒューリスティックなアルゴリズムツールが開発された。これらの新しいアルゴリズムは、独立した興味を持たれるかもしれない。

・Group Action Key Encapsulation and Non-Interactive Key Exchange in the QROM [Asiacrypt 2022]

Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, Doreen Riepel

耐量子計算機暗号の文脈では、暗号学的群作用は、CSIDH における同種ベース暗号の一般化である。本研究では、過去に提案された 2 つの自然なプロトコル、群作用ハッシュ ElGamal 鍵カプセル化 (Group Action Hashed ElGamal key encapsulation mechanism、GA-HEG KEM) と、群作用ハッシュ Diffie-Hellman 非対話鍵交換 (Group Action Hashed Diffie-Hellman non-interactive key-exchange、GA-HDH NIKE) の安全性についての再検討が行われている。後者のプロトコルは、既に Post-Quantum WireGuard (S&P '21) や OPTLS (CCS '20) などの実用的なプロトコルで利用が検討されている。

本研究で、量子ランダムオラクルモデル (QROM) における 2 つのプロトコルのアクティブ安全性 (active security) は、DDH オラクルへの任意の量子アクセスをもつ攻撃者に対す

る群作用強 CDH 問題（Group Action Strong CDH）の非常に強い亜種に本質的に依存することが証明される。さらに、古典的な強い CDH 仮定、すなわち DDH オラクルへの古典的なアクセスを持つ CDH から、QROM の安全性を持つプロトコル変形が提案される。最初の変形は、鍵確認を用いるため、KEM の設定にのみ適用可能である。提案された第二の、しかしかなり効率の悪い変形は、Cash ら(EUROCRYPT '08)による技術に基づいている。特に、標準 CDH 仮定から、QROM 安全性を備えたアクティブ安全性を持つ最初の同種ベースの NIKE を生成する。

・Horizontal racewalking using radical isogenies [Asiacrypt 2022]

Wouter Castryck, Thomas Decru, Marc Houben, Frederik Vercauteren

本論文では、Castryck, Decru, Vercauteren が Asiacrypt 2020 で発表した、有限体上の楕円曲線間の間の、固定された小さな次数の長い列の計算における根基同種（radical isogeny）の使用に関する 3 つの主な未解決問題を扱う。まず、与えられた次数 N の根基同種を求めるための補間法を提示し、大きな関数体上の多項式を因数分解する必要を回避する。この方法を用いることで、$N \leqq 13$から$N \leqq 37$という範囲まで、自由に公式を求めることができる範囲を拡大できる。次に、既知の手法とアドホックな操作を組み合わせて、N≤19 に対してこれらの公式の最適化版を導き出すことで、いくつかのインスタンスを 2020 年に得られたものより 2 倍以上高速に実行した。第三に、$p \equiv 7 \pmod 8$の場合の$\mathbb{F}_p$上の、曲面に沿った超特異楕円曲線間のウォーク（すなわち、$\mathbb{F}_p$上の自己準同型環が$\mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$となる超特異楕円曲線の間の同種写像たち）において、根基同種を正しく選択する問題を解決した。これはNが偶数の場合非自明で、$N = 4$では PKC2022 で Onuki と Moriya によって初めて解決した。著者らは、Nが偶数の場合の予想を建て、$N \leq 14$に対してそれを証明する。これらの手法により、次のように CSIDH の実質的な高速化が得られた：次数16の同種を用いて、512 ビットの素数体上での次数2の同種の長い列の計算が 3 倍に、根基同種を用いた CSIDH の実装が 12%程度高速化された。

・Log-$\mathcal{S}$-unit lattices using Explicit Stickelberger Generators to solve Approx Ideal-SVP [Asiacrypt 2022]

Olivier Bernard, Andrea Lesavourey, Tuong-Huy Nguyen, Adeline Roux-Langlois

2020 年、Bernard と Roux-Langlois は、Pelle-Mary, Hanrot, Stehlé による PHS アルゴリズムに基づき、任意の数体上の理想格子に対する Approx-SVP を解く Twisted-PHS アルゴリズムを発表した。彼らは、拡大次数高々70かつ素数導手を持つ円分体上で実験し、主なボトルネックの 1 つは、log-$\mathcal{S}$-unit の格子の計算で、劣指数時間が必要であることを示した。本論文の主な貢献は、これらの実験をほとんどの導手mに対して拡大次数 210 までの円分体に拡張したことである。Bernard と Kučera による Stickelberger ideal の新しい結果に基

づいて、フルランク log-$\mathcal{S}$-unit 部分格子を構築するために明示的な生成器を用いて、完全な Twisted-PHS 格子の近似の役割を果たすことができる。最良の近似領域において、本論文の結果は、Twisted-PHS アルゴリズムが Cramer、Ducas、Wesolowski による CDW アルゴリズムを彼らの実験範囲において上回り、時にはその漸近ボリューム下限を破ることを示す。さらに、類数のプラス部分が $h_m^+ \leq O(\sqrt{m})$ であるという穏やかな制約のもと、これらの明示的な Stickelberger 生成器を用いて、CDW アルゴリズムのほぼ全ての量子ステップを削除することができる。

·A Non-heuristic Approach to Time-space Tradeoffs and Optimizations for BKW [Asiacrypt 2022]

Hanlin Liu, Yu Yu

NTRU 問題は、格子の中に例外的に短い非零ベクトルを含むという仮定のもとで、短い非零ベクトルを見つけるインスタンスとみなすことができる。さらに対象となる格子は、数体の整数環上の階数 2 の加群構造を持っている。この問題を加群一意最短ベクトル問題（module unique shortest vector problem)、略して mod-uSVP と呼ぶことにする。本論文では、NTRU 問題が mod-uSVP の単なる特殊なケースではなく、計算の観点からそれを代表するものであることを証明する 2 つの簡約法が示される。まず、ワーストケースの mod-uSVP をワーストケースの NTRU に還元する。これは、理想格子における短い非零ベクトルを求める問題である id-SVP のオラクルに依存している。そして、Pellet-Mary and Stehlé [ASIACRYPT'21]のワーストケース id-SVP からワーストケース NTRU への削減法を用いて、ワーストケース NTRU がワーストケース mod-uSVP と等価であることが示される。第二に、mod-uSVP のランダムな自己帰着を行う。著者らは mod-uSVP のインスタンスに対する、とある分布 D を提案している。この分布 D において無視できない確率でサンプリングされる mod-uSVP を解くことで、mod-uSVP をワーストケースで解くことができる。最初の結果を用いると、ワーストケースの mod-uSVP から、分布が D から継承されている場合での、平均的な NTRU インスタンスを解くことができるようになる。このワーストケースを解くことから平均的な場合を解くことへの帰着は、id-SVP のオラクルを必要としている。

·Nostradamus goes Quantum [Asiacrypt 2022]

Barbara Jiabao Benedikt, Marc Fischlin, Moritz Huppert

Kelsey と Kohno によって導入された Nostradamus 攻撃（Eurocrypt 2006）では、攻撃者はイテレーションされたハッシュ関数Hのハッシュ値yをコミットする必要がある。また後に与えられたプレフィックスPに対して、攻撃者は$H(P||S) = y$となる適切な″suffix explanation″ S を見つけることができなければならなかった。Kelsey と Kohno は、Hの圧縮関数（出力と状態がnビット）の$2^{2n/3}$回の評価による集め攻撃（herding attack）を示し、この攻撃を計算量の点で原像攻撃と衝突探索の間に位置づ

けた。

本論文は、量子敵対者に対する Nostradamus 攻撃の安全性を調査し、Nostradamus 問題に対して量子集め攻撃アルゴリズムを提案した。これは$n^{1/3}\ 2^{3n/7}$回の圧縮関数への代入を行うものであり、古典ケースにおける評価を大幅に改善している。また、量子集め攻撃は$2^{3n/7}$回の代入では実行できないことを証明し、著者らのアルゴリズムが（本質的に）最適であることを示す。また、ランダムな圧縮関数に対する一般的な Nostradamus 攻撃について、およそ$2^{\frac{3n}{7}-s}$回の評価についても議論されている（ここでsは敵対的に選択したサフィックスSの最大ブロック長である）。

資料３－３

2022 年度暗号技術評価委員会　電子メールによる審議とその結果

1　目的

CRYPTREC 暗号リストの改定のため、暗号技術検討会がパブリック・コメント（意見公募手続）を 2 月中に開始する予定である。2022 年度第二回暗号技術評価委員会の開催日は、2023 年 2 月 27 日（月）であるため、パブリック・コメント開始までに CRYPTREC 暗号リストに係る審議を終えておく必要がある。そのため、以下の内容を電子メールによる審議扱い[1]としたい。

2　審議内容

2.1　富士通株式会社からの暗号技術取り下げ申請への対応

ECDSA、ECDH 及び SC2000 の応募暗号について取り下げの申請（表 1）があったため、暗号技術評価委員会として以下の対応（表 2）を行う。

表 1：暗号技術取下げ申請書からの抜粋（表現は若干変更）

取り下げ理由	暗号アルゴリズムを取り巻く状況の変化により、窓口を継続する必要性がなくなったため。
ECDSA 及び ECDH	主流となっている AES などの標準と同様の取り扱いとして欲しい。
SC2000	普及していないため、取り下げの影響はない。

表 2：取り下げへの対応案

	理由
ECDSA 及び ECDH	取り扱いを応募暗号技術から CRYPTREC が選出した暗号技術に変更し、現状通り、電子政府推奨暗号リストに記載しておくことは妥当であると判断する。仕様書の参照先についても変更無しとする。
SC2000	応募者（社）の判断を尊重し、取り下げを認める。推奨候補暗号リストから当該暗号技術を削除することは妥当であると判断する。

2.2　CRYPTREC 暗号リスト改定に係る暗号技術の選定

毎年実施している暗号技術の安全性及び実装に係る監視及び評価[2]の活動に伴う小改定を経て、現在の CRYPTREC 暗号リストを維持してきている。

2.1 節の事項以外で、今年度の監視活動による変更がなければ、CRYPTREC 暗号リスト改定

---

[1] 暗号技術評価委員会の運営方針 項番 4

[2] 「CRYPTREC 暗号等の監視」、「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」及び「推奨候補暗号リストへの新規暗号（事務局選出）の追加」の 3 つの活動

に係る暗号技術の構成は現状のままとする。

3. 審議結果

電子メールによる審議の結果、委員から異論がなかったため、原案のとおり承認とする。

以上

資料３－４

2022 年度暗号技術調査 WG（耐量子計算機暗号）活動報告

1. 2022 年度暗号技術調査 WG（耐量子計算機暗号）活動報告の概要

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。2022 年度第 1 回暗号技術評価委員会において、2021 年度と同様に、PQC WG において下記 2 点について実施することが承認された。

(1) 耐量子計算機暗号の研究動向調査をもとに、主要な耐量子計算機暗号についてのガイドラインを 2021 年度から 2022 年度にかけて作成する。

(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」の予測図を更新する。

2. 委員構成（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）
委員：草川 恵太（NTT）
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：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案の作成

3.1. スケジュール（2021 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
2021 年度	第 1 回（9 月初旬頃を想定）	記載すべき項目及びその章立てを議論 記載する暗号方式の選定基準を議論 記載する暗号方式の候補を議論 執筆担当者を議論
	第 2 回（2 月を想定）	章立ての決定 記載する暗号方式の選定基準の決定 記載する暗号方式の候補の決定 執筆担当者の決定
2022 年度	第 1 回（8 月下旬頃を想定）	執筆内容の中間報告
	第 2 回（2 月を想定）	執筆内容の最終報告

3.2. 2021 年度第 1 回 PQC WG（9/7）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

  ガイドラインは暗号理論に精通していない利用者を、調査報告書は暗号理論の研究者や技術者を対象とする。そのため、基本的には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。ただし、暗号理論に精通していない利用者のために、PQC の活用方法をガイドラインでは記載し、調査報告書には記載しない。

- 記載すべき項目及び章立てと執筆担当者

	執筆担当者
i. はじめに	事務局（青野、篠原、高安）
ii. PQC の活用方法（ガイドラインのみ）	伊藤委員（ガイドラインのみ）
iii. 格子に基づく暗号技術	下山委員、安田（雅）委員、事務局（青野）
iv. 符号に基づく暗号技術	草川委員
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

- 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及び候補について

  公開鍵暗号を中心にまとめる。主要な暗号方式（NIST PQC 標準化への提案方式等）を記載するが、対象とする暗号方式は執筆担当者が選定する。

- 2021 年度第 2 回 PQC WG での調査内容の報告について
  各章の執筆担当者が 2021 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. 2021 年度第 2 回 PQC WG（1/28）での実施内容及び決定事項

- 調査内容の報告について
  各章の執筆担当者が 2021 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した。
- ガイドラインの案及び調査報告書の案の執筆方針について
  執筆方針が決定された。

3.4. 2022 年度第 1 回 PQC WG（9/26）での実施内容及び決定事項

- ガイドラインの案及び調査報告書の案の執筆に関する中間報告
  各章の執筆担当者が、ガイドラインの案及び調査報告書の案に執筆する内容の概要について節単位で説明することで、PQC WG において執筆内容を確認し、修正案等について議論した。
- 2022 年度第 2 回 PQC WG までのスケジュールについて
  2022 年度第 1 回 PQC WG での議論を基に、各章の執筆担当者及び担当事務局員の合意の取れた原稿を 2023 年 1 月 16 日を目処に完成させることが決定された。
  【各章の執筆担当者と担当事務局員】

	執筆担当者	担当事務局員
2 章：PQC の活用方法	伊藤委員	篠原
3 章：格子に基づく暗号技術	下山委員、安田雅哉委員	青野
4 章：符号に基づく暗号技術	草川委員	高安
5 章：多変多項式に基づく暗号技術	安田貴徳委員	篠原
6 章：同種写像に基づく暗号技術	高島委員	横山
7 章：ハッシュ関数に基づく署名技術	廣瀬委員	五十部、大東

3.5. 2022 年度第 2 回 PQC WG（1/30）での実施内容及び決定事項

- ガイドラインの案及び調査報告書の案の執筆に関する最終報告
  各章の執筆担当者が、ガイドラインの案及び調査報告書の案に執筆する内容の概要について節単位で説明することで、PQC WG において執筆内容を確認し、修正案等について議論した。
- ガイドライン及び調査報告書を公開するまでのスケジュール
  以下の項目が決定された。
  - ガイドライン及び調査報告書の発行日を 2023 年 3 月とする。
  - ガイドライン及び調査報告書を 2023 年 4 月に Web で公開する。
- ガイドラインの案の目次については資料 3-5 別紙を参照

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. 予測図の更新に関する説明文の修正

【現状】

〈今後の予測図の取扱い〉

(1) 予測図を従来通り、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当面の間更新していく。なお、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。

〈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、今後は、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。

【修正案】

＜今後の予測図の取り扱い＞

(1) ~~予測図を従来通り、~~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。~~なお、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、より現状に即した評価であり、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。~~

〈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、~~今後は、~~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。

(1)の修正について

・「予測図を」が述語と離れているので、近づけた。（→予測図を当面の間更新していく。）

・「従来通り」が修飾する語と離れているので、近づけた。(→従来どおり直線で引き)

・「なお、～」は、安全サイドに倒した評価の意味を補う意味で付けた文章であるので、予測図の取り扱いから外して、注として付ける。

・第一回WG時の指摘とおりに、「則した」及び「それらよりも」を修正した。

(2)の修正について

・「今後は、」はタイトルと意味が重複するので、削除した。

4.2. 2022 年度予測図の更新

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、2022 年 6 月・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 1、2)。

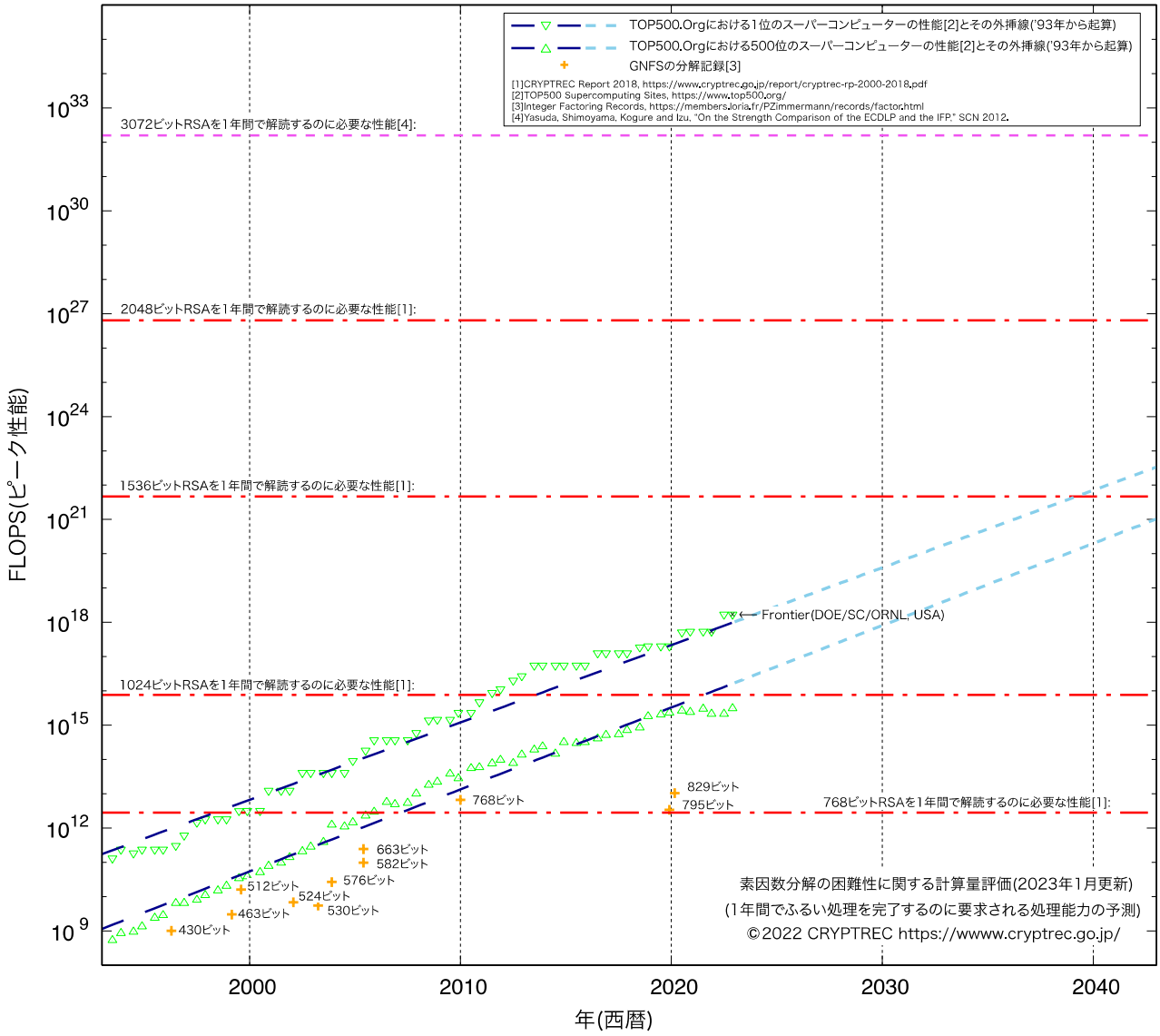


図 1：素因数分解の困難性に関する計算量評価(2023 年 1 月更新)

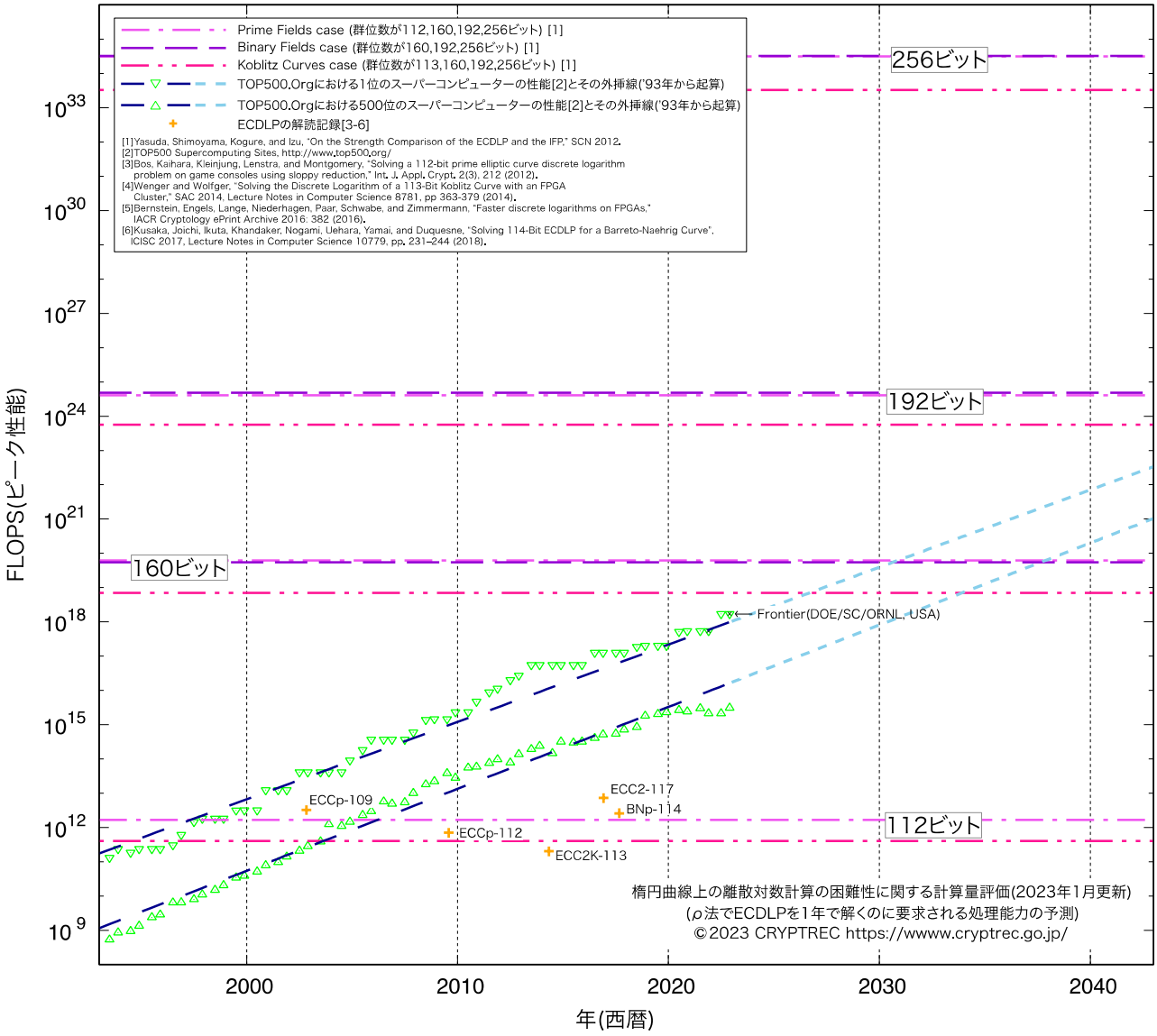


図 2：楕円曲線上の離散対数計算の困難性に関する計算量評価(2023 年 1 月更新)

以上

資料3－5

CRYPTREC 暗号技術ガイドライン
（耐量子計算機暗号）

CRYPTREC 暗号技術調査ワーキンググループ
（耐量子計算機暗号）

2023 年 3 月

目次

第 1 章

はじめに

現在広く使用されている公開鍵暗号方式として RSA 暗号と楕円曲線暗号が挙げられる．RSA 暗号と楕円曲線暗号が安全であるために，素因数分解問題や楕円曲線上の離散対数問題が計算量的に困難であることが必要であり，現在普及しているコンピュータでは効率的に解くことはできないと信じられている．ただし，量子コンピュータの開発が十分に進むと Shor のアルゴリズム [39, 40] により整数の素因数分解や離散対数を高速に計算できるため，RSA 暗号と楕円曲線暗号の安全性は大きく低下する．そのため，古典コンピュータ[*1]上で効率的な実装が可能であり，なおかつ古典・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公開鍵暗号方式が必要とされており，そのような暗号方式は耐量子計算機暗号 (Post-Quantum Cryptography: PQC) とよばれている．同様に，共通鍵暗号方式においても量子コンピュータによって安全性は低下することが知られているが [16]，公開鍵暗号に比べるとその影響は小さいと考えられている．その根拠として，電子政府推奨暗号の安全性を評価・監視し，暗号技術の適切な実装法・運用法を調査・検討するプロジェクトである CRYPTREC（図 1.1）による，量子コンピュータに対する共通鍵暗号の安全性の影響に関する 2019 年度の調査 [12] を挙げる．これらを踏まえ，本ガイドライン及び調査報告書において PQC は共通鍵暗号方式を含まず，公開鍵暗号方式を意味する．本ガイドライン及び調査報告書は PQC に関する内容をまとめたものである．

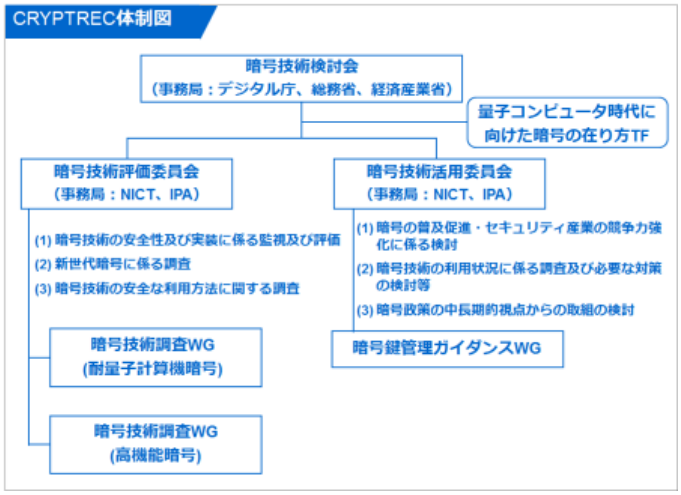


図 1.1: 2022 年度 CRYPTREC 体制図

[*1] 量子コンピュータに対して，現在普及しているコンピュータを古典コンピュータと呼ぶことにする．

近年の世界的な量子コンピュータの開発にともない，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実施されており，国内でも PQC の研究動向を把握する必要性が高まっている．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，2021 年度及び 2022 年度において本ガイドライン及び調査報告書を作成した．

本調査では 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，主に 2022 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．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．ガイドラインと調査報告書の 1 章は共通であり，一般的な読者にむけた内容としてまとめた．ガイドラインの第 2 章には暗号初学者向けに PQC の活用方法に関する内容，特に守秘・鍵共有・署名のための PQC の利用などについてを記載しており，この章は調査報告書には記載していない．ガイドラインの第 3 章以降，そして調査報告書の第 2 章以降で，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抜粋となっており，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 耐量子計算機暗号 (PQC) の必要性について

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 [49, 22]．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，トポロジカル量子計算，ホロノミック量子計算等に大別できる[*2]．上記分類とは別に，量子アニーリング (Quantum Annealing: QA) と呼ばれる計算フレームワークが存在する．[*3] これらのうち，超伝導，イオントラップによる量子回路型コンピュータおよび，超伝導磁束量子ビットによる量子アニーリングを行う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レベルにまで達し，非専門の技術者レベルでもクラウドを通じた利用が容易となったことから注目されている．そのため，冒頭に挙げた基本的操作の種類による分類の他に，応用目的の文脈から量子回路型と量子アニーリング型のみを取り上げ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対比されることもある [49, 第 2 章, p.11]．

超伝導を用いた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ここ数年で急速に進展しており，2019 年 10 月に Google が 53qubits [1]，2021 年 12 月には Rigetti が 80qubits [36]，2022 年 11 月には IBM が 433qubits のプロセッサ [20] を発表している．また，中国においても中国科学技術大学が 66qubits の祖沖之 2.1 [52] を，百度 (Baidu) が 2022 年 8 月に 10qubits コンピュータ [6] を発表している．日本でも富士通が 2023 年度中の 64qubits 量子コンピュータの公開を目指している [37]．超伝導方式以外の開発も進んでおり，イオントラップ式では 2020 年 10 月に IonQ が 32qubits [8]，2022 年 6 月には Quantinuum が 20qubits [35] を発表している．シリコン量子ビット式では Intel，日立製作所が開発を進めている [45]．

2020 年代終盤までのロードマップとして 10-100 万 qubits を搭載し，量子誤り訂正を組み込むことによりほぼノイズの無い量子回路型コンピュータの開発目標が Google, IBM 等から公表されている．なお，量子コンピュータの性能

[*2] この分類に関しては [22, 47] および [42, Sect 1.6] を参照.

[*3] 元々は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 Apolloni ら [2] によりこの名前が付けられたが，現在では断熱計算のモデル [3, 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ものと見做されているようである [34, § 3].

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みではなく，量子誤り訂正，量子ランダムアクセスメモリ等，2022 年現在では実用化されていない技術を用いる必要があり，それらの開発スピードの予測困難性が，量子コンピュータの暗号に与える影響を予測する際の難所となっている．

量子コンピュータによる現代暗号への脅威: Peter 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 [40] 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，実現時期の見積りに関する研究が進められている [15, 17, 43, 4, 28, 30, 31]．特に，現在標準的に用いられている，2048bits の合成数を公開鍵とする RSA 暗号 RSA-2048 の危殆化時期に関して様々な予測が存在する．学術的な研究に基づいたものでは 2039 年以降 [43]，2050 年前後 [4] 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．量子コンピューティングの専門家へのアンケートを元にした 2021 年時点での予測 [31] 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 50% 程度であると考える専門家が半数程度存在する．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 [32, p. (II-4)48,52] ではある程度コヒーレンス時間の長い数百 qubits 規模の量子回路コンピュータの登場は 2033 年頃としているため，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．一方で，セキュリティ分野の専門家の予測では，PQCrypto2014 の招待講演における Mariantoni の予測では 2029 年 [28]，Workshop on Cybersecurity in a Post-Quantum World (2015 年) における Mosca の予測では 2026-2031 年 [30] と若干早めの時期を想定している．

将来的に RSA 暗号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．量子回路型コンピュータ実機を用いた Shor のアルゴリズムの実験に関しては，CRYPTREC 外部調査報告書「Shor のアルゴリズム実装動向調査」[17] に挙げられているもの及びその後の [24, 44, 48] 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である．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，近年では IBM Quantum を用いてほぼ完全な回路を実装した実験報告 [5] や離散対数問題の実装実験報告 [4] が出版されるなど，実際に素因数分解できた合成数の大きさには表れない量子回路規模の拡大は着実に続いている．

Shor のアルゴリズムを用いない素因数分解の計算手法として，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として定式化するものがある．特に，量子アニーリングを用いた実験がこの 10 年間で多数報告されている．初期にはハミルトニアンに合わせて有機化合物を合成し NMR 測定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利用可能になって以降は実験報告が相次いでいる [50]．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，実機を用いた最も大きな実験 [23] では 19bits の合成数 376289=571×659 の分解に成功しているが，その要因は主に方程式内の変数の省略によるものである．また，同様の組み合わせ最適化問題を量子回路型コンピュータの上で Quantum Approximate Optimization Algorithm (QAOA) を用いて解く実験 [27](143，291311 を分解)，Digitized adiabatic quantum computation を用いて解く実験 [19](2479 を分解) も報告されている．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 [38] 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，[51] ではこれをさらに最適化問題に落とし込み，QAOA を 10qubits 回路上で実行することで 48bits の素因数分解実験を行ったという報告がされている．

PQC の必要性について: 上述のように，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．しかし，上に挙げた民間企業のロードマップが予定通りに達成された場合には，今後数十年で現代暗号の解読を行うのに十分な大きさ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可能性がある．そのような量子コンピュー

表 1.1: NIST 耐量子計算機暗号 Call for proposal[33] における安全性レベルと計算量の対応表

レベル	量子回路の最大深さ	古典論理ゲート数
レベル 1	2^{170}	2^{143}
レベル 2	–	2^{146}
レベル 3	2^{233}	2^{207}
レベル 4	–	2^{210}
レベル 5	2^{298}	2^{272}

タが出現した場合，守秘・鍵共有に用いられる現代の暗号方式の中で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 RSA 暗号・楕円曲線暗号が危殆化するリスクがある．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 PQC の研究及び標準化等に関する動向

PQC に関する研究成果は Crypto，Eurocrypt，Asiacrypt 等，暗号分野の国際会議で 1980 年代から議論されている．さらに PQC を専門に扱う国際会議として PQCrypto が挙げられ，その第 1 回会議は 2006 年に開催され，2022 年までに計 13 回開催されている．

PQC の標準化に関する近年の動向については，まず 2015 年 8 月，アメリカ国家安全保障局 (NSA) は PQC への将来的な移行計画を発表している．また，アメリカ国立標準技術研究所 (NIST) は 2016 年から PQC の公募を開始し，その締切である 2017 年 11 月 30 日までに 82 件の暗号方式が提案され，そのうち公募条件を満たした暗号方式は 69 件あり，その後 5 件の取り下げがあった．2019 年 1 月 30 日には，NIST から PQC の標準化の第 2 ラウンドへ進む方式として 26 件が発表された．2020 年 7 月 22 日には，NIST から PQC の標準化の第 3 ラウンドへ進む方式として，Finalist として 7 件，Alternate Candidates として 8 件が発表された．そして，2022 年 7 月 5 日には，NIST から標準化方式として公開鍵暗号方式 1 件と電子署名方式 3 件が発表された．同時に，第 4 ラウンドへ進む方式として，公開鍵暗号方式として 4 件が発表され，電子署名方式については再公募を行うこととした．欧州では ETSI が PQC の調査活動を行い [14]，ISO/IEC でも標準化に向けた議論が始まっている [21]．

NIST の標準化において，安全性をレベル 1 から 5 で定義しており，各暗号方式は提案パラメータとそれによって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秘密鍵探索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と同等かそれ以上の計算量とされている．ここで，公開鍵暗号では，適応的選択暗号文攻撃に対する識別不可能性（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，電子署名では，適応的選択文書攻撃に対する存在的偽造困難性（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．計算時間を制限するために，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最大の深さによって，古典コンピュータを利用可能な攻撃者に対しては古典論理ゲート数によってレベル 1 から 5 の安全性における計算量を評価しており，それぞれ表 1.1 のようになると見積もっている [33]．

CRYPTREC の暗号技術調査ワーキンググループにおいても 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 [10]．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 [11]．

1.3 本調査で対象とした PQC の種類

この節では本調査の対象として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, 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を選択した理由及びその説明に必要な事項について述べる．

代表的な公開鍵暗号方式は，その安全性が数学的な計算問題の困難性と関わりがある．例えば RSA 暗号では，二つの同程度の大きさでかつ異なる素数 p, q が秘密鍵，それらの積 $N = pq$ が公開鍵として使用される．N が素因数分解されると秘密鍵 p, q が計算されてしまい，RSA 暗号は解読されてしまう．楕円曲線暗号の場合も楕円曲線暗号の公開鍵から楕円曲線上の離散対数問題が定義され，それを解くことでその秘密鍵が計算できてしまう．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 暗号と同様に，それらの安全性はそれぞれで利用される数学的な計算問題の困難性と関わりがある．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が，それら 5 種類の暗号方式が PQC とされている理由である．（本調査の対象である暗号方式と数学的な計算問題の関係は各章の第 1 節で説明する．）

同種写像に基づく暗号技術を除く 4 種類の暗号技術の研究の歴史は長く，格子に基づく暗号技術は 20 年以上，符号に基づく暗号技術は 40 年以上，多変数多項式に基づく暗号技術は 30 年以上，ハッシュ関数に基づく署名技術は 40 年以上研究が行われている．従って，本調査ではこれらの暗号技術を有力な PQC として調査した．同種写像に基づく暗号技術は新しい暗号技術ではあるが，研究が近年活発に進められており，また，NIST の PQC に関する報告書 NISTIR 8105 において，同種写像に基づく暗号技術は代表的な PQC として扱われている．(上述の各暗号方式の歴的な事実については各章の第 4 節に記載した．) 以上の理由から，本調査ではこれら 5 種類の暗号技術を調査対象とした．

1.4　耐量子計算機暗号ガイドライン執筆者リスト

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	草川 恵太	日本電信電話株式会社
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	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	野島 良	国立研究開発法人情報通信研究機構
事務局	青野 良範	国立研究開発法人情報通信研究機構
事務局	五十部 孝典	国立研究開発法人情報通信研究機構
事務局	伊藤 竜馬	国立研究開発法人情報通信研究機構
事務局	大久保 美也子	国立研究開発法人情報通信研究機構
事務局	大東 俊博	国立研究開発法人情報通信研究機構
事務局	小川 一人	国立研究開発法人情報通信研究機構
事務局	金森 祥子	国立研究開発法人情報通信研究機構
事務局	黒川 貴司	国立研究開発法人情報通信研究機構
事務局	高安 敦	国立研究開発法人情報通信研究機構
事務局	横山 和弘	国立研究開発法人情報通信研究機構
事務局	吉田 真紀	国立研究開発法人情報通信研究機構
事務局	篠原 直行	国立研究開発法人情報通信研究機構

第 1 章の参照文献

[1] F. Arute, K. Arya, R. Babbush , D. Bacon, J. Bardin, R. Barends, R. Biswas, S. Boixo et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574, 505-510, 2019.

[2] D. de Falco, B. Apolloni, N. Cesa-Bianchi. A numerical implementation of "quantum annealing." *Proceedings Of the Ascona-Locarno conference*, 97-111, 1988.

[3] T. Albash, D. A. Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, vol. 90, Iss. 1, 015002, 2018.

[4] 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, 1-21, 2022.

[5] M. Amico, Z. H. Saleem, M. Kumph. Experimental study of Shor's factoring algorithm using the IBM Q Eexperience. *Physical Review A*, vol. 100, 012305, 2019.

[6] ACM NEWS. China's Baidu unveils 10-qubit quantum computer. `https://cacm.acm.org/news/264095-chinas-baidu-unveils-10-qubit-quantum-computer/fulltext`, 2022-08-26. （2023-01-23 閲覧）

[7] J. Buchmann, E. Dahmen, M. Szydlo. Hash-based Digital Signature Schemes. *Post-Quantum Cryptography*, 35-93, 2009.

[8] P. Chapman, Introducing the world's most powerful quantum computer. `https://ionq.com/posts/october-01-2020-introducing-most-powerful-quantum-computer`, 2020-10-01.（2023-01-23 閲覧）

[9] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, D. Smith-Tone. Report on Post-Quantum Cryptography. *NISTIR 8105*, `https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf`, 2016.

[10] CRYPTREC. 格子問題等の困難性に関する調査. *CRYPTREC Report 2014*, `https://www.cryptrec.go.jp/exreport/cryptrec-ex-2404-2014.pdf`, 2015.

[11] CRYPTREC. 耐量子計算機暗号の研究動向調査報告書. *CRYPTREC Report 2018*, `https://www.cryptrec.go.jp/report/cryptrec-tr-2001-2018.pdf`, 2018.

[12] CRYPTREC. 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 Report 2019*, `https://www.cryptrec.go.jp/exreport/cryptrec-ex-2901-2019.pdf`, 2019.

[13] CRYPTREC. Shor のアルゴリズム実装動向調査. *CRYPTREC Report 2020*, `https://www.cryptrec.go.jp/exreport/cryptrec-ex-3005-2020.pdf`, 2020.

[14] ETSI. Quantum-Safe Cryptography. `https://www.etsi.org/technologies-clusters/technologies/quantum-safe-cryptography`.（2023-03-15 閲覧）

[15] C. Gidney, M. Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5, 433, 2021.

[16] L. K. Grover. A fast quantum mechanical algorithm for database search. *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, 212-219, 1996.

[17] É. Gouzien, N. Sangouard. Factoring 2048-bit RSA integers in 177 days with 13 436 qubits and a multimode memory. *Physical Review Letters*, 127, 2021.

[18] A. Huelsing, D. Butin, S. Gazdag, J. Rijneveld, A. Mohaisen. XMSS: eXtended Merkle Signature Scheme. *RFC8391*, https://tools.ietf.org/html/rfc8391, 2018-05.（2023-01-23 閲覧）

[19] N. N. Hegade, K. Paul, F. Albarrán-Arriagada, X. Chen, E. Solano. Digitized adiabatic quantum factorization. *Physical Review A*, 104, 2021.

[20] IBM. IBM、400 量子ビット超えの量子プロセッサーと次世代 IBM Quantum System Two を発表. https://jp.newsroom.ibm.com/2022-11-10-IBM-Unveils-400-Qubit-Plus-Quantum-Processor-and-Next-Generation-IBM-Quantum-System-Two, 2022-11-10.（2023-01-23 閲覧）

[21] ISO. PQCRYPTO: Post-quantum cryptography for long-term security. https://www.iso.org/organization/5984715.html.

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

[23] S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble, S. Kais. Quantum Annealing for Prime Factorization *Scientific Reports*, vol. 8, 17667, 2018.

[24] E. G. Johansen, T. Simula. Prime number factorization using a spinor Bose-Einstein condensate inspired topological quantum computer. *Quantum Information Processing*, vol. 21, 31, 2022.

[25] J. Kelly. A preview of bristlecone, Google's new quantum processor. https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html, 2018-03-05.（2023-01-23 閲覧）

[26] N. Kunihiro. Quantum factoring algorithm: Resource estimation and survey of experiments. *International Symposium on Mathematics, Quantum Theory, and Cryptography. Mathematics for Industry*, vol. 33. 39-55, 2021.

[27] L. Qiu, M. Alam, A. Ash-Saki, S. Ghosh. Resiliency analysis and improvement of variational quantum factoring in superconducting qubit. *ISLPED '20: Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, 229-234, 2020.

[28] M. Mariantoni, Building a superconducting quantum computer. *PQCrypto 2014*, Invited talk, 2014-10-01.

[29] E. Martín-López, A. Laing, T. Lawson, R. Alvarez, X. -Q. Zhou, J. L. O'Brien. Experimental realization of Shor's quantum factoring algorithm using qubit recycling. *Nature Photonics*, vol. 6, num. 11, 773-776, 2012.

[30] M. Mosca. Cybersecurity in a quantum world: will we be ready? *Workshop on Cybersecurity in a Post-Quantum World*, https://csrc.nist.gov/csrc/media/events/workshop-on-cybersecurity-in-a-post-quantum-world/documents/presentations/session8-mosca-michele.pdf, 2015-04-03.（2023-01-23 閲覧）

[31] M. Mosca, M. Piani. 2021 Quantum threat timeline report: Global risk institute. https://globalriskinstitute.org/publication/2021-quantum-threat-timeline-report-global-risk-institute-global-risk-institute/, 2022-01-24.（2023-01-23 閲覧）

[32] 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.

[33] 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.

[34] 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. **数理解析研究所講究録**, 2059（量子システム推定の数理）, 13-23, 2017.

[35] Quantinuum. Quantinuum sets new record with highest ever quantum volume. https://www.quantinuum.com/news/quantinuum-completes-hardware-upgrade-achieves-20-fully-connected-qubits, 2022-06-14.（2023-01-23 閲覧）

[36] Rigetti. Rigetti Computing announces next-generation 40Q and 80Q quantum systems. https://investors.rigetti.com/news-releases/news-release-details/rigetti-computing-announces-next-generation-40q-and-80q-quantum, 2021-12-15.（2023-01-23 閲覧）

[37] 佐藤 信太郎. 量子コンピューティング：現状と産業化への課題 (量子技術の実用化推進 WG 第 3 回資料）. https://www8.cao.go.jp/cstp/ryoshigijutsu/jitsuyo_wg/3kai/siryo2-3.pdf, 2022-12-06.

[38] C. P. Schnorr. Fast factoring integers by SVP algorithms, corrected. *Cryptology ePrint Archive*, 2021/933, 2021.

[39] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *35th Annual Symposium on Foundations of Computer Science*, 124-134, 1994.

[40] P. W. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.*, volume 26, number 5, 1484-1509, 1997.

[41] 清水 俊也, 伊豆 哲也, 篠原 直行, 盛合 志帆, 國廣 昇. アニーリング計算による素因数分解について. *2019* 年 **暗号と情報セキュリティシンポジウム**, SCIS 2019, 2B4-3, 2019.

[42] S. P. Jordan. Quantum computation beyond the circuit model. *arXiv*, 0809.2307, 2008.

[43] J. Sevilla, C. J. Riedel. Forecasting timelines of quantum computing. *arXiv*, 2009.05045, 2020.

[44] U. Skosana, M. Tame. Demonstration of Shor's factoring algorithm for $N = 21$ on IBM quantum processors. *Scientific Reports*, 11, Article number 16599, 2021.

[45] 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, 2022-12-06.

[46] 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, 883-887, 2001.

[47] D. -S. Wang. A comparative study of universal quantum computing models: Toward a physical unification. *Quantum Engineering*, vol. 3, Iss. 4, e85, 2021.

[48] W. Wang, Z. You, S. Wang, Z. Tang, H. Ian. Computing Shor's algorithmic steps with classical light beams. *Scientific Reports*, vol. 12, Article number: 21157, 2022.

[49] 山本 俊. 量子情報技術（令和 3 年度　科学技術に関する調査プロジェクト）. **国立国会図書館調査及び立法考査局**, 2022.

[50] 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**. vol. 67, 290-296, 2022.

[51] B. Yan, Z. Tan, S. Wei, H. Jiang, W. Wang, H. Wang, L. Luo, Q. Duan et al. Factoring integers with

sublinear resources on a superconducting quantum processor. *arXiv*, 2212.12372, 2022.

[52] Q. Zhu, S. Cao, F. Chen, M. -C. Chen, X. Chen, T. -H. Chung, H. Deng, Y. Du et al. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Science Bulletin*, vol. 67, num. 3, 240-245, 2022.

第 2 章

PQC の活用方法

現在社会において，公開鍵暗号は様々な局面で利用されている．本節では，公開鍵暗号の幾つかの利用形態を紹介し，それらの利用形態における量子コンピュータによる脅威，及び脅威への対策を記載する．また，公開鍵暗号に対する脅威を検討する上では，暗号の利用形態のほかに，保護対象となるデータの保護期間も重要となる．そこで本章では，利用形態や保護対象を踏まえた対策について概説する．

2.1　暗号の利用形態

公開鍵暗号には幾つかの利用形態が存在するが，本書では「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」（以下，CRYPTREC 暗号リストと呼ぶ）に合わせ，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記載する．

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，その改竄を確認することができる．そのため，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，コンテンツの改竄を防止することができる．コンテンツはドキュメントデータであることもあれば，鍵情報[*1]を含むこともある．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等により）所定の人物等と紐づいている場合においては，コンテンツの生成者等を確認（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，署名者等の認証，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 [4] におけるクライアント認証やサーバ認証，OS のコードサイン等に広く利用されている．また，公開鍵の配布手段の 1 種である公開鍵暗号基盤（PKI）においても，電子証明書 [1] 等の形態で広く利用されている．加えて，タイムスタンプ署名 [2] では，コンテンツに対して署名が付与された時刻を確認するために利用されている．CRYPTREC 暗号リストには，DSA，ECDSA，RSA-PSS，RSASSA-PKCS1-v1_5 及び EdDSA が署名用途の公開鍵暗号として記載されている．

[*1] 鍵情報には暗号鍵やメタデータを含まれ [16]，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．

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解読することは困難となる．そのため，守秘用途の公開鍵暗号は，意図した相手にデータを提示する為に利用できる．暗号化処理による保護は，ドキュメントデータ等に対して行われることもあれば，別の暗号鍵の鍵情報[*2]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具体的な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，公開鍵暗号により共通鍵暗号の共通鍵を保護し，またその共通鍵を利用した共通鍵暗号によりコンテンツの秘匿性や完全生を保護するというアプローチが存在する．

守秘用途の公開鍵暗号を上記アプローチで用いる場合においては，共通鍵暗号の共通鍵は送信者により作成され，配送される．このアプローチにおいて仮に，ある時点で秘密鍵が漏洩した場合，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暗号の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できず，仮に送信者が別の通信相手との共通鍵暗号の共通鍵を使い回し等していても察知することができない．そのため，昨今の TLS 通信等における共通鍵の共有においては，秘匿用途の公開鍵暗号でなく，（次節の）鍵共有用途での公開鍵暗号を一時的（Ephemeral) な鍵を用いて利用することが望ましいと考えられるようになった [16, 7]．なお，「TLS 暗号設定ガイドライン」[16] においても，鍵交換（鍵共有・守秘）においては，Perfect Forward Security（PFS）の特性をもつ DHE（または ECDHE）を選択するの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[*3]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 [7]，4.1 節において（守秘用途である）RSA key tranport は利用すべきでないと記載されており，4.2 節でも一時的（Ephemeral) な鍵を用いる暗号スイートが推奨されている．

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 2 者が，同一の鍵情報[*4]を共有するために利用される．守秘用途の公開鍵暗号を 2 者間の通信で利用する場合は，任意のコンテンツに対する暗号処理を実施できる．一方，鍵共有用途の公開鍵暗号を 2 者間の通信で利用する場合，そこで共有されるものは，（共通鍵暗号や公開鍵暗号等の）何らかの暗号アルコリズムで利用するための暗号鍵となる．

近年利用されている 2 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とも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何らかの影響を与えることとなり，一方のみの計算で暗号鍵を導出することができない．そのため，守秘用途でのデータ送付と異なり，送信者が予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 各利用形態における課題

Michel Mosca [3] によると，量子コンピュータによる攻撃の脅威は，保護対象の暗号の保護期間を X 年，暗号処理の実装の置き換えに要する期間を Y 年，暗号解読に利用可能な量計算機が実現するまでの期間を Z 年としたときに，

[*2] 秘密鍵，共通鍵，鍵導出鍵，及びそれら鍵のメタデータを含む．

[*3] 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．

[*4] 共通鍵暗号の共通鍵，鍵導出機能の鍵やパラメータ等

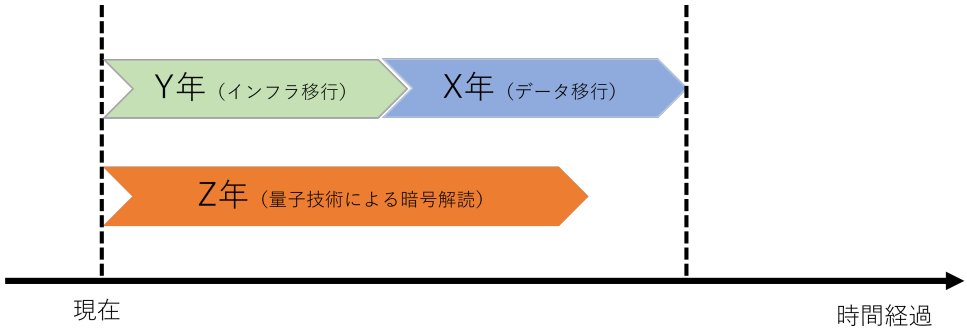


図 2.1: Mosca の発表 [3] より

これらが $X + Y > Z$ である場合には，心配する必要があるとしている（図 2.1）．もっとも，暗号解読が可能な量子コンピュータの実現時期は未だ不透明であり，Z を予想するのは難しい．また X は，暗号のみならず，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．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 を導出することが難しいこともありうる．加えて Y は，暗号の実装形態により大きく変化する．また，X 及び Y は，暗号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．

上記を踏まえ，ある公開鍵暗号アプリケーションが与えられた時に，その公開鍵暗号アプリケーションが量子コンピュータの脅威について備える必要性があるかを判断しようとした場合において，Z は不確定であり，X や Y も将来においては変動しうるため，判断が難しいという課題がある．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必要があり，場合によっては判断に必要な情報を収集するために相当の期間を要する可能性もある．

なお，多様な暗号ライブラリが，耐量子計算機暗号を相互運用可能な形態で受け入れるためには，共通の OID(Object Identifier) 体系を利用することが望ましい．この OID 体系については，IETF にて標準化が進んでおり [15, 14]，近い将来に解決されることが見込まれているため，以降は議論しない．

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 [4] 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その場限りとなり，それほど長期間に渡り検証されるわけではない．一方で，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りうる．特に，後者においては，仮に電子証明書に有効期間等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．このように，デジタル署名用途において，図 2.1 における X は大きく異なり，個別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 等を利用していた場合，トラストアンカーの置き換え等に時間を要することになり，図 2.1 における Y が 10 年以上となることも珍しくない．なお，e-mail を保護する S/MIME プロトコルにおいては，S/MIME 証明書（に対応する秘密鍵）によって保護される対象を通信プロトコルと捉えると（data in transit に対するセキュリティとなり）TLS 同様保護の対象となる期間は短いが，S/MIME 証明書（に対応する秘密鍵）によって保護される対象をメールコンテンツと捉えると（data at rest に対するセキュリティとなり）保護の対象期間が非常に長くなりうる．このように，X を導出する上では，個別のデータの保護期間のみでなく，プロトコルがサポートする機能も明確化する

ことが望ましい.

2.2.2 守秘用途での課題

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いことを想定していることや，場合によっては無期限で保護されることを想定していることがある．具体的には，患者を特定または類推可能な形態で保管された遺伝性の疾病に関する情報や，外交上重要な情報，及び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は無期限に秘匿することを想定しており，公開することを想定していない情報等も存在する．このような場合においては，図 2.1 における X は非常に大きくなるため，$X + Y > Z$ が成立すると仮定した上で，速やかに量子コンピュータによる攻撃に対する何らかの対策を行うことが望ましい.

ここへの脅威をより具体的に記載すると，無期限に保護しないといけないコンテンツを暗号化した暗号文が，パブリックな空間に保管されていた場合においては，攻撃者がその暗号文をあらかじめ保管しておき，将来に量子コンピュータが登場したのちに解読されるという脅威が含まれる．また，そのような暗号文が既にパブリックな空間に保管されている又は過去に保管されていた場合は，その暗号文が将来解読される脅威も含まれる.

2.2.3 鍵共有用途での課題

鍵共有用途での課題は，守秘用途の課題と同種の課題を含む．例えば，鍵共有で共有された共通鍵暗号の共通鍵が，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，図 2.1 における X は非常に大きくなり，$X + Y > Z$ が成立すると想定され，速やかに量子コンピュータによる攻撃に対する何らかの対策が必要となる.

加えて，守秘用途で存在しない新たな課題もある．例えば，一時的（Ephemeral) な鍵情報を用いた DH 鍵共有を採用した，PFS な情報システムが存在したとする．また，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．そして，その情報システムの DH 鍵共有処理部分を，標準化された耐量子計算機暗号を用いて置き換えることとする．この置き換え操作においては，以下の 2 つの方針が考えられる.

1) 鍵共有用途の耐量子計算暗号を利用して置き換える
2) 守秘用途の耐量子計算機暗号を利用して置き換える

標準化された鍵共有用途の公開鍵暗号が存在するのであれば，1) を選択可能であり，それは比較的容易に実現可能だと考えられる．しかしながら，標準化された鍵共有用途の公開鍵暗号が存在しなく，守秘用途の公開鍵暗号しか標準化されたものがないのであれば，2) を選択することとなり，守秘用途の耐量子計算機暗号を用いて鍵共有部分を構成することとなる．ここで，守秘用途の暗号を単純に導入した場合，PFS の性質を持たなくなる可能性があり，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可能性が生じる．また，守秘用途の公開鍵暗号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り得る．上記のような課題は，既存のシステムが鍵共有用途の公開鍵暗号を利用しており，今後採用する耐量子計算機暗号が守秘用途の場合に発生する課題となる.

2.3 各利用形態における対策

いずれの利用形態においても，量子コンピュータの脅威への対策を検討するにあたり，図 2.1 における X,Y,Z を意識することが重要となる．図 2.1 における X や Y は利用形態や暗号モジュール毎に異なることも想定され，それらの値の関係は将来において変動しうる．加えて Z は不確定であり，予想することが難しい．そのような状況で，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も予想され，結果として本当に保護が必要なデータに手が回らない可能性がある．そこで，優先度の高いものを割り出し，その優先度に応じて対応を行うことが適切だと考えられる．

例えば，NIST は耐量子計算機暗号への移行の一環として情報システムが利用している暗号の棚卸しや整理の方法を検討しており [10]，以下のような流れで対策を行うことを想定していると考えられる．

- ある程度以上の価値のある情報を扱う情報システムをリストアップする
- 上記情報システム内において利用される，暗号モジュール及び暗号方式をリストアップする
- 上記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）を把握する．保護期間が設定されていない場合は設定する
- 各暗号モジュールの移行に要する時間（Y）を把握する
- X 又は Y が一定の値以上で，またある程度以上の価値のある情報を扱う情報システムをリストアップする
- 上記リストに対してプライオリティ付けを行う
- プライオリティ順に，Cryptographic agility[6] の向上や暗号方式の耐量子計算機暗号への移行を実施する
- 耐量子計算機暗号への速やかな移行が難しい場合は，以下の公開鍵暗号の用途に応じた対策を行う

暗号モジュールや暗号方式のリストアップは，既に管理簿や仕様書等が存在するのであればそれを利用できる．管理簿や詳細な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，効率の面からもミスを減らす面からも望ましい．そのようなツールの利用を検討する上では，(2022 年末時点では最終報告は公開されていないものの，今後出てくるであろう)NIST NCCoE の検討 [10] が参考になるものと思われる．

保護期間の把握に際しては，過剰な期間が設定されているものに対しての保管期間の短縮，不要な情報の消去，公開可能な情報の公開等を合わせて行っても良い．そのような処理を行うことで X が減少することが期待でき，より効率的に対策を行えることが期待される．

暗号の移行に際しては，速やかに耐量子計算機暗号に移行するというアプローチと，予め Cryptographic agility[6][*5] を向上させつつ，ある程度以上の Cryptographic agility が達成された後に移行するというアプローチが存在する．Cryptographic agility が上昇した場合，Y が減少することとなる．そのため，例えば耐量子計算機暗号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ケースにおいては，当面の間は Cryptographic agility 上昇に努めるというアプローチが合理的であると考えられる [18]．なお，Cryptographic agility 上昇に伴う Y が減少により，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することもできる．

[*5] より迅速に暗号を変更可能とする性質.

2.3.1 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）が比較的長く，速やかな耐量子計算機暗号への移行が難しい可能性もある．このようなケースにおいても，以下のような対応を取ることが望ましい．

PKI においては一般に Y が大きくなる傾向にあるが，電子証明書の有効期間短縮や，1 枚の電子証明書に対して従来暗号と耐量子計算機暗号の 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，ある程度は Y も減少することが期待できる [11, 12]．なお，後者の 2 つの公開鍵暗と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性が大きく増加すことが考えられ，その点には注意を払う必要がある．

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挙げられる．例えば，ERS[5] 等を利用することで，タイムスタンプの更新や，暗号の更新を行うことができる．Z 年が経過する前に，既存の公開鍵暗号を耐量子計算機暗号に更新することが可能であれば，X,Y,Z の関係性によらず，データは保護される．一方で，実装の複雑性が増加する可能性があり，Y が上昇する可能性がある点は注意が必要となる．

2.3.2 秘匿及び鍵共有用途固有の対策

根本的な対策は，暗号を耐量子計算機暗号に移行することである．しかしながら，2.2.2 節及び 2.2.3 節に記載した通り，秘匿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を想定していることもある．そのような情報は既に将来における量子コンピュータでの解読リスクに晒されており，一刻も早く対応を始めない限り，リスクに晒された情報は増加し続ける．一方で，全ての秘匿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，全ての暗号文に対して迅速な移行を行うことには困難が想定される．

そのような状況においても，秘匿及び鍵共有用途固有の対策を効率的に行う方法 [18] として，以下のようなアプローチが考えられる．Z に対して $X + Y$ が非常に小さい（$X + Y << Z$）と予測される暗号文に対しては，CRYPTREC による注意喚起情報 [17] に注意を払いつつ，今まで通りの対応を行う．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 節前段で述べたような，耐量子計算機暗号への移行や，暗号文の保護期間である X 年の短縮と，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 年の短縮を行う（その結果 X や Y が十分に小さくすることができるのであれば，上記の今まで通りの対応を行う）．一方で，前述検討の上でも $X + Y > Z$ と予想される又はそうなることが避けられない暗号文に対してしては，暗号システムの対量子計算機暗号への移行を進めつつも，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予め検討することが望ましい．

2.3.3 耐量子計算機暗号の活用方法

耐量子計算機暗号を活用する上では，2.3 節冒頭で述べたとおり，現状において，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を予め把握しておくことで，より効率的な対応ができると考えられる [18]．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が望ましい．上記のような検討を予め行うことで，Cryptographic agility[6] の向上も見込むことができ，より効果的に耐量子計算機暗号を活用できることが期待できる．量子コンピュータ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と，図 2.1 における X,Y,Z の関係を踏まえて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.

第 2 章の参照文献

[1] D. Cooper, S. Santesson, S. Farrell, S, Boeyen, R. Housley, W. Polk. Internet $X.509$ Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile https://www.rfc-editor.org/rfc/rfc5280.

[2] C. Adams, P. Cain, D. Pinkas, R. Zuccherato. Internet $X.509$ Public Key Infrastructure Time-Stamp Protocol (TSP) https://www.rfc-editor.org/rfc/rfc3161.

[3] M. Mosca. Cybersecurity in a Quantum World: will we be ready? https://csrc.nist.gov/csrc/media/events/workshop-on-cybersecurity-in-a-post-quantum-world/documents/presentations/session8-mosca-michele.pdf.

[4] E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3 https://www.rfc-editor.org/rfc/rfc8446.

[5] T. Gondrom, R. Brandner, U. Pordesch. Evidence Record Syntax (ERS) https://www.rfc-editor.org/rfc/rfc4998.

[6] R. Housley. Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms https://www.rfc-editor.org/rfc/rfc7696.

[7] Y. Sheffer, R. Holz, P. Saint-Andre. Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) https://www.rfc-editor.org/rfc/rfc7525.

[8] 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.

[9] National Security Agency. The Commercial National Security Algorithm Suite 2.0 and Quantum Computing FAQ https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI_CNSA_2.0_FAQ_.PDF.

[10] W. Barker, M. Souppaya, W. Newhouse. MIGRATION TO POST-QUANTUM CRYPTOGRAPHY https://www.nccoe.nist.gov/sites/default/files/legacy-files/pqc-migration-project-description-final.pdf.

[11] D. Stebila, S. Fluhrer, S. Gueron. Hybrid key eXchange in TLS 1.3 https://datatracker.ietf.org/doc/html/draft-ietf-tls-hybrid-design.

[12] M. Punsworth, J. Gray. TComposite KEM For Use In Internet PKI https://datatracker.ietf.org/doc/html/draft-ounsworth-pq-composite-kem-00.html.

[13] A. Becker, R. Guthrie, M. Jenkins. Non-Composite Hybrid Authentication in PKIX and Applications to Internet Protocols https://www.ietf.org/id/draft-becker-guthrie-noncomposite-hybrid-auth-00.html.

[14] S. Turner, P. Kampanakis, J. Massimo, B. Westernbaan. Algorithm Identifiers for NIST's PQC Algorithms for Use in the Internet $X.509$ Public Key Infrastructure https://datatracker.ietf.org/doc/html/draft-

turner-lamps-nist-pqc-kem-certificates.

[15] J. Massimo, P Kampanakis, S. Turner, B. Westernbaan. Algorithms and Identifiers for Post-Quantum Algorithms https://datatracker.ietf.org/doc/draft-massimo-lamps-pq-sig-certificates.

[16] 独立行政法人情報処理推進機構. SSL/TLS 暗号設定ガイドライン https://www.ipa.go.jp/security/ipg/documents/ipa-cryptrec-gl-3001-3.0.1.pdf.

[17] Cryptography Research and Evaluation Committees. 注意喚起一覧 https://www.cryptrec.go.jp/er.html.

[18] T. Ito, M. Une, T. Seito. 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 https://www.imes.boj.or.jp/research/papers/japanese/19-J-15.pdf.

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

$$\begin{cases} 14s_1 + 15s_2 + 5s_3 + 2s_4 \approx 8 & (\mathrm{mod}\ 17) \\ 13s_1 + 14s_2 + 14s_3 + 6s_4 \approx 16 & (\mathrm{mod}\ 17) \\ 6s_1 + 10s_2 + 13s_3 + s_4 \approx 12 & (\mathrm{mod}\ 17) \\ \vdots & \\ 6s_1 + 7s_2 + 16s_3 + 2s_4 \approx 3 & (\mathrm{mod}\ 17) \end{cases}$$

が与えられたとする. (この数値例は [55] 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. 上記の数値例では $\mathbf{s} = (0, 13, 9, 11) \in \mathbb{Z}_{17}^4$ 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 3.1 (NTRU 問題 [39]) 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.

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 [58]. 2018 年に，篩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)[12] 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 [26]. （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，LWE チャレンジにおいては，$(n, \alpha) = (45, 0.030), (50, 0.025), (55, 0.020), (90, 0.005)$ 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, 7 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，$(n, \alpha) = (50, 0.025)$ 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約 592 時間で求解されたと報告されている：

- HardwareCPU : AMD EPYC$^{\text{TM}}$7002 Series 128@2.6GHz
- RAM : 1.5TB
- GPU : 8 ∗ NVIDIA®GeForce®RTX 3090
- VRAM : 8 ∗ 24GB（936.2 GB/s）

3.2 代表的な格子に基づく暗号方式の説明

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 [35]. 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 [2] 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 $(\sigma - u)A = 0$ である場合に検証を受理する.

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 [33] 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) を法とする冪乗剰余演算 $g(x) = g^x \mod n$ 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 [33]. 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 の冪乗 (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．$h_{\hat{a}}(\hat{z}) = \hat{a} \cdot \hat{z} = a_1 z_1 + ... + a_m z_m \in R$.

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 $c = hash(h_{\hat{a}}(\hat{z}) - Sc, M)$ が成り立つ場合に検証を受理する．

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

3.3.1　CRYSTALS-Kyber

歴史: CRYSTALS-Kyber は NICT PQC 公募への応募方式のひとつとして 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 [5], 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 [14]．NIST PQC 標準化の Round 3 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 [49].

NIST PQC 標準化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[7] 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[*1]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 [37] 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 3.2，3.3，3.4 に Lindner-Peikert[44] 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．

[*1] 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 [45] 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 $n = 256, q = 3329 = 13 \cdot 256 + 1$ と固定されている[*2]．
- 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 $(\boldsymbol{u}, v)$ 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，そのビット反転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 から十分な長さの疑似乱数列を生成し，それを Parse 関数により R_q の元に変換する．
- $\mathsf{CBD}_\eta(\mathsf{PRF}(\sigma, i))$: 疑似乱数生成器 PRF は長さ 32Bytes の σ と 1Byte の i をシードとして 512ηbits の疑似乱数列 $\beta_0\beta_1\cdots\beta_{512\eta-1}$ 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．

疑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疑似乱数生成器 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 Round 2 に合わせてアップデートされたバージョン 2.0[6] 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 3.1 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，AES-256 の CTR モードを 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，

[*2] Round 1 提出時には $q = 7681$ であったが，Round 2 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[*3] を参照した．

表 3.1: CRYSTALS-Kyber における疑似乱数生成器の実装 [7, 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 3.2: 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 [44, Sect. 3.1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	CRYSTALS-Kyber [7, Algorithm 4] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
0:		$d \xleftarrow{\$} \mathcal{B}^{32}$ $(\rho, \sigma) \leftarrow \mathrm{G}(d)$
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 3.2 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 の疑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数論変換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見做される．

$\boldsymbol{s}, \boldsymbol{e} \in R_q^k$ 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 3.3 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 [7, Sect. 1.5]．

[*3] https://github.com/pq-crystals/kyber/blob/master/ref/symmetric-shake.c

表 3.3: 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[44, Sect. 3.1] $\mathsf{Enc}(pk=(A,B),\boldsymbol{m}\in\{0,1\}^{\ell})\to\mathsf{ct}$	CRYSTALS-Kyber [7, 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	$\boldsymbol{r}[i]\leftarrow\mathsf{CBD}_{\eta_1}(\mathsf{PRF}(r,i))$ 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:	$\boldsymbol{u}=\boldsymbol{s}'A+\boldsymbol{e}'$	$\hat{\boldsymbol{r}}\leftarrow\mathsf{NTT}(\boldsymbol{r})$ $\boldsymbol{u}\leftarrow\mathsf{NTT}^{-1}(\hat{\boldsymbol{A}^T}\circ\hat{\boldsymbol{r}})+\boldsymbol{e}_1$ $v\leftarrow\mathsf{NTT}^{-1}(\hat{\boldsymbol{t}}^T\circ\hat{\boldsymbol{r}})+e_2+\mathsf{Decompress}_q(\mathsf{Decode}_1(m),1)$ $c_1\leftarrow\mathsf{Encode}_{d_u}(\mathsf{Compress}_q(\boldsymbol{u},d_u))$ $c_2\leftarrow\mathsf{Encode}_{d_v}(\mathsf{Compress}_q(v,d_v))$
return	$\mathsf{ct}=(\boldsymbol{u},\boldsymbol{v})$	$\mathsf{ct}=(c_1\|\|c_2)$

表 3.4: 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[44, Sect. 3.1] $\mathsf{Dec}(sk,\mathsf{ct})\to\boldsymbol{m}'$	CRYSTALS-Kyber [7, 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'

CRYSTALS-Kyber の復号関数（表 3.4 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 [25] 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．

以下の表で δ は CCA2-KEM に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 [7, Algorithm 9] で非受理となる確率である．

表 3.5: CRYSTALS-Kyber のパラメータ [7, Table 1] および [3, 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	1632	32	768	2^{-139}
$(256, 3, 3329)$	$(2, 2)$	$(10, 4)$	レベル 3	1184	2400	32	1088	2^{-164}
$(256, 4, 3329)$	$(2, 2)$	$(11, 5)$	レベル 5	1568	3168	32	1568	2^{-174}

3.3.2 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 [22] され，その後論文内での予告通りに 2017 年 11 月に NIST PQC 公募への応募方式 [20] 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 [21] が存在する．

NIST PQC 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[15] 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[41] 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 [42].

アルゴリズムの詳細: 表 3.6，3.7，3.9 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 [15, Fig. 1] および実装のための疑似コード [15, 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．

- $\mathsf{NTT}(\mathsf{a})$ 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，$r = 1753$, $r_i = r^{brv(128+i)} \bmod q$, $brv(k)$ 関数は k を 8bits の 2 進数としてみたときのビット反転．[15, 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．
- $\mathsf{ExpandA}(\rho)$: 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．
- $\mathsf{ExpandS}(\rho')$: 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．$\mathbb{Z}_q$ の多項式 $t \in R_q$, 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 $\mathsf{Power2Round}_q(t_{j,i}, d) \to (t_{j,i,1}, t_{j,i,0})$ とすれば，$\mathsf{Power2Round}_q(\boldsymbol{t}, d) \to (\boldsymbol{t}_1, \boldsymbol{t}_0)$ は $\boldsymbol{t}_1 = (\sum q_{j,i} x^i)_{j=1,\ldots,k}$, $\boldsymbol{t}_0 = (\sum r_{j,i} x^i)_{j=1,\ldots,k}$, ただし $t_{j,i} = q_{j,i} \cdot 2^d + r_{j,i}$ と定義したものである．
  また，α を $q-1$ の約数としたとき，上記と同様に整数 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．復元成功の十分条件は [15, Lemma 4] で与えられている．
- $\mathsf{SampleInBall}(\tilde{c})$ 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 $\boldsymbol{h} = \sum_{i=0}^{255} h_i$ の中で $h_i = 1$ となる項の数を表す．

表 3.6: CRYSTALS-Dilithium における鍵生成関数の比較

	格子ベース署名 [41, Fig. 4] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	CRYSTALS-Dilithium テンプレート [15, Fig. 1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	CRYSTALS-Dilithium 実装のための疑似コード [15, 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	$sk = (a, \hat{\boldsymbol{s}}), pk = (a, \boldsymbol{t})$	$sk = (A, \boldsymbol{t}, s_1, s_2)$, $pk = (A, \boldsymbol{t})$	$sk = (\rho, K, tr, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0)$, $pk = (\rho, \boldsymbol{t}_1)$

表 3.6 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 $S_\eta = \{\boldsymbol{w} \in R : \|\boldsymbol{w}\|_\infty \leq \eta\}$ から一様ランダムにサンプリングされる．

Fiat-Shamir 型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 3.7 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 3.7 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 3.8 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，$-c\boldsymbol{s}_2 + c\boldsymbol{t}_0$ 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 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

表 3.7: CRYSTALS-Dilithium における署名生成関数の比較

	格子ベース署名 [41, Fig. 4] $\mathsf{Sign}(sk=(a,\hat{\boldsymbol{s}}),$ $\mu\in\{0,1\}^*)\to\sigma$	CRYSTALS-Dilithium テンプレート [15, Fig. 1] $\mathsf{Sign}(sk=(A,t,\boldsymbol{s}_1,\boldsymbol{s}_2),$ $\mu\in\{0,1\}^*)\to\sigma$	CRYSTALS-Dilithium 実装のための疑似コード [15, 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: 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** $(\boldsymbol{z},\boldsymbol{h})\leftarrow\perp$

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 問題の変種ではあるが，タイトではないものの帰着が知られているため，Module-SIS 問題を安全性の根拠と考えることもできる．

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 3.10 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 3.9: CRYSTALS-Dilithium における署名検証関数の比較

	格子ベース署名 [41, Fig. 4]	CRYSTALS-Dilithium テンプレート [15, Fig. 1]	CRYSTALS-Dilithium 実装のための疑似コード [15, 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** $\perp$	$\boldsymbol{w}_1'=\mathsf{HighBits}(A\boldsymbol{z}-c\boldsymbol{t},2\gamma_2)$ **if** $\|\boldsymbol{z}\|_\infty<\gamma_1-\beta$ AND $c=H(M\|\|\boldsymbol{w}_1')$ **then** accept **else** $\perp$	$\boldsymbol{w}_1'\leftarrow\mathsf{UseHint}_q(\boldsymbol{h},A\boldsymbol{z}-c\boldsymbol{t}_1\cdot 2^d,2\gamma_2)$ **if** $\|\boldsymbol{z}\|_\infty<\gamma_1-\beta$ AND $\tilde{c}=H(\mu\|\|\boldsymbol{w}_1')$ AND $\#_1\boldsymbol{h}\le\omega$ **then** accept **else** $\perp$

表 3.10: CRYSTALS-Dilithium 署名方式のパラメータ [15, Table 1] , [3, 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	1312	2528	2420
$(256,6,5,8380417)$	$(4,2^{19},261888,196,49,13)$	レベル 3	1952	4000	3293
$(256,8,7,8380417)$	$(2,2^{19},261888,120,60,13)$	レベル 5	2592	4864	4595

注： 秘密鍵サイズは仕様書 [15] 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 [3] を参照した．

変種: [15, Table 3] には NIST の提唱する Category1 よりも弱いパラメータ，5 よりも強いパラメータが掲載されている．

その他補足情報: NIST PQC の第 3 ラウンド報告書において署名方式 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．[3, p.19].

ハッシュ関数 H の長さについて，V3.0 までは 384bits であったが衝突耐性を考えれると 256bits セキュリティを持たず，Category 5 の安全性を持たないことが判明したため最新の V3.1 では 512bits に修正されている [56, p.5].

3.3.3 FALCON

歴史: FALCON は 2017 年 11 月の NIST PQC 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 [31]．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[32] 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 [35] 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 の提案方式では Category 1,5 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 3.11，3.12，3.13 に，Gentry ら [35] 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 $\mathbb{Z}_q/\phi$ 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 $(f(\zeta_k))_{k=0,\dots,n-1}$ で定義する．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．
- $\mathsf{HashToPoint}(\mathsf{str}, q, n)$: 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．

表 3.11: 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 [35, Sect. 7.1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	FALCON[32, 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

$$fG - gF = q \bmod \phi \tag{3.1}$$

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 フレークワーク [35] における行列 A, B と捉えると，

$$A = \begin{bmatrix} 1 \\ h \end{bmatrix},\ B = \begin{bmatrix} g & -f \\ G & -F \end{bmatrix} \tag{3.2}$$

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[23] 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

$$\mathsf{FFT}(B) = \begin{bmatrix} \mathsf{FFT}(g) & \mathsf{FFT}(-f) \\ \mathsf{FFT}(G) & \mathsf{FFT}(-F) \end{bmatrix} \tag{3.3}$$

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[*4] 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 3.12: 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 [35, Sect. 7.1] $\mathsf{Sign}(sk = (\hat{B}, T), m \in \{0,1\}^*) \to \sigma$	FALCON[32, 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 3.12 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 (3.1), (3.2) 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 [32, 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

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

[*4] $B^{\star}$ は体 $\mathbb{Q}[x]/\phi$ におけるエルミート共役．詳細は [32, p.23]

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 $\|\hat{\boldsymbol{s}}\|^2 \leq \lfloor \beta^2 \rfloor$ を満たしていれば invFFT により通常空間の表現に戻し，Compress 関数を用いて署名文字列 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 3.13: 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 [35, Sect. 7.1]	FALCON[32, 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** $(s_2 = \perp)$
		retrun $\perp$
		$s_1 \leftarrow c - s_2 h \bmod q$
		if $\|(s_1, s_2)\|^2 \leq \lfloor \beta^2 \rfloor$
		retrun accept
		else
		retrun $\perp$

表 3.13 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，$\|(s_1, s_2)\|^2 \leq \lfloor \beta^2 \rfloor$ 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，上記の 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 3.14: FALCON のパラメータ [32, Table 3.3], [3, 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 [*5]	署名サイズ
$(\ 512, 12289, 165.736617183, 34034726)$	レベル 1	897	7553	666
$(1024, 12289, 168.388571447, 70265242)$	レベル 5	1793	13953	1280

[*5] 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 [3, Sect. D] を参照した．

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 [30], [19], [27]．特筆すべき点として，格子ベース署名 SOLMAE[40] が韓国の耐量子計算機暗号公募 KpqC へと提出されている．

その他補足情報: 格子ベースの Hash-and-sign 署名においてエラーベクトルの圧縮表現などを用いて署名長を短くするテクニックが提案 [28] されており，標準化の際にはこれを盛り込んだ方式に修正される可能性があったが，2022 年 11 月に開催された第 4 回標準化会議におけるアップデート報告 [53] では別の削減方法を用いることが発表された．

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 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 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た．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 [48]．NIST PQC の最終ラウンドまでの選定に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 [29]，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 [34]．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[13] および CECPQ2[18] にそれぞれ NewHope の USENIX 発表バージョン [11] 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 [43]．DNS サーバの一種である PowerDNS において，耐量子機能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 [36]．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 [38]，WolfSSL への CRYSTALS-Kyber，FALCON の実装 [60]，OpenSSH への Streamlined NTRU Prime の実装 [51] 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 [52]．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．

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 (Learning With Rounding) 問題, BDD (Bounded Distance Decoding) 問題, SIS (Small Integer Solution) 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．

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[1] 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[9] 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 [50] 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 [39][*6] 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 [57] 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[4] 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.

2005 年に Regev[54] 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. ひとつは問題の 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 [17].

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 [45, 46] により Ring-LWE 問題が, 2015 年には Langlois ら [47] 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 [10].

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 [24] により示された. この方向性の研究として, Bai ら [16] により提案された, Rényi エントロピーを用いた, 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議論するものがある.

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[58] が開催されている. 第 3.1 節に, 2022 年 8 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.

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.

[*6] 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] M. Ajtai. Generating hard instances of lattice problems. *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pp. 99-108, 1996.

[2] M. Ajtai. Generating Hard Instances of the Short Basis Problem. *Proceedings of ICALP'99*, Springer LNCS vol.1644, pp.1–9, 1999.

[3] G. Alagic, D. Apon, D. Cooper, Q. Dang, J. Kelsey, J. Lichtinger, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. `https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413.pdf`.

[4] M. Albrecht, S. Bai, L. Ducas. A subfield lattice attack on overstretched NTRU assumptions. *Advances in Cryptology – CRYPTO 2016 – 36th Annual International Cryptology Conference*, Part I, Springer LNCS vol. 9814, pp. 153-178, 2016.

[5] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Kyber – Submission to the NIST post-quantum project. `https://pq-crystals.org/kyber/data/kyber-specification.pdf`, 2023-02-21 閲覧.

[6] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Kyber – Submission to the NIST post-quantum project. `https://pq-crystals.org/kyber/data/kyber-specification-round2.pdf`, 2023-02-21 閲覧.

[7] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Kyber – Submission to the NIST post-quantum project. `https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf`, 2023-02-21 閲覧.

[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!(2018). *Security and Cryptography for Networks – 11th International Conference*, Springer LNCS vol. 11035, pp. 351-367, 2018. `https://estimate-all-the-lwe-ntru-schemes.github.io/docs/`.

[9] M. Ajtai, C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, pp. 284-293, 1997.

[10] M. Albrecht, A. Deo. Large modulus Ring-LWE $\geq$ Module-LWE. *Advances in Cryptology – ASIACRYPT 2017 – 23rd International Conference on the Theory and Applications of Cryptology and Information Security*, Part I, Springer LNCS vol. 10624, pp. 267-296, 2017.

[11] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe. Post-Quantum key exchange: A new hope. *Proceedings of the 25th USENIX Conference on Security Symposium*. SEC'16. Austin, USENIX Association, pp. 327-343,

2016.

[12] M. R. Albrecht, L. Ducas, G. Herold, E. Kirshanova, E. W. Postlethwaite, M. Stevens. The general sieve kernel and new records in lattice reduction. *Advances in Cryptology – EUROCRYPT 2019*, Springer LNCS vol. 11477, pp. 717–746, 2019.

[13] M. Braithwaite. Experimenting with Post-Quantum Cryptography. `https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html`, 2023-02-21 閲覧.

[14] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS – Kyber: A CCA-secure module-lattice-based KEM. *IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 353-367, 2018.

[15] S. Bai, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Dilithium algorithm specifications and supporting documentation (February 8, 2021). `https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf`.

[16] S. Bai, T. Lepoint, A. R.-Langlois, A. Sakzad, D. Stehlé, R. Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. *Journal of Cryptology*, vol. 31, Iss. 2, pp. 610-640, 2018.

[17] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, D. Stehlé,. Classical hardness of learning with errors. *Symposium on Theory of Computing Conference, STOC'13*, pp. 575-584, 2013.

[18] The Chromium Projects. CECPQ2. `https://www.chromium.org/cecpq2/`, 2023-02-21 閲覧.

[19] C. Chuengsatiansup, T. Prest, D. Stehlé, A. Wallet, K. Xagawa. ModFalcon: compact signatures based on Module-NTRU lattices. *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, ASIA CCS ' 20, pp. 853-866, 2020.

[20] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Dilithium algorithm specifications and supporting documentation (November 30, 2017). `https://pq-crystals.org/dilithium/data/dilithium-specification.pdf`.

[21] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS–Dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, Iss. 1, pp. 238-268, 2018.

[22] L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS–Dilithium: Digital signatures from module lattices. *Cryptology ePrint Archive*, 2017/633, 2017.

[23] L. Ducas, T. Prest. Fast Fourier Orthogonalization. *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC ' 16, pp. 191-198, 2016.

[24] N. Döttling, J. M.-Quade. Lossy codes and a new variant of the learning-with-errors problem. *Advances in Cryptology – EUROCRYPT 2013 – 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer LNCS vol. 7881, pp. 18-34, 2013.

[25] L. Ducas, J. Schanck. `https://github.com/pq-crystals/security-estimates`, 2023-02-21 閲覧.

[26] L. Ducas, M. Stevens, W. v. Woerden. Advanced lattice sieving on GPUs, with tensor cores. *Advances in Cryptology – EUROCRYPT 2021*, Springer LNCS vol. 12697, pp. 249–279, 2021.

[27] 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. *Advances in Cryptology – EUROCRYPT 2022*, Springer LNCS, vol. 13277, pp. 222-253, 2022.

[28] T. Espitau, M. Tibouchi, A. Wallet, Y. Yu. Shorter hash-and-sign lattice-based signatures. *Advances in Cryptology – CRYPTO 2022 - 42nd Annual International Cryptology Conference*, Springer LNCS 13508, 245–275, 2022.

[29] Federal Office for Information Security. BSI – Technical Guideline (Cryptographic Mechanisms: Recommendations and Key Lengths). *Version 2023-01*, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile&v=10, 2023-02-21 閲覧.

[30] P.-A. Fouque, F. Gérard, M. Rossi, Y. Yu. Zalcon: an alternative FPA-free NTRU sampler for Falcon. *Third PQC Standardization Conference*, https://csrc.nist.gov/Presentations/2021/zalcon-an-alternative-fpa-free-ntru-sampler-for-fa, 2023-02-21 閲覧.

[31] P. -A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, Z. Zhang. 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

[32] P. -A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, Z. Zhang. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Specification v1.2 – 01/10/2020. https://falcon-sign.info/falcon.pdf.

[33] A. Fiat, A. Shamir. How to prove yourself: Practical solutions to identiation and signature problems. *Advances in Cryptology – CRYPTO '86*, Springer LNCS vol. 186, pp. 186-194, 1986.

[34] General Intelligence and Security Service. Prepare for the threat of quantumcomputers. https://english.aivd.nl/publications/publications/2022/01/18/prepare-for-the-threat-of-quantumcomputers, 2023-02-21 閲覧.

[35] C. Gentry, C. Peikert, V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. STOC ' 08, pp. 197- 206, 2008.

[36] M. Grillere, P. Thomassen, N. Wisiol, FALCON-512 in PowerDNS. Website: https://blog.powerdns.com/2022/04/07/falcon-512-in-powerdns/.

[37] D. Hofheinz, K. Hövelmanns, E. Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. *Theory of Cryptography – TCC 2017*, LNCS,vol. 10677, pp. 341-371, 2017.

[38] A. Hülsing, K. -C. Ning, P. Schwabe, F. Weber, P. R. Zimmermann. Post-quantum WireGuard. *IEEE Symposium on Security and Privacy (SP)*, pp. 304-321, 2021.

[39] J. Hoffstein, J. Pipher, J. H. Silverman. NTRU: A ring-based public key cryptosystem. *Algorithmic Number Theory, 3rd International Symposium, ANTS-III*, Springer LNCS vol. 1423, pp. 267-288, 1998.

[40] K. Kim, M. Tibouchi, A. Wallet, T. Espitau, A. Takahashi, Y. Yu, S. Guilley. https://kpqc.or.kr/images/pdf/SOLMAE.pdf, KpqC Competition Round 1.

[41] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. *Advances in Cryptology – ASIACRYPT 2009*. LNCS vol. 5912, pp. 319-339, 2009.

[42] V. Lyubashevsky. CRYSTALS-Dilithium Round 3 Presentation. *Third PQC Standardization Conference*, 2021-06-07, https://csrc.nist.gov/Presentations/2021/crystals-dilithium-round-3-presentation, 2023-02-21 閲覧.

[43] M. Lantz, M. Hill, World' s first quantum computing safe tape drive. https://www.ibm.com/blogs/research/2019/08/crystals/, 2019-08-23, 2023-02-21 閲覧.

[44] R. Lindner, C. Peikert. Better Key sizes (and attacks) for LWE-Based encryption. *Topics in Cryptology – CT-RSA 2011 – The Cryptographers' Track at the RSA Conference 2011*, Springer LNCS vol. 6558, pp. 319-339.

[45] V. Lyubashevsky, C. Peikert, O. Regev. On ideal lattices and learning with errors over rings. *Advances in Cryptology – EUROCRYPT 2010 – 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, LNCS vol. 6110, pp. 1-23, 2010.

[46] V. Lyubashevsky, C. Peikert, O. Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM*, Vol. 60, Iss. 6, pp. 1-35, 2013.

[47] A. Langlois, D. Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, Vol. 75, Iss. 3, pp. 565-599, 2015.

[48] 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, 2022-09-07.

[49] National Institute of Standards and Technology. Selected Algorithms 2022. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022, 2023-02-21 閲覧.

[50] P. Q. Nguyen, J. Stern. Cryptanalysis of the Ajtai-Dwork cryptosystem. *Advances in Cryptology – CRYPTO '98, 18th Annual International Cryptology Conference*, LNCS vol. 1462, pp. 223-242, 1998.

[51] OpenSSH. OpenSSH 8.9 was released on 2022-02-23. https://www.openssh.com/txt/release-8.9, 2023-02-21 閲覧.

[52] Open Quantum Safe. Algorithms in liboqs. https://openquantumsafe.org/liboqs/algorithms/, 2023-02-21 閲覧.

[53] T. Prest. FALCON Update. *Fourth PQC Standardization Conference*, 2022-11-29, https://csrc.nist.gov/csrc/media/Presentations/2022/falcon-update/images-media/session-1-prest-falcon-pqc2022.pdf, 2023-02-21 閲覧.

[54] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp. 84-93.

[55] O. Regev. The learning with errors problem (invited survey). *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010*, pp. 191-204, 2010.

[56] P. Schwabe. 6 years of NIST PQC, looking back and ahead. *The 13th International Conference on Post-Quantum Cryptography – PQCRYPTO 2022*, Invited talk, 2022.

[57] D. Stehlé, R. Steinfeld,. Making NTRU as secure as worst-case problems over ideal lattices. *Advances in Cryptology – EUROCRYPT 2011 – 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer LNCS vol. 6632, pp. 27-47, 2011.

[58] TU Darmstadt UC San Diego. LWE Challenge. https://www.latticechallenge.org/lwe_challenge/challenge.php

[59] wolfSSL. ポスト量子暗号 Falcon 署名方式を追加しました, https://www.wolfssl.jp/wolfblog/2021/12/21/integration-falcon-signature-scheme-wolfssl/, 2023-02-21 閲覧.

[60] wolfSSL. wolfSSL support for Apache httpd and curl (Post-Quantum Edition). https://github.com/wolfSSL/osp/blob/master/apache-httpd/README_post_quantum.md, 2023-02-21 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 LPN 問題やシンドローム復号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 の冪とする．すなわち, ある正整数 k が存在して $q = p^k$ である．

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.

- 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, 非ゼロの係数の数である．すなわち, $\mathsf{HW}(\boldsymbol{v}) = \#\{v_i \mid v_i \neq 0\}$ 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: 自然数 n および 素数冪 q について, $\mathbb{F}_q$ 上の n 次元ベクトル空間の部分空間を $\mathbb{F}_q$ 上の線形符号と呼び, $\mathcal{C}$ で表す．n を符号長と呼ぶ．$\mathbb{F}_q$ 上の線形符号 $\mathcal{C}$ の符号語 $\boldsymbol{c}$ は, 一次独立な k 個の符号語 $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K$ の $\mathbb{F}_q$ 係数一次結合で表すことができる．このとき $[\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k]$ を基底と呼ぶ．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 $\boldsymbol{c} \cdot \boldsymbol{H}^\top = \boldsymbol{0}$ となるものである．$\boldsymbol{H}$ の行が線形独立であれば, $r = n - k$ である.

$[n, k]_q$ -線形符号 $\mathcal{C}$ の最小距離 d とは, 符号 $\mathcal{C}$ の非ゼロ符号語の中で最小のハミング重みをもつ符号語のハミング重みのことを言う．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 $\boldsymbol{G}$ が決まっている場合, メッセージ $\boldsymbol{s} \in \mathbb{F}_q^k$ を 符号語 $\boldsymbol{sG}$ と一対一対応させることができる．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, 符号語に加えられた誤りを訂正することができる．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．受信者は符号の復号アルゴリズムを用いて復号を行い, $\boldsymbol{y}$ 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, $t \leq \lfloor (d-1)/2 \rfloor$ であることが分かる．復号アルゴリズムはパリティ検査行列を用いることもある．その際に, $\boldsymbol{t} = \boldsymbol{y}\boldsymbol{H}^\top$ を計算し, これをシンドロームと呼ぶ.

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」 [17] 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 Learning Parity with Noise （LPN） 問題や符号に関連する問題の困難性について調査結果を述べる.

4.1.1 LPN 問題とは

LPN 問題とは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. 1993 年に, Blum, Furst, Kearns, Lipton [5] 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.

■ LPN 問題: $\mathbb{F}_2$ 上の分布 χ および $\boldsymbol{s} \in \mathbb{F}_2^k$ 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 $\mathbb{F}_2^k$ 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) $(\boldsymbol{a}, b)$ を出力する.

また, オラクル $\mathcal{U}$ を $(\boldsymbol{a}, b) \leftarrow \mathbb{F}_2^{k+1}$ と一様ランダムな組を出力するオラクルとして定義する.

定義 4.1 (探索版 LPN 問題) 探索版 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, $\boldsymbol{s}$ を出力する問題である.

特に $\chi = \mathsf{Ber}_\tau$ のとき, $\mathsf{LPN}_{k,\tau}$ 問題と呼ぶ. また $\mathsf{LPN}_{k,\tau}$ 問題でオラクルからのサンプル数が $n = n(k)$ に制限されるものを, $\mathsf{LPN}_{k,n,\tau}$ 問題と呼ぶ.

定義 4.2 (探索版 LPN 仮定) $\mathbb{F}_2$ 上の確率分布 χ について, 敵 $\mathcal{A}$ の優位性を

$$\mathsf{Adv}_{\mathcal{A}}(k) = \Pr_{\boldsymbol{s} \leftarrow \mathbb{F}_2^k}[\mathcal{A}^{\mathcal{O}_{\boldsymbol{s},\chi}}(1^k) = \boldsymbol{s}]$$

で定義する. 任意の多項式時間の敵 $\mathcal{A}$ について, その優位性が無視できるとき, 探索版 LPN 仮定が成立するという.

暗号プリミティブや暗号プロトコルの安全性証明のために, 判定版 LPN 仮定を用いることも多い. 判定版 LPN 問題と判定版 LPN 仮定は以下で定義される.

定義 4.3 (判定版 LPN 問題) 判定版 LPN 問題とは, オラクル $\mathcal{O}_{\boldsymbol{s},\chi}$ またはオラクル $\mathcal{U}$ へのアクセスが与えられたときに, どちらのオラクルにアクセスしているかを判定する問題である.

定義 4.4 (判定版 LPN 仮定) $\mathbb{F}_2$ 上の確率分布 χ について, 敵 $\mathcal{A}$ の優位性を

$$\mathsf{Adv}_{\mathcal{A}}(k) = \left| \Pr_{\boldsymbol{s} \leftarrow \mathbb{F}_2^k}[\mathcal{A}^{\mathcal{O}_{\boldsymbol{s},\chi}}(1^k) = 1] - \Pr[\mathcal{A}^{\mathcal{U}}(1^k) = 1] \right|$$

で定義する. 任意の多項式時間の敵 $\mathcal{A}$ について, その優位性が無視できる関数であるとき, 判定版 LPN 仮定が成立するという.

探索版 LPN 問題にはランダム自己帰着が存在する [5]. すなわち, 一様ランダムに選ばれた $\boldsymbol{s} \in \mathbb{F}_2^k$ について探索版 LPN 問題を解けるならば, 任意の $\boldsymbol{s} \in \mathbb{F}_2^k$ について探索版 LPN 問題を解くことが出来る.

Katz, Shin, Smith [29] によれば, [5, 44] と同様に判定版 LPN 仮定を探索版 LPN 仮定に帰着することが出来る.

定理 4.5 ([29]) 判定版 $\mathsf{LPN}_{k,\tau}$ 仮定を破る t ステップ, n 回のクエリ, 優位性 δ の敵が存在すると仮定する. このとき, 探索版 $\mathsf{LPN}_{k,\tau}$ 仮定を破る t' ステップ, n' 回のクエリ, 優位性 δ' の敵が存在する. ここで,

$$t' = O(\delta^{-2}tk\log k),\ n' = O(\delta^{-2}n\log k),\ \delta' \geq \delta/4.$$

■**変種**: 以上に列挙した LPN 問題・仮定では, 基礎となる体として $\mathbb{F}_2$ を用いていた. 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. 特に q を素数とした場合には LWE 問題と非常によく似た問題・仮定となるが, 誤差分布 χ の定義が異なることが多い.

LWE 問題では剰余環 $\mathbb{Z}_q$ を用いている. 応用の観点からは, 誤差分布 χ からのサンプル x の絶対値が高い確率で小さいことが重視される. 一方, LPN 問題では有限体 $\mathbb{F}_q$ を用いている. また, 符号からの要求としてハミング重みを考えることが多いため, 誤差分布 χ は 0 を取る確率が大きいことが求められる. たとえば, ベルヌーイ分布の一般化として, 確率 τ で 0 を確率 $1-\tau$ で $\mathbb{F}_q \setminus \{0\}$ のランダムな値を取る分布が用いられる. これは格子問題と符号問題のアナロジーとして考えることができる.

4.1.2 LPN 問題の拡張

4.1.2.1 復号問題

オラクルからのサンプル数を固定し $n = n(k)$ とする. $\mathsf{LPN}_{k,n,\tau}$ 問題での m 個のサンプル $(\boldsymbol{a}_1, b_1), (\boldsymbol{a}_2, b_2), \ldots, (\boldsymbol{a}_n, b_n)$ を行列・ベクトル表示して,

$$\boldsymbol{A} = [\boldsymbol{a}_1^\top \boldsymbol{a}_2^\top \ldots \boldsymbol{a}_n^\top] \in \mathbb{F}_2^{k\times n}, \boldsymbol{b} = \boldsymbol{s}\cdot\boldsymbol{A} + \boldsymbol{e}$$

とする. 符号理論の観点からは, 一様ランダムな行列 $\boldsymbol{A} \in \mathbb{F}_2^{k\times n}$ を生成行列とする $[n,k]_2$ -線形符号の受信語 $\boldsymbol{b}$ から元のメッセージ $\boldsymbol{s}$ を復元する問題と捉えることができる.

4.1.2.2 シンドローム復号問題

先ほど挙げた復号問題の “双対” として, シンドローム復号問題が挙げられる. シンドローム復号問題 $\mathsf{SD}_{k,n,w}$ とは,

$$\boldsymbol{H} = [\boldsymbol{h}_1^\top \boldsymbol{h}_2^\top \ldots \boldsymbol{h}_n^\top] \in \mathbb{F}_2^{(n-k)\times n}, \boldsymbol{u} \in \mathbb{F}_2^k$$

および自然数 w が与えられた時に, $\boldsymbol{e}\cdot\boldsymbol{H}^\top = \boldsymbol{u}$ かつハミング重みが w 以下となる $\boldsymbol{e} \in \mathbb{F}_2^n$ を求める問題である.

$\boldsymbol{A} \in \mathbb{F}^{k\times n}$ で生成される符号のパリティ検査行列を $\boldsymbol{H} \in \mathbb{F}^{(n-k)\times n}$ とし, $\boldsymbol{b}\cdot\boldsymbol{H}^\top (= \boldsymbol{e}\cdot\boldsymbol{H}^\top)$ を $\boldsymbol{u}$ とすれば, $\mathsf{LPN}_{k,n,\tau}$ 問題や復号問題をシンドローム復号問題 $\mathsf{SD}_{k,n,O(\tau n)}$ に変換可能である.

4.1.2.3 Module-LPN 問題

Module-LWE 問題 (3.1.1 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, $\boldsymbol{s} \in R_q^k$ を出力する問題である.

4.1.3 LPN 問題に対する評価

サンプル数を固定した場合, $\boldsymbol{A}$ および $\boldsymbol{b}$ の最悪時を考えると NP 困難になることが Berlekamp, McEliece, van Tilborg [12] によって示されている. [*1] また, Håstad [25] により近似版 LPN 問題[*2] 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. そのため LPN 問題を解くための提案されたアルゴリズムについて調査を行った.

$\mathsf{LPN}_{k,n,\tau}$ 問題を解くための素朴な方法として, 時間 $\mathrm{poly}(n,k)\cdot O(2^k)$ で動作する総当たり法がある. 閾値 $d \geq 1$ を固定する. $\boldsymbol{s} \in \mathbb{F}_2^k$ の候補ごとに, $\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{s}\boldsymbol{A}$ を計算し, $\boldsymbol{e}$ のハミング重みが $(1+1/d)\tau n$ 以下であれば $\boldsymbol{s}$ を解として出力するというものである. Chernoff の補題[*3]から $\boldsymbol{e} \leftarrow \mathsf{Ber}_\tau^n$ としたとき, $d \geq 1$ について $\Pr[\mathsf{HW}(\boldsymbol{e}) \leq (1+1/d)\tau n] \leq \exp(-\tau n/3d^2)$ のため, 高い確率で成功する.

以降では, $O(2^k)$ 以下の時間で解を求めるアルゴリズムについて考察する. 現在, 大別して以下の 3 つのアルゴリズムが知られている.

1. Blum, Kalai, Wasserman [7] の BKW アルゴリズム
2. Arora, Ge [2] の「再線形化」アルゴリズム
3. シンドローム復号問題として解くアルゴリズム

4.1.3.1 BKW アルゴリズムおよびその改良

Blum, Kalai, Wasserman [7] は BKW アルゴリズムと呼ばれるアルゴリズムを提案した.

基本アイデアは以下である. オラクルからのサンプル $(\boldsymbol{a}, b)$ が常に $\boldsymbol{a} = (1, 0, \ldots, 0)$ 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. そこでオラクル $\mathcal{O}_{\boldsymbol{s},\tau}$ からのサンプルを用いて, 上記のようなサンプルを生成することを目指す.

Blum らの見積もりによれば, うまくパラメータを設定することで, 時間計算量・空間計算量ともに $2^{O(k/\log k)}$ を得る. その後, Levieil と Fouque [33], Kirchner [27], Guo, Johansson, Löndahl [23], Zhang, Jiao, Wang [46], Bogos と Vaudenay [13], Esser, Kübler, May [19], Esser, Heuer, Kübler, May, Sohler [18] 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.

■サンプル数が少ない場合: これまでに挙げてきた BKW アルゴリズムおよびその改良では, サンプルが $O(2^{k/\log k})$ 個必要であった. Lyubashevsky [35] はサンプル数が $k^{1+\epsilon}$ 個と少ない場合であっても, BKW アルゴリズムを適用できるような指数個のサンプルの構成法を示している. Kirchner [27] も同様の構成法を示している. Esser, Kubler, May [19] はサンプル数が少ない場合の BKW および Gauss アルゴリズムについて, より詳細な解析を行った.

4.1.3.2 Arora-Ge アルゴリズム

Arora と Ge [2] 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 $\mathsf{LPN}_{k,n,\tau}$ 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.

[*1] $\boldsymbol{A}$ 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.

[*2] $\boldsymbol{A}$ 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.

[*3] $X_1, \ldots, X_n$ を相互に独立な $\{0,1\}$ 確率変数とし, $X = X_1 + \cdots + X_n$ の期待値を μ とすると, $\Pr[X \geq (1+\epsilon)\mu] < \left(\exp(\epsilon)/(1+\epsilon)^{(1+\epsilon)}\right)^\mu$. 特に, $0 < \epsilon \leq 1$ の場合, $\Pr[X \geq (1+\epsilon)\mu] < \exp(-\epsilon^2\mu/3)$.

表 4.1: 確率 1/2 以上で SD 問題を解く場合のパラメータ例 (Full Distance Decoding の場合)

	lg(Time)/n	lg(Space)/n	備考
Pra62 (Lee-Brickel)	0.121	–	[43], [31]
Stern89	0.117	0.0135	[45]
MMT11	0.112	0.0216	[37]
BJMM12	0.102	0.0286	[6]
MO15	0.0967	0.0???	[38]
BM17	0.0953	0.0910	[10]; MO15 を最適化したもの
BM18	0.0885	0.0736	[11]

$\mathrm{poly}(k^w) = 2^{O(\tau n \log k)}$ であるから, $\tau = o(k/(n\log^2 k))$ のようにエラーがスパースであれば, 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, エラーを上記のようにスパース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.

4.1.3.3 SD 問題を経由するアルゴリズム

$\mathsf{LPN}_{k,n,\tau}$ に対応するシンドローム復号問題を考える. ハミング重みを $w \approx \tau n$ とし, $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ および $\boldsymbol{u} \in \mathbb{F}_2^{n-k}$ が与えられ, $\boldsymbol{e}\cdot\boldsymbol{H}^\top = \boldsymbol{u}$ となるような, ハミング重みが w 以下の $\boldsymbol{e} \in \mathbb{F}_2^n$ を探索する問題である.

対応する線形符号の最小距離を d と置く. (2 進符号の場合, Gilvert-Varshamov 限界により, $k/n \approx 1 - H(d/n)$ である[*4].) $w \approx d$ の場合を Full Distance Decoding の場合と呼び, $w \approx d/2$ の場合を Half Distance Decoding と呼ぶ.

この問題を総当りで解く場合には, 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 $O(\binom{n}{w})$ 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 [43] 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 $\boldsymbol{H}$ の列ベクトルを入れ替え, $\tilde{\boldsymbol{H}} = \boldsymbol{H}\cdot\boldsymbol{P}$ とする. ($\boldsymbol{P}$ は置換行列.)
2. $\tilde{\boldsymbol{H}}$ を組織化し, $[\boldsymbol{I}_{n-k} \mid \boldsymbol{Z}] = \boldsymbol{S}\cdot\tilde{\boldsymbol{H}}$ とする.
3. $\boldsymbol{u}' = \boldsymbol{u}\boldsymbol{S}^\top$ を計算する.
4. $\boldsymbol{u}'$ のハミング重みが w 以下であれば, この置換 $\boldsymbol{P}$ を採用し $\boldsymbol{e} = (\boldsymbol{u}', \boldsymbol{0}_k)\cdot\boldsymbol{P}^\top$ を出力する.

$\boldsymbol{u}'$ のハミング重みが w 以下であるため, $\boldsymbol{e}$ のハミング重みも w 以下である. また, $\boldsymbol{e}\cdot\boldsymbol{H}^\top = (\boldsymbol{u}', \boldsymbol{0}_k)\cdot\boldsymbol{P}^\top\boldsymbol{H}^\top = (\boldsymbol{u}', \boldsymbol{0}_k)\cdot\tilde{\boldsymbol{H}}^\top = (\boldsymbol{u}, \boldsymbol{0}_k)\boldsymbol{S}^\top\tilde{\boldsymbol{H}}^\top = (\boldsymbol{u}, \boldsymbol{0}_k)\cdot[\boldsymbol{I}_{n-k} \mid \boldsymbol{Z}]^\top = \boldsymbol{u}$ が成立する. よって, ステップ 4 のチェックを通るならば, $\boldsymbol{e}$ はシンドローム復号問題の解となっている. このような置換は全部で $\binom{n-k}{w}$ 通りあるため, 探索できる確率は $\binom{n-k}{w}/\binom{n}{w}$ となる. 期待計算量は $\mathrm{poly}(n,k)\cdot O(\binom{n}{w}/\binom{n-k}{w})$ となり, 先ほどの列挙法よりも速くなる.

Stern [45] 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アルゴリズムが多数提案されている. 以下では, Both と May [11] による時間計算量の表を, 表 4.1 および 4.2 に示す. この表は, 時間計算量を最小化した場合の $R = k/n$ の最悪時 (1/2 の少し下) 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.

[*4] ここで $H(p) = -p\log(p) - (1-p)\log(1-p)$.

表 4.2: 確率 1/2 以上で SD 問題を解く場合のパラメータ例 (Half Distance Decoding の場合)

	lg(Time)/n	lg(Space)/n	備考
Pra62 (Lee-Brickel)	0.0576	–	[31]
Stern89	0.0557	0.0135	[45]
BLP	0.0555	0.0148	[9]
MMT11	0.0537	0.0216	[37]
BJMM12	0.0494	0.0286	[6]
MO15	0.0473	0.0???	[38]
BM18	0.0465	0.0294	[11]

パラメータ設定によっては, $\mathsf{LPN}_{k,n,\tau}$ 問題を $\mathsf{SD}_{k,n,O(\tau n)}$ 問題に置き換えることで, これらの SD 問題用アルゴリズムも検討する必要がある.

4.1.3.4 量子アルゴリズムへの耐性

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. しかし量子アルゴリズムを利用した攻撃の高速化方法を Kachigar と Tillich [30] が提案している.[*5] Esser, Kübler, May [19] は, BKW や Gauss アルゴリズムの変種を量子アルゴリズムで高速化できる点を指摘している.

4.1.3.5 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.

1. $\mathbb{F}_2$ 係数の一様ランダムな線形符号に対するシンドローム復号問題
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シンドローム復号問題
4. Goppa 符号を用いた Niederreiter 暗号の場合のシンドローム復号問題 (Classic McEliece の一方向性に対応 4.3.1)
5. QC-MDPC 符号に基づくシンドローム復号問題のチャレンジ (BIKE や HQC の一方向性に対応 4.3.2 4.3.3)

のカテゴリが用意されている. 1, 2, 4, 5 に関しては研究および攻撃が進んでおり, 2022 年 10 月現在,

- 1. $n = 550$, $k = n/2$ に対して $w = 67$ (成定, 福島, 清本 2022/02)
- 2. $n = 1280$, $k = n/2$ の場合に $w = 215$ (Neves 2020/06)
- 4. $n = 1284$, $k = 0.8n$ に対して $w = 24$ (Esser, May, Zweydinger 2021/08)
- 5. $n = 3138$, $k = n/2$ に対して $w = 56$ (Esser, Zweydinger 2022/04)

での解が得られている。1 の結果については, Narisada, Fukushima, Kiyomoto [39] を, 4 の結果については, Esser, May, Zweydinger [20] を参照されたい.

[*5] Kirshanova [28] が Kachigar と Tillich の結果 [30] 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, 2018 年時点でのベストな量子アルゴリズムは Kachigar と Tillich [30] であると考えられる.

4.2 代表的な符号に基づく暗号方式の説明

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, S_n で n 次対称群を表し, $\mathrm{GL}_k(\mathbb{F}_q)$ で k 次の $\mathbb{F}_q$ 要素正則行列全体がなす群を表す.

4.2.1 暗号方式 1: McEliece 暗号

McEliece [36] 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 : 誤り訂正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 $\boldsymbol{G}$ 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 $\boldsymbol{c} = \boldsymbol{m}\tilde{\boldsymbol{G}} + \boldsymbol{e}$ を計算する.

復号: $\hat{\boldsymbol{v}} = \boldsymbol{cP}^{-1}$ を計算する. $\hat{\boldsymbol{v}}$ を誤り訂正符号で訂正し $\boldsymbol{m}' \in \mathbb{F}_2^k$ 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,

$$\hat{\boldsymbol{v}} = \boldsymbol{m}\tilde{\boldsymbol{G}}\boldsymbol{P}^{-1} + \boldsymbol{e}\boldsymbol{P}^{-1} = \boldsymbol{mSG} + \boldsymbol{eP}^{-1}$$

を得る. $\boldsymbol{mSG}$ は符号語であり, $\boldsymbol{eP}^{-1}$ は誤りである. $\boldsymbol{eP}^{-1}$ のハミング重みが t 以下であれば, 誤り訂正符号の復号により, $\boldsymbol{m}' = \boldsymbol{mS}$ 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 $\tilde{\boldsymbol{G}}$ が一様ランダムであれば, 暗号文 $\boldsymbol{c}$ は LPN 仮定の下で疑似ランダムである. $\tilde{\boldsymbol{G}}$ が疑似ランダムであることを言うためには, McEliece 仮定と呼ばれる仮定が必要となる.

定義 4.7 (McEliece 仮定) $[n, k]_q$ -符号のクラス $\mathcal{C}$ を固定する. 敵 $\mathcal{A}$ の優位性を

$$\mathsf{Adv}_{\mathcal{A}}(n) = \left| \Pr[\boldsymbol{S} \leftarrow \mathrm{GL}_k(\mathbb{F}_q), \boldsymbol{G} \leftarrow \mathcal{C}, \boldsymbol{P} \leftarrow S_n : \mathcal{A}(1^n, \tilde{\boldsymbol{G}} = \boldsymbol{SGP}) = 1] - \Pr[\tilde{\boldsymbol{G}} \leftarrow \mathbb{F}_q^{k \times n} : \mathcal{A}(1^n, \tilde{\boldsymbol{G}}) = 1] \right|$$

で定義する. 任意の確率的多項式時間の敵 $\mathcal{A}$ について, その優位性が無視できる関数であるとき, McEliece 仮定が成立するという.

左側の敵は McEliece 暗号の公開鍵 (または Niederreiter 暗号の公開鍵の双対) を受け取っている. そのため, この仮定は, McEliece 暗号の公開鍵はランダムな同サイズの行列と見分けが付かないということを意味する.

McEliece 暗号の暗号文の疑似ランダム性は, 判定版 LPN 仮定および McEliece 仮定から言える. また, 暗号文の一方向性は, 探索版 LPN 仮定および McEliece 仮定から言える.

4.2.2 暗号方式 2: Niederreiter 暗号

Niederreiter [40] が 1986 年に提案した. のちに McEliece 暗号と「等価」であることが示された. 詳しくは [32] 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：誤り訂正符号の誤り訂正能力

鍵生成: 誤り訂正能力が t である $[n,k]$ -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 $\boldsymbol{d} = \boldsymbol{e} \cdot \tilde{\boldsymbol{H}}^\top \in \mathbb{F}_2^{n-k}$ を計算する.

復号: $\hat{\boldsymbol{w}} = \boldsymbol{d} \cdot \boldsymbol{T}^{-\top}$ を計算する. $\hat{\boldsymbol{w}}$ を誤り訂正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. $\boldsymbol{d} = \boldsymbol{e} \cdot \tilde{\boldsymbol{H}}^\top$ として, $\hat{\boldsymbol{w}} = \boldsymbol{dT}^{-\top}$ を計算すると,

$$\hat{\boldsymbol{v}} = \boldsymbol{e} \cdot \tilde{\boldsymbol{H}}^\top \boldsymbol{T}^{-\top} = \boldsymbol{e} \cdot \boldsymbol{Q}^\top \boldsymbol{H}^\top \boldsymbol{T}^\top \boldsymbol{T}^{-\top} = \boldsymbol{eQ}^\top \cdot \boldsymbol{H}^\top$$

を得る. $\boldsymbol{eQ}^\top$ のハミング重みが t 以下であれば, 誤り訂正符号の復号により, $\boldsymbol{e}' = \boldsymbol{eQ}^\top$ 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 $\tilde{\boldsymbol{H}}$ が一様ランダムであれば, 暗号文 $\boldsymbol{d}$ は 判定版 LPN 仮定の下で疑似ランダムである. $\tilde{\boldsymbol{H}}$ が疑似ランダムであることを言うためには, McEliece 暗号で McEliece 仮定を考えたように, Niederreiter 仮定を考えればよい.

定義 4.8 (Niederreiter 仮定) $[n,k]_q$ -符号のクラス $\mathcal{C}$ を固定する. 敵 $\mathcal{A}$ の優位性を

$$\mathsf{Adv}_{\mathcal{A}}(n) = \left| \Pr[\boldsymbol{T} \leftarrow \mathrm{GL}_{n-k}(\mathbb{F}_2), \boldsymbol{H} \leftarrow \mathcal{C}, \boldsymbol{Q} \leftarrow S_n : \mathcal{A}(1^n, \tilde{\boldsymbol{H}} = \boldsymbol{THQ}) = 1] - \Pr[\tilde{\boldsymbol{H}} \leftarrow \mathbb{F}_2^{(n-k)\times n} : \mathcal{A}(1^n, \tilde{\boldsymbol{H}}) = 1] \right|$$

で定義する. 任意の確率的多項式時間の敵 $\mathcal{A}$ について, その優位性が無視できる関数であるとき, Niederreiter 仮定が成立するという.

暗号文の疑似ランダム性は, 判定版 LPN 仮定および Niederreiter 仮定から言える. また, 暗号文の一方向性は, 探索版 LPN 仮定および Niederreiter 仮定から言える.

4.3 符号に基づく主要な暗号方式の説明

本稿では以下の暗号方式を取り上げる. いずれも NIST PQC 標準化の中で Round 4 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 暗号的な構成を採用している. McEliece 暗号を採用, 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/LP 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.

4.3.1 暗号方式 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.

表 4.3: 符号に基づく暗号の分類

文献	暗号化	鍵交換	署名
Classic McEliece [47]	○	○	–
BIKE [48]	○	○	–
HQC [49]	○	○	–

- 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: t 誤りを訂正できる 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 $\boldsymbol{T} \in \mathbb{F}_2^{(n-k)\times k}$ とする．復号鍵を符号生成に使ったパラメータ Γ (t 次のモニックな $\mathbb{F}_q$ 係数既約多項式および相異なる $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_q$) とする．

  暗号化 $E(pk, \boldsymbol{e})$: 入力を, $pk = \boldsymbol{T} \in \mathbb{F}_2^{(n-k)\times k}$ と $\boldsymbol{e} \in \mathcal{S}_H(n, t)$ とする．$\tilde{\boldsymbol{H}} = [\boldsymbol{I}_{n-k} \mid \boldsymbol{T}]$ とし, 暗号文として $\boldsymbol{c} = \tilde{\boldsymbol{H}} \cdot \boldsymbol{e} \in \mathbb{F}_2^{n-k}$ を出力する．

  復号 $D(sk, \boldsymbol{c})$: 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, $\boldsymbol{v} = (\boldsymbol{c}, \boldsymbol{0}_k) \in \mathbb{F}^n$ 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, $\mathsf{U}_m^{\not\perp}$ 変換をかけたものとみなせる．(Round 3 までは $\mathsf{HU}_m^{\not\perp}$ を用いていた．以下ではハッシュ関数 H: $\{0,1\}^* \to \{0,1\}^{256}$ 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 T である．復号鍵は Γ に加えて, n ビットの一様ランダムな文字列 s を用いる．

  鍵カプセル化: 1. $e \leftarrow \mathcal{S}_H(n, t)$ をあるアルゴリズムに従ってランダム生成する．
  2. $C = E(pk, e)$ を計算する．
  3. $K = \mathsf{H}(1, e, C)$ とする．
  4. 暗号文は C , セッション鍵は K となる．

  デカプセル: 1. $C \in \mathbb{F}_2^{n-k}$ を入力とする．
  2. $b = 1$ とする．
  3. $e \leftarrow D(sk, C)$ とする．$e = \perp$ であれば, $b = 0$, $e = s$ と上書きする
  4. $K = \mathsf{H}(b, e, C)$ を計算する．
  5. K を出力する．

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。今回末尾に f が着くものは扱っていない（鍵長・暗号文長は f 無しのものと変わらない）．表 4.4 に鍵カプセル化方式の鍵長および暗号文長, 想定セキュリティレベルをまとめた．

表 4.4: Classic McEliece のパラメータ. 単位は全て octet とする.

パラメータ名	レベル	公開鍵長	秘密鍵長	暗号文長
mceliece348864	1	261 120	6 492	96
mceliece460896	3	524 160	13 608	156
mceliece6688128	5	1 044 992	13 932	208
mceliece6960119	5	1 047 319	13 948	194
mceliece8192128	5	1 357 824	14 120	208

4.3.2 暗号方式 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. 以下では, $\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1)$ とする.

  鍵生成: 秘密鍵 h_0 および h_1 を $\mathcal{S}_H(n, w/2)$ から一様ランダムに選ぶ. 公開鍵を $h = h_1/h_0 \in \mathcal{R}$ とする. ((h_0, h_1) を QC-MDPC 符号のパリティ検査行列とし, $(1, h)$ をその組織符号化したものとみなすことができる.)

  暗号化 $E(pk, (e_0, e_1))$: (e_0, e_1) を $\mathcal{S}_H(2n, t)$ 中のベクトルとみなす. $c = e_0 + e_1 h \in \mathcal{R}$ を出力する.

  鍵生成 $D(sk, c)$: ハミング重み t 以下のベクトル (e_0, e_1) を復号する.

  1. ch_0 を計算する.
  2. QC-MDPC 符号の復号アルゴリズムを用いて, ch_0 をシンドロームとするベクトル (e_0, 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. 基本方式から, 平文 m と乱数 (e_0, e_1) に対して $(c_0 = E(pk, (e_0, e_1)), c_1 = m \oplus \mathsf{L}(e_0, e_1))$ と暗号化を行う, 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 $\mathsf{FO}_m^{\not\perp}$ 変換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. (乱数の生成方法は省略する.) 公開鍵は同じく h である. 復号鍵は (h_0, h_1) に加えて, ℓ ビットの一様ランダムな文字列 s を用いる.

  鍵カプセル化: 1. $m \leftarrow \{0,1\}^{256}$ を一様ランダムに選ぶ.
  2. $(e_0, e_1) = \mathsf{G}(m)$ を計算する.
  3. $c_0 = e_0 + e_1 h$ と, $c_1 = m \oplus \mathsf{L}(e_0, e_1)$ を計算する.
  4. $K = \mathsf{H}(m, (c_0, c_1))$ を計算する.
  5. 暗号文を c とし, 鍵を K とする.

  デカプセル: 1. 復号鍵 (h_0, h_1) を用いて c_0 を復号して, (e_0', e_1') を得る.
  2. 復号に失敗したら, $\perp$ を出力して停止
  3. $m' \leftarrow c_1 \oplus \mathsf{L}(e_0', e_1')$ を計算する.
  4. $(e_0', e_1') = \mathsf{G}(m')$ ならば, $K = \mathsf{H}(m', (c_0, c_1))$ を出力して停止

表 4.5: BIKE のパラメータ. 単位は全て octet とする.

パラメータ名	レベル	公開鍵長	秘密鍵長	暗号文長
BIKE-Level1	1	1 541	281	1 573
BIKE-Level3	3	3 083	419	3 115
BIKE-Level5	5	5 122	580	5 154

5. そうでなければ, $K = \mathsf{H}(s, (c_0, c_1))$ を計算し, 出力する.

表 4.5 に鍵カプセル化方式の鍵長および暗号文長をまとめた. 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.

4.3.3 暗号方式 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/LP 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. 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 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: $a \leftarrow \mathcal{R}$, $x, y \leftarrow \mathcal{S}_H(n, w)$ とし, $b = ax + y$ を計算する. 公開鍵を $pk = (a, b) \in \mathcal{R}^2$ とし, 秘密鍵を $sk = (x, y)$ とする.

  暗号化 $E(pk, m; s, e_1, e_2)$: $c = (u, v) = (sa + e_1, sb + e_2 + \mathsf{encode}(m))$ を出力する. ($s \leftarrow \mathcal{S}_H(n, w_r)$, $e_1, e_2 \leftarrow \mathcal{S}_H(n, w_e)$ としている)

  復号 $D(sk, c)$: $c = (u, v)$ に対して, $\mathsf{decode}(v - ux)$ 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, $\mathsf{HFO}^{\perp}$ 変換を適用したものとみなせる. 以下ではハッシュ関数 H, H': $\{0,1\}^* \to \{0,1\}^{256}$ を用いる. また, XOF [*6] として H_G: $\{0,1\}^{\to}\{0,1\}^*$ も用いる. (Round 4 で G への入力に seed と salt が追加された.)

  鍵生成: 同上. ただし a の生成をシード seed から行うこととし, $pk = (\text{seed}, b)$ を公開する. また, 秘密鍵にもシード seed を加える.

  鍵カプセル化: 1. $m \leftarrow \mathbb{F}_2^k$ を一様ランダムにとる.
  2. $\text{salt} \leftarrow \mathbb{F}_2^{128}$ を一様ランダムにとる.
  3. $\theta \leftarrow \mathsf{H}_G(m, \text{seed}, \text{salt})$ を計算する. θ から s, e_1, e_2 を生成する.
  4. $c = (u, v) = E(pk, m; s, e_1, e_2)$ を計算する. $d = \mathsf{H}'(m)$ とする. $K = \mathsf{H}(m, c)$ とする.
  5. 暗号文 $C = (c, d, \text{salt})$, セッション鍵 K を出力する.

  デカプセル: 1. $m' \leftarrow D(sk, c)$ を計算する.
  2. $\theta' = \mathsf{H}_G(m', \text{seed}, \text{salt})$ を計算する. θ' から s', e_1', e_2' を生成する.
  3. $c \neq E(pk, m'; s', e_1', e_2')$ or $d \neq d'$ ならば $\perp$ を出力して停止する.

[*6] eXtendable-Output Functions の略. SHAKE128 や SHAKE256 が例として知られている.

表 4.6: HQC のパラメータ. 単位は全て octet とする.

パラメータ名	レベル	公開鍵長	秘密鍵長	暗号文長
hqc-128	1	2 249	40	4 497
hqc-192	3	4 522	40	9 042
hqc-256	5	7 245	40	14 485

4. $K = \mathsf{H}(m, c)$ を出力する.

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 4.6 に鍵カプセル化方式の鍵長および暗号文長をまとめた. 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 a 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.

4.4 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, 誤り確率 τ が十分大きい場合の LPN 問題を確率的多項式時間で効率的に解くことは困難であると予想されている.

共通鍵や公開鍵の分野で多くの方式が LPN 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
- ID ベース暗号や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, 攻撃に用いられるアルゴリズムの研究は理論的なものが多く, 攻撃実験報告は小さいパラメータに対して行ったものが多い. そのため, 攻撃実験に関する研究もこれから非常に重要である.

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, 枯れていない暗号・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] B. Applebaum, D. Cash, C. Peikert, A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In S. Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009.

[2] S. Arora and R. Ge. New algorithms for learning in presence of errors. In L. Aceto, M. Henzinger, and J. Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.

[3] M. Alekhnovich. More on average case vs approximation complexity. *Computational Complexity*, 20(4):755–786, 2011.

[4] É. Barelli and A. Couvreur. An efficient structural attack on NIST submission DAGS. In Peyrin and Galbraith [42], pages 93–118.

[5] A. Blum, M. L. Furst, M. J. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In D. R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 278–291. Springer, 1993.

[6] A. Becker, A. Joux, A. May, and A. Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In D. Pointcheval and T. 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 520–536. Springer, 2012.

[7] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[8] D. J. Bernstein and T. Lange. Never trust a bunny. In J.-H. Hoepman and I. Verbauwhede, editors, *Radio Frequency Identification. Security and Privacy Issues - 8th International Workshop, RFIDSec 2012, Nijmegen, The Netherlands, July 2-3, 2012, Revised Selected Papers*, volume 7739 of *Lecture Notes in Computer Science*, pages 137–148. Springer, 2012.

[9] D. J. Bernstein, T. Lange, and C. Peters. Smaller decoding exponents: Ball-collision decoding. In P. 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 743–760. Springer, 2011.

[10] L. Both and A. May. Optimizing BJMM with nearest neighbors: Full decoding in $2^{2n/21}$ and McEliece security. In *WCC 2017*, 2017. See `http://wcc2017.suai.ru/proceedings.html`.

[11] L. Both and A. May. Decoding linear codes with high error rate and its impact for LPN security. In Lange and Steinwandt [34], pages 25–46.

[12] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Information Theory*, 24(3):384–386, 1978.

[13] S. Bogos and S. Vaudenay. Optimization of LPN solving algorithms. In J.-H. Cheon and T. Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 703–728, 2016.

[14] N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In C. Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 157–174. Springer, 2001.

[15] S. Devadas, L. Ren, and H. Xiao. On iterative collision search for LPN and subset sum. In Y. Kalai and L. Reyzin, editors, *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 729–746. Springer, 2017.

[16] T. Debris-Alazard and J.-P. Tillich. Two attacks on rank metric code-based schemes: RankSign and an IBE scheme. In Peyrin and Galbraith [42], pages 62–92.

[17] 電子情報通信学会. 知識ベース 1 群 (信号・システム) 2 編　(符号理論). `https://ieice-hbkb.org/portal/doc_608.html`

[18] A. Esser, F. Heuer, R. Kübler, A. May, and C. Sohler. Dissection-BKW. In H. Shacham and A. Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 638–666. Springer, 2018.

[19] A. Esser, R. Kübler, and A. May. LPN decoded. In J. Katz and H. 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 486–514. Springer, 2017.

[20] A. Esser, A. May, and F. Zweydinger. McEliece needs a break – Solving McEliece-1284 and quasi-cyclic-2918 with modern ISD. In O. Dunkelman and S. 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 433–457. Springer, 2022.

[21] J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, and J.-P. Tillich. A distinguisher for high rate McEliece cryptosystems. In *2011 IEEE Information Theory Workshop, ITW 2011, Paraty, Brazil, October 16-20, 2011*, pages 282–286. IEEE, 2011.

[22] J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, and J.-P. Tillich. A distinguisher for high-rate McEliece cryptosystems. *IEEE Trans. Information Theory*, 59(10):6830–6844, 2013.

[23] Q. Guo, T. Johansson, and C. Löndahl. Solving LPN using covering codes. In P. Sarkar and T. Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2014.

[24] H. Gilbert, M. J. B. Robshaw, and Y. Seurin. HB$^{\#}$: Increasing the security and efficiency of HB^{+}. In N. P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 361–378. Springer, 2008.

[25] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[26] S. Heyse, Eike Kiltz, Vadim Lyubashevsky, Christof Paar, and Krzysztof Pietrzak. Lapin: An efficient authentication protocol based on ring-LPN. In A. Canteaut, editor, *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*, pages 346–365. Springer, 2012.

[27] P. Kirchner. Improved generalized birthday attack. *IACR Cryptology ePrint Archive*, 2011:377, 2011.

[28] E. Kirshanova. Improved quantum information set decoding. In Lange and Steinwandt [34], pages 507–527.

[29] J. Katz, J. S. Shin, and A. D. Smith. Parallel and concurrent security of the HB and hb^{+} protocols. *J. Cryptology*, 23(3):402–421, 2010.

[30] G. Kachigar and J.-P. Tillich. Quantum information set decoding algorithms. In T. Lange and T. Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, volume 10346 of *Lecture Notes in Computer Science*, pages 69–89. Springer, 2017.

[31] P. J. Lee and E. F. Brickell. An observation on the security of mceliece's public-key cryptosystem. In C. G. Günther, editor, *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, volume 330 of *Lecture Notes in Computer Science*, pages 275–280. Springer, 1988.

[32] Y. Li, R. H. Deng, and X. Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Trans. Information Theory*, 40(1):271–273, 1994.

[33] É. Levieil and P.-A. Fouque. An improved LPN algorithm. In R. De Prisco and M. Yung, editors, *Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006, Proceedings*, volume 4116 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2006.

[34] T. Lange and R. Steinwandt, editors. *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings*, volume 10786 of *Lecture Notes in Computer Science*. Springer, 2018.

[35] V. Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the

subset sum problem. In C. Chekuri, K. Jansen, J. D. P. Rolim, and L. Trevisan, editors, *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, volume 3624 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2005.

[36] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *Jet Propulsion Laboratory DSN Progress Report*, 42–44:114–116, January and February 1978. `https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF`.

[37] A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In D. H. Lee and X. 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 107–124. Springer, 2011.

[38] A. May and I. Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 203–228. Springer, 2015.

[39] S. Narisada, K. Fukushima, S. Kiyomoto. Multiparallel MMT : Faster ISD algorithm solving high-dimensional syndrome decoding problem. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 2022. `https://search.ieice.org/bin/summary_advpub.php?id=2022CIP0023&category=A&lang=E&abst=`

[40] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problemy Upravleniia i Teorii Informatsii*, 15:19–34, 1986.

[41] R. Nojima, H. Imai, K. Kobara, and K. Morozov. Semantic security for the McEliece cryptosystem without random oracles. *Des. Codes Cryptography*, 49(1-3):289–305, 2008.

[42] T. Peyrin and S. 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*. Springer, 2018.

[43] E. Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Information Theory*, 8(5):5–9, 1962.

[44] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009. Preliminary version was presented at STOC 2005.

[45] J. Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1988.

[46] B. Zhang, L. Jiao, and M. Wang. Faster algorithms for solving LPN. In M. Fischlin and J.-S. Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the*

Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I, volume 9665 of *Lecture Notes in Computer Science*, pages 168–195. Springer, 2016.

[47] D.J. Bernstein et al. Classic McEliece (Round 4 submission). `https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions`

[48] N. Aragon et al. BIKE (Round 4 submission). `https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions`

[49] C. Aguilar Melchor et al. HQC (Round 4 submission). `https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions`

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[*1]（Multivariate Public Key Cryptosystems）における暗号方式の特徴として，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なことが挙げられる．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 次以上に限定できる．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 問題は次のように述べられる．

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．

[*1] かつては「多次多変数公開鍵暗号」と呼ばれていた．

連立方程式 (5.1) の右辺の各 d_i を左辺に移行して $p_i(\mathbf{x})$ 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 [19].

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（2022/9/31 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	$(148, 74)$	$(74, 37)$	$(76, 38)$	$(69, 103)$	$(19, 28)$	$(20, 30)$

5.1.2 MinRank 問題

MinRank 問題 整数 r と行列 $M_1, \ldots, M_k \in \mathbb{F}_q^{m \times n}$ に対し，$x_1, \ldots, x_k \in \mathbb{F}_q$ で，$(x_1, \ldots, x_k) \neq (0, \ldots, 0)$ かつ

$$\operatorname{Rank}\left(\sum_{i=1}^{k} x_i M_i\right) \leq r$$

なるものを求めよ．（$\operatorname{Rank} M$ は行列 M のランクを表す．）

MinRank 問題は HFEv- 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 [9, 4, 28, 1]．MinRank 問題は MP 問題に帰着できることが知られている [23, 17, 2].

例えば，Support minor modeling [2] では以下のように MinRank 問題が MP 問題に帰着される．$x_1, \ldots, x_k$ が MinRank 問題の解であるとするならば，$(S, C) \in \mathbb{F}_q^{m \times r} \times \mathbb{F}_q^{r \times n}$ で

$$SC = \sum_{i=1}^{k} x_i M_i \tag{5.2}$$

なるものが存在する．$\mathbf{r}_j$ を $\sum_{i=1}^{k} x_i M_i$ の第 j 行とすると，(5.2) 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 $C'_j \in \mathbb{F}_q^{(r+1) \times n}$ を

$$C'_j = \begin{pmatrix} \mathbf{r}_j \\ C \end{pmatrix}$$

で定めると，各 $j = 1, \ldots, m$ に対して，$\operatorname{Rank} C'_j \leq r$ 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．$\sharp T = r$

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 と表すことにすると，C'_j の任意の $(r+1)\times(r+1)$ 小行列の行列式は変数 $x_1,\ldots,x_k$ と c_T $(T \subset \{1,\ldots,m\},\ \sharp T = r)$ を用いて多項式で表すことができる．これらの変数の個数は $k + \binom{n}{r}$ である．つまり，MinRank 問題は $k + \binom{n}{r}$ 個の変数の $m\binom{n}{r+1}$ 個の方程式からなる MP 問題に帰着される．

5.1.3 IP 問題，EIP 問題

IP（Isomorphism of Polynomials）問題は以下のように述べられる．

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述べられる．

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 節で見るように，EIP 問題は双極型システムで構成される暗号方式，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 代表的な多変数多項式に基づく暗号方式の説明

5.2.1 双極型システム

IP 問題ベース [24] や MinRank 問題ベース [9, 1, 4, 28] 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 [10] 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，上の例のような 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．$G(\mathbf{x}) \in \mathcal{C}_{\mathrm{cent}}$ 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 $\mathcal{C}_{\mathrm{cent}}$ は公開鍵長（や秘密鍵長）を出来るだけ小さくするために 2 次の多変数多項式系の部分集合で選ぶことが多い．双極型システムは暗号方式，署名方式両方の構成に用いることができる．

暗号方式の暗号化・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 $M \in \mathbb{F}_q^m$ 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，どのようなメッセージ $M \in \mathbb{F}_q^m$ 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 $\mathcal{C}_{\mathrm{cent}}$ 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 } とすると暗号方式が得られる．双極型システムにおいて，$\mathcal{C}_{\rm cent}$ 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 [20].

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．以下では，big field 方式の代表として署名方式 HFE および HFEv-，simple field 方式の代表として署名方式 Rainbow について説明する．

5.2.2 HFE 方式，HFEv-方式

5.2.2.1 暗号方式 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，**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 $\mathcal{C}_{\rm HFE}$ に対し，$\mathcal{C}_{\rm cent} = \mathcal{C}_{\rm HFE}$ の双極型システムは暗号方式を構成する．この暗号方式を HFE [24] 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 [23]．その後，HFE から派生した亜種方式がいくつか提案されており，以下の HFEv- もその 1 つである．

5.2.2.2 署名方式 HFEv-

HFEv- [24] は，暗号方式 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．

$$\mathcal{G}(X) = \sum_{0 \le i \le j}^{q^i+q^j \le D} \alpha_{i,j} X^{q^i+q^j} + \sum_{0 \le i}^{q^i \le D} \beta_i(x_{n+1}, \ldots, x_{n+v}) X^{q^i} + \gamma(x_{n+1}, \ldots, x_{n+v}) \qquad (\alpha_{i,j} \in K). \tag{5.3}$$

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，$\beta_i(x_{n+1}, \ldots, x_{n+v})$ は線形関数，$\gamma(x_{n+1}, \ldots, x_{n+v})$ は 2 次関数である．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 $\mathcal{C}_{\text{HFEv-}}$ と定める．基本的には，これを $\mathcal{C}_{\text{cent}} = \mathcal{C}_{\text{HFEv-}}$ 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 $M \in \mathbb{F}_q^{n-a}$ 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．

HFEv- と同じ構造を持つ署名方式 GeMSS は NIST PQC 標準化の第 3 round の候補に選ばれたが，[30] で効率的攻撃法が提案されたため，第 4 round に進むことはできなかった．

5.2.3 署名方式 Rainbow

署名方式 Rainbow [12] は，双極型システムを用いており，署名方式 UOV [21] を多層化した構造を持っている．UOV の詳細については，5.3.1 節で説明する．

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 の個数は v_i で，O_i 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 $1 \le h \le t$ 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．

署名生成

1. メッセージ $\mathbf{M} \in \mathbb{F}_q^m$ に対し，$\mathbf{c} = (c_{v_1+1}, \ldots, c_n) \leftarrow T^{-1}(\mathbf{M})$ とする．
2. $b_1, \ldots, b_{v_1} \in \mathbb{F}_q$ をランダムにとる．
3. $h = 1, 2, \ldots, t$ に対し，以下を実行：

   $g_{v_h+1}(\mathbf{x}), \ldots, g_{v_{h+1}}(\mathbf{x})$ に $x_1 = b_1, \ldots, x_{v_h} = b_{v_h}$ を代入し，$x_{v_h+1}, \ldots, x_{v_{h+1}}$ に関する 1 次の多項式系 $\bar{g}_{v_h+1}(x_{v_h+1}, \ldots, x_{v_{h+1}}), \ldots, \bar{g}_{v_{h+1}}(x_{v_h+1}, \ldots, x_{v_{h+1}})$ を得る．1 次方程式

   $$\begin{cases} \bar{g}_{v_h+1}(x_{v_h+1}, \ldots, x_{v_{h+1}}) = c_{v_h+1} \\ \qquad\qquad \vdots \\ \bar{g}_{v_{h+1}}(x_{v_h+1}, \ldots, x_{v_{h+1}}) = c_{v_{h+1}} \end{cases}$$

   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 **2** に戻る．
4. $\mathbf{b} \leftarrow (b_1, \ldots, b_n)$.
5. $\sigma \leftarrow S^{-1}(\mathbf{b})$. （$\sigma$ が署名となる．）

Rainbow は NIST PQC 標準化の第 3 round の候補に選ばれたが，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 [5, 7]，その結果，レベル 1, 3, 5 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 157 bits に，272 bits が 206 bits に下がった），第 4 round に進むことはできなかった．

5.3 多変数多項式に基づく主要な暗号方式

MPKC で標準化が期待されるのは署名方式の UOV である．5.2.3 節で述べたように Rainbow は UOV を多層化したものであるが，その多層化が結果として暗号の強度を弱めることになった [5, 7]．しかし，多層化していない UOV は高い安全性を維持している．また，NIST の PQC の新たな署名方式の公募 [29] では，短い署名と効率的な検証を持つ方式を望んでおり，UOV はその性質を持つため標準化が期待される．

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 [21]			○

5.3.1 署名方式 UOV

5.3.1.1 UOV の概要

UOV [21] は，双極型システムを用いた署名方式であり，署名長が短く，検証が速いという長所を持つ．署名生成では，線形連立方程式の求解を利用しており，2 次以上の連立方程式の求解は必要としない．UOV の亜種方式として，QR-UOV [18] や MAYO [6] が提案されている．

v, o を正の整数とし，$m = o$, $n = v + o$ 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．$G(\mathbf{x})$ は逆写像を効率的に計算することができる．具体的には，任意の $\mathbf{c} = c_i) \in \mathbb{F}_q^m$ 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 $x_1 = b_1, \ldots, x_v = b_v$ を代入し，$x_{v+1}, \ldots, x_n$ に関する 1 次の多項式系 $\bar{g}_1(x_{v+1}, \ldots, x_n)$, $\ldots, \bar{g}_m(x_{v+1}, \ldots, x_n)$ を得る．1 次方程式

$$\begin{cases} \bar{g}_1(x_{v+1}, \ldots, x_n) = c_1 \\ \qquad\qquad \vdots \\ \bar{g}_m(x_{v+1}, \ldots, x_n) = c_m \end{cases}$$

   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 **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，アフィン同型写像 T は UOV の安全性には貢献しないので，通常は T は恒等写像で選ぶ．$\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$ を暗号学的ハッシュ関数とする．

鍵生成

1. $G(\mathbf{x}) \in \mathcal{C}_{\mathrm{UOV}}$ 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 をランダムに選ぶ．
3. S^{-1} を計算する．
4. $F(\mathbf{x}) = G(\mathbf{x}) \circ S$.

公開鍵は $F(\mathbf{x})$，秘密鍵は $G(\mathbf{x}), S^{-1}$ 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. $\mathbf{c} = (c_{v+1}, \ldots, c_n) \leftarrow \mathcal{H}(M)$.
2. $b_1, \ldots, b_v \in \mathbb{F}_q$ をランダムにとる．
3. $g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})$ に $x_1 = b_1, \ldots, x_v = b_v$ を代入し，$x_{v+1}, \ldots, x_n$ に関する 1 次の多項式系 $\bar{g}_1(x_{v+1}, \ldots, x_n)$, $\ldots, \bar{g}_m(x_{v+1}, \ldots, x_n)$ を得る．1 次方程式

$$\begin{cases} \bar{g}_1(x_{v+1}, \ldots, x_n) = c_1 \\ \qquad\qquad\qquad\vdots \\ \bar{g}_m(x_{v+1}, \ldots, x_n) = c_m \end{cases}$$

   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 **3** に戻る．
4. $\mathbf{b} \leftarrow (b_1, \ldots, b_n)$.
5. $\sigma \leftarrow S^{-1}(\mathbf{b})$.

σ が署名となる．最後に検証である．

検証

1. $\mathbf{h} \leftarrow \mathcal{H}(M)$.
2. $\mathbf{h}' \leftarrow F(\sigma)$.
3. $\mathbf{h} = \mathbf{h}'$ の真偽を返す．

検証者は，$\mathbf{h} = \mathbf{h}'$ のとき，署名を受理し，それ以外は棄却する．

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，有限体の位数 q，方程式数 m, 変数の個数 n である．PQC Forum [27] 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
$(16, 64, 160)$	レベル 1	3,297,280 Bytes	412,160 Bytes	640 bits
$(256, 44, 112)$	レベル 1	2,227,456 Bytes	278,432 Bytes	896 bits
$(256, 72, 184)$	レベル 3	9,803,520 Bytes	1,225,440 Bytes	1472 bits
$(256, 96, 244)$	レベル 5	22,955,520 Bytes	2,869,440 Bytes	1952 bits

5.4　多変数多項式に基づく暗号技術に関するまとめ

多変数公開鍵暗号は位数が小さな有限体上の多項式を利用しており，暗号化や検証が効率的に実行できる．また，署名方式 UOV に代表されるように署名長を短く抑えることができる．双極型システムでは公開鍵（や秘密鍵）が多変数多項式写像の係数集合となるため，鍵長が大きくなりやすいことが課題である．また，署名方式に比べ，暗号方式の構成が難しいことや高機能暗号への応用が少ないことも課題である．

署名方式 Rainbow は NIST PQC 標準化の第 4 round には進めなかったが，致命的な攻撃が報告されているわけではないので今後も安全な方式として期待される．Big field 方式を用いて構成される暗号方式は現時点では有力なものはないが，定期的に big field 方式を用いた暗号方式は提案されているので，今後の新たな方式の出現が期待できる．本稿では MQ 問題ベースの双極型システムのみを説明したが，まだ数は少ないが IP 問題ベースや MinRank 問題ベースの方式も存在する．近年，MinRank 問題の解析が進歩しているため，MinRank 問題ベースの新たな方式の出現も今後，期待できる．

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. IACR Cryptol. ePrint Arch. 2022: 1501, 2022.

[2] 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'20, Springer LNCS vol. 12491, pp. 507–536. 2020.

[3] M. Bardet, J.-C. Faugère, B. Salvy, B.-Y. Yang. Asymptotic expansion of the degree of regularity for semi-regular systems of equations. In Effective Methods in Algebraic Geometry (MEGA), pp. 71–74, 2004.

[4] E. Bellini, A. Esser, C. Sanna, J. Verbel. MR-DSS - smaller MinRank-based (ring-)signatures. Cryptology ePrint Archive, Paper 2022/973, 2022.

[5] W. Beullens. Improved cryptanalysis of UOV and Rainbow. Cryptology ePrint Archive, Paper 2020/1343, 2020.

[6] W. Beullens. MAYO: Practical Post-Quantum Signatures from Oil-and-Vinegar Maps. Cryptology eprint Archive, Paper 2021/1144, 2021.

[7] W. Beullens. Breaking Rainbow Takes a Weekend on a Laptop. Crypto'22, Springer LNCS vol. 13508, pp. 464–479, 2022.

[8] A. Caminata, E. Gorla. Soving degree, last fall degree, and related invariants. Journal of Symbolic Computation, vol. 114, pp. 322–335, 2023.

[9] N. T. Courtois. Efficient Zero-Knowledge Authentication Based on a Linear Algebra Problem MinRank. ASIACRYPT'01, Springer LNCS vol. 2248, pp. 402–421, 2001.

[10] J. Ding, J. E. Gower, D. S. Schmidt. Multivariate Public Key Cryptosystems. Advances in Information Security 25, Springer, 2006.

[11] J. Ding and T. J. Hodges. Inverting HFE Systems Is Quasi-Polynomial for All Fields. CRYPTO'11, Springer LNCS vol. 6841, pp. 724–742, 2011.

[12] J. Ding and D. Schmidt. Rainbow, a New Multivariable Polynomial Signature Scheme. ACNS'05 Springer LNCS vol. 3531, pp. 164–175, 2005.

[13] J. Ding, B.-Y. Yang, C.-H. O. Chen, M.-S. Chen, C. M. Cheng. New Differential-Algebraic Attacks and Reparametrization of Rainbow. ACNS'08, Springer LNCS vol. 5037, pp. 242–257, 2008.

[14] V. Dubois and N. Gama. The Degree of Regularity of HFE Systems. ASIACRYPT'10, Springer LNCS vol. 6477, pp. 557–576, 2010.

[15] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). J. Pure Appl. Algebra, vol. 139 pp. 61–88, 1999.

[16] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). Proceedings of ISSAC'02, pp. 75–83, ACM Press, 2002.

[17] 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. In ISSAC'10, Proceedings, pp. 257–264, ACM, 2010.

[18] H. Furue, Y. Ikematsu, Y. Kiyomura, and T. Takagi. A New Variant of Unbalanced Oil and Vinegar Using Quotient Ring: QR-UOV. ASIACRYPT'21, Springer LNCS vol. 13093, pp. 187–217, 2021.

[19] M. R. Garay and D. S. Johnson. A Guide to the Theory of NP-Completeness. In Computers and Intractability, W.H.Freeman, 1979.

[20] L. Goubin and N.T. Courtois. Cryptanalysis of the TTM Cryptsystem. ASIACRYPT'00, Springer LNCS vol. 1976, pp. 44–57, 2000.

[21] A. Kipnis, L. Patarin, L. Goubin. Unbalanced Oil and Vinegar Schemes. EUROCRYPT'99, Springer LNCS vol. 1592, pp. 206–222, 1999.

[22] A. Kipnis, and A. Shamir. Cryptanalysis of the Oil and Vinegar Signature Scheme. CRYPTO'98, Springer LNCS vol. 1462, pp. 257–266, 1998.

[23] A. Kipnis, and A. Shamir. Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. CRYPTO'99, Springer LNCS, vol. 1666, pp. 19–33, 1999.

[24] J. Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. EUROCRYPTO'96, Springer LNCS vol. 1070, pp. 33–48, 1996.

[25] J. Patarin. The Oil and Vineger Signature Scheme. Dagstuhl Workshop on Cryptography, 1997.

[26] A. Petzoldt, M.-S. Chen, B.-Y. Yang, C. Tao, J. Ding. Design principles for HFEv-based multivariate signature schemes. ASIACRYPT'15, Springer LNCS vol. 9742 , pp. 311–334, 2015.

[27] PQC Forum, `https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/B1RFy31rH8I/m/km5Ow_GmAgAJ`.

[28] B. Santoso, Y. Ikematsu, S. Nakamura, T. Yasuda. Three-Pass Identification Scheme Based on MinRank Problem with Half Cheating Probability. CoRR abs/2205.03255, 2022.

[29] 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`

[30] C. Tao, A. Petzoldt, J. Ding. Efficient Key Recovery for all HFE Signature Variants. CRYPTO'21, Springer LNCS vol. 12825, pp. 70–93, 2021.

[31] C. Wolf. Taxonomy of public key schemes based on the problem of Multivariate Quadratic equations. Cryptology ePrint Archive, Paper 2005/077, 2005.

[32] B.-Y. Yang and J.-M. Chen. All in the XL Family: Theory and Practice. ICISC'04, Springer LNCS vol. 3506, pp. 67–86, 2005.

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いる．

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 [10] に対する解法 [3, 27, 30] が適用できない計算問題として，自己準同型環計算問題及び SQISign（Short Quaternion and Isogeny Signature）署名方式 [15] の安全性に関する計算問題の概要を記述していく．SIDH 同種写像問題に対する解法に関しては調査報告書を参照のこと．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 [18] を取り上げる．6.3 節では, 主要な暗号方式として，GPS 署名を改良した 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しか扱わない．しかし，通常楕円曲線に基づく CRS (Couveignes–Rostovtsev–Stolbunov) 鍵共有法 [9, 31] を改良した De Feo ら [14] の方式は, それ自体は実用的な性能にはまだ遠いが, 調査報告書に記載されている CSIDH 鍵共有の原型を与えているという点で重要である．

同種写像の数学的詳細については, De Feo の概説記事 [13] や Washington の楕円曲線の教科書 [35] を，四元数環については Voight の教科書 [34] を参照のこと．また, Galbraith–Vercauteren による同種写像関連問題のサーベイ [19] も参照する．

■記法　$x \leftarrow_R X$ は, x を集合 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, 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．

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

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, $C = \ker(\varphi)$ となる 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 [33]. (モンゴメリ型楕円曲線に対する Vélu の公式に関しては, [29] 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 [19]) *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.

定義 6.1 において, φ の次数が多項式サイズであれば, 上記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, Galbraith ら [19] は, j 不変量を使って, 上記問題を定式化しているが, 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して扱う必要性が生じるため, 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 [22] や Galbraith による同種写像問題に関する研究 [17] 及び Couveignes と Rostovtsev–Stolbunov による初期の暗号応用への提案 [9, 31] 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 [4] 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．古典解読アルゴリズムは Galbraith [17] による中間一致攻撃で, 解読時間は $\tilde{O}(\sqrt{p})$ であり，量子解読アルゴリズムは Biasse ら [1] 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 [5] 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 [7], Longa ら [26] による報告，Udovenko–Vitto [32] による$IKEp182 Challenge [6] 解読報告，Jaques–Schanck [21] 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

号化 [20]）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．

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 [22], Galbraith [17], Couveignes [9] 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 [23] 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．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 [11, 12, 36]，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 [18, 15, 16]．

■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 [24, 25] を参照する．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 [22]) 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 [11, 12] 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 [36] 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．

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，[12] 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 [25, 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．

[*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．

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 署名の安全性に関する計算問題

次に，SQISign 署名の安全性を示すために必要な計算問題を述べる．

■SQISign 署名の健全性に関する計算問題 まずは，SQISign 署名の健全性（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（SEP：Smooth Endomorphism Problem）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 [15, 25]) 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．

上記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，[12] 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，上記問題は，本質的に自己準同型環計算問題と同値である [15]．よって，$\tilde{O}(\sqrt{p})$ 時間での古典アルゴリズム [12] 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．

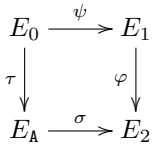
標数 p と ∞ のみで分岐する四元数環 $\mathcal{B}_{p,\infty} := \mathbb{Q}[i, j]$ における極大整環 $\mathcal{O}_0 \subset \mathcal{B}_{p,\infty}$ は，最小判別式の 2 次整環である $\mathfrak{O} \subset \mathcal{O}_0 \cap \mathbb{Q}[i]$ による $\mathfrak{O} + j\mathfrak{O} \subset \mathcal{O}_0$ が部分整環であり $\mathfrak{O} \subset (j\mathfrak{O})^\perp$ と直交分解しているとき[*2]，特殊極値的

[*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

(special extremal) であるという．詳細は [15, 25] を参照．$p = 7 \bmod 12$ の時，上記の 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 及び SQISign 署名の公開パラメータの一部として必要である．

■SQISign 署名の零知識性に関する計算問題　SQISign 署名では，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]．[15, 16] 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 署名のランダム識別問題 [15, 25]) $\tau : E_0 \to E_{\mathtt{A}}$ 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 $\mathtt{Iso}_{D,j(E_{\mathtt{A}})}$ から一様に選ばれたか，$[\tau]_*\mathcal{P}_{N_\tau}$ から一様に選ばれたかを判定せよ．

SQISign 署名の提案者によると，現在のところ，SQISign 署名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 [15, 25]．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，上記の SQISign 署名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，調査報告書に記載されている最近の 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　代表的な同種写像に基づく暗号方式の説明

6.2.1　GPS 署名

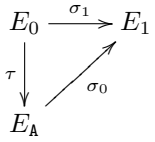
Galbraith–Petit–Silva (GPS) [18] によって始めて自己準同型環の知識証明に基づく署名方式が提案された．GPS 署名は実際に利用するのは困難であろうと思われているが，現在，GPS 署名は，SQISign 署名の原型を与えているという点で重要である．6.1.2 節で述べた Deuring 対応と KLPT アルゴリズム [23] が GPS 署名の理論的基礎を与える．

役の定義は，例えば [25] 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．

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 アルゴリズムによって「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 署名である．ここでは [18, 第 4 章] と [25, 5.1.2 節] に基づいて GPS 署名を記述する．また，[18, 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，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急攪拌性定理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 アルゴリズムによって「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，不受理を出力する．

GPS 署名は，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 [18, 定理 10]．GPS 署名では，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 [25, 5.1.2 節]．以上，GPS 署名には (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の説明

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．

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 [15]			○

6.3.1 SQISign 署名

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 署名を概説する．SQISign 署名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．6.2.1 節で述べた GPS 署名を基にして改良を加えた署名方式が SQISign 署名であり，ASIACRYPT 2020 で De Feo–Kohel–Leroux–Petit–Wesolowski [15] により提案された．6.2.1 節末尾に付した GPS 署名の 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演算効率も大きく改善した [15, 16]．

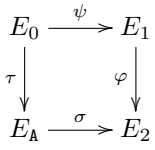
また，安全性に関しては，健全性は超特異平滑自己準同型写像計算問題（定義 6.3）の困難性に基づき，零知識性は定義 6.4 で述べた SQISign 署名のランダム識別問題の困難性に基づいている．初期提案 [15] では，ノルム方程式を解くサブルーチンに不備があり，生成される署名同種写像 σ に偏りが生じていたことが [16] において指摘された．そして，更に [16] でその不備を除去したアルゴリズム提案が行われた．

具体的なパラメータ，特に適切な SQISign 素数（SQISign-friendly prime）p を生成する問題は非自明であり，初期提案 [15] から始まり，[8, 16, 2] と現在も進行中の研究テーマである．その現状報告を後ほど行う．また，小貫 [28] により，SQISign 鍵生成で得られる鍵の分布の理論・実験による解析がなされており，[15] で述べられた従来仕様の不備を指摘して，改善アルゴリズムを提案している．

以下では，方式記述，SQISign 署名パラメータ，実装報告の順に既存の研究報告をまとめる．

■SQISign 署名方式記述　SQISign 署名では，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 アルゴリズムで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．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, 不受理を出力する．

既に述べたように，SQISign 署名の安全性は，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．

■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 であるので，それら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，超特異楕円曲線の位数 $\#E(\mathbb{F}_{p^2}) = p^2 - 1$ を考慮して，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，$T \approx p^{5/4+\epsilon}$ ([2] 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 [16].

NIST 安全性レベル 1： SQISign 素数は，NIST 安全性レベル 1 については，[16] において，XGCD アルゴリズムに基づいて，$B = 3923$ に関して B-平滑な T を持つ以下の 254 bits 素数 p が生成された．$f = 66$ であり，T は式 (6.1) で灰色でない B 以下の奇素因数の積で与えられる．T に関しては他の SQISign 素数についても同様である．また，[15, 25] によればレベル 1 パラメータでは署名 σ の次数 $D = 2^e$ の指数 e が $e = 1000$ で与えられている．

$$\begin{aligned} p + 1 &= 2^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 19 \cdot 29^2 \cdot 37^2 \cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461 \cdot 521 \cdot 3923 \cdot 62731 \cdot 96362257 \cdot 3924006112952623, \\ p - 1 &= 2 \cdot 3^{65} \cdot 13 \cdot 17 \cdot 43 \cdot 79 \cdot 157 \cdot 239 \cdot 271 \cdot 283 \cdot 307 \cdot 563 \cdot 599 \cdot 607 \cdot 619 \cdot 743 \cdot 827 \cdot 941 \cdot 2357 \cdot 10069. \end{aligned} \tag{6.1}$$

同じく NIST 安全性レベル 1 で最良の SQISign 素数として，[2] において，$B = 523$ に関して B-平滑な T を持つ以下の 254 bits 素数 $p = 2r^3 - 1$ ($r = 204614491255000374748856320$) が生成されている．$f = 47$ である．

$$\begin{aligned} p + 1 &= 2^{46} \cdot 5^3 \cdot 13^3 \cdot 31^3 \cdot 73^3 \cdot 83^3 \cdot 103^3 \cdot 107^3 \cdot 137^3 \cdot 239^3 \cdot 271^3 \cdot 523^3, \\ p - 1 &= 2 \cdot 3^3 \cdot 7 \cdot 11^2 \cdot 17^2 \cdot 19 \cdot 101 \cdot 127 \cdot 149 \cdot 157 \cdot 167 \cdot 173 \cdot 199 \cdot 229 \cdot 337 \cdot 457 \cdot 479 \cdot \\ &\quad 141067 \cdot 3428098456843 \cdot 484047594531861479165862 1. \end{aligned} \tag{6.2}$$

NIST 安全性レベル 3 : NIST 安全性レベル 3,5 に関しては [16] までは適切なパラメータ例が与えられていなかったが，レベル 3 について，後続研究である [2] において最良の SQISign 素数として，$B = 10243$ に関して B-平滑な T を持つ以下の 382 bits 素数 $p = 2r^6 - 1$ ($r = 11896643388662145024$) が生成されている．$f = 80$ である．

$$\begin{aligned} p + 1 &= 2^{79} \cdot 3^6 \cdot 23^{12} \cdot 107^6 \cdot 127^6 \cdot 307^6 \cdot 401^6 \cdot 547^6, \\ p - 1 &= 2 \cdot 5^2 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 47 \cdot 71 \cdot 79 \cdot 109 \cdot 149 \cdot 229 \cdot 269 \cdot 283 \cdot 349 \cdot 449 \cdot 463 \cdot 1019 \cdot 1033 \cdot 1657 \cdot 2179 \cdot \\ &\quad 2293 \cdot 4099 \cdot 5119 \cdot 10243 \cdot 381343 \cdot 19115518067 \cdot 740881808972441233 \cdot 832321437914821351639 21. \end{aligned} \tag{6.3}$$

NIST 安全性レベル 5 : また，レベル 5 についても，[2] において最良の SQISign 素数として，$B = 150151$ に関して B-平滑な T を持つ以下の 508 bits 素数 $p = 2r^6 - 1$ ($r = 2669797390044648368060825 6$) が生成されている．$f = 86$

である．

$$p+1 = 2^{85} \cdot 17^{12} \cdot 37^6 \cdot 59^6 \cdot 97^6 \cdot 233^6 \cdot 311^{12} \cdot 911^6 \cdot 1297^6, \tag{6.4}$$
$$p-1 = 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11^2 \cdot 23^2 \cdot 29 \cdot 127 \cdot 163 \cdot 173 \cdot 191 \cdot 193 \cdot 211 \cdot 277 \cdot 347 \cdot 617 \cdot 661 \cdot 761 \cdot 1039 \cdot 4637 \cdot 5821 \cdot 15649 \cdot 19139 \cdot 143443 \cdot 150151 \cdot 3813769 \cdot 358244059 \cdot 992456937347 \cdot 353240481781965369823897507 \cdot 860102006951457440137 1658891403021.$$

■SQISign 署名実装報告　NIST レベル 3,5 パラメータ (式 (6.3), (6.4)) に関しては，まだ実装報告が公開されていない．以下では，NIST レベル 1 パラメータ (式 (6.1), (6.2)) に関する実装報告をまとめる．

データサイズ : 式 (6.1), (6.2) の NIST レベル 1 パラメータに関して，秘密鍵サイズ 16 Bytes，公開鍵サイズ 64 Bytes，署名サイズ 204 Bytes が報告されている [15, 16].

演算時間 : 式 (6.1) の NIST レベル 1 パラメータに関して，演算時間中央値は，鍵生成 218 ms（ミリ秒），署名生成 1081 ms，署名検証 19 ms と報告されている [16]．詳細は [16] を参照のこと．

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．特に，SQISign 署名に関して, 関連する GPS 署名も含めた方式記述と安全性研究についてまとめてきた.

[9] 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 [22] や Galbraith [17] 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 25 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 第 4 ラウンド コンペティションにも選ばれた SIKE 暗号化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 [3] を始めとする一連の攻撃法 [27, 30] は SIDH 鍵共有に対して決定的な結果をもたらした.

一方，それは，この 25 年の研究の一つの到達点として，同種写像暗号全体に対して基本的な安全性指標を提示することになった．現在，その新しい安全性指標に基づいて更に安全性解析が進展していると共に，また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．例えば，調査報告書に記載されている最近新規提案された M-SIDH 鍵共有・MD-SIDH 鍵共有の安全性研究は，今後の重要な研究課題の一つである.

現在，特に，公開鍵と署名を合わせたサイズが短い SQISign 署名が注目されていると共に，調査報告書に記載されている 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，標準化などを考慮するのであれば，実装研究を進める必要があり，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もある.
- 調査報告書に記載されている CSIDH ベース署名である SeaSign 署名と CSI-FiSh 署名についても，今後の耐量子計算機署名として研究が進められているが，安全性評価が定まった実用的な署名を得るためには，まだ今後の安全性・実装研究が必要である．そして，リング署名・グループ署名などの高機能暗号系への応用研究も重要であり，今後の研究動向に注目する必要がある.

- （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関しては，これまで主に，SIKE 暗号化パラメータに関係して，具体的な安全性評価が行われてきた．SQISign 署名パラメータを具体的に決めていくためには，SQISign 署名パラメータに対して，上記問題に対する古典・量子アルゴリズムの詳細な解析・見積もりを行うことが重要であり, 今後の課題である．
- 上で述べたように現在研究が進展している新しい安全性指標に基づいて，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暗号技術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] J. Biasse, D. Jao, and A. Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In *INDOCRYPT 2014*, pages 428–442. Springer, 2014.

[2] G. Bruno, M. C.-R. Santos, C. Costello, J. K. Eriksen, M. Meyer, M. Naehrig, and B. Sterner. Cryptographic smooth neighbors. *IACR Cryptol. ePrint Arch.*, 2022. `http://eprint.iacr.org/2022/1439`.

[3] W. Castryck and T. Decru. An efficient key recovery attack on SIDH (preliminary version). *IACR Cryptol. ePrint Arch.*, 2022. `http://eprint.iacr.org/2022/975`, To appear in EUROCRYPT 2023.

[4] D. Charles, K. Lauter, and E. Goren. Cryptographic hash functions from expander graphs. *J. Crypt.*, 22(1):93–113, 2009.

[5] A. M. Childs, D. Jao, and V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *J. Math. Cryptol.*, 8(1):1–29, 2014.

[6] C. Costello. The case for SIKE: A decade of the supersingular isogeny problem. *IACR Cryptol. ePrint Arch.*, 2021. `http://eprint.iacr.org/2021/543`.

[7] C. Costello, P. Longa, M. Naehrig, J. Renes, and F. Virdia. Improved classical cryptanalysis of SIKE in practice. In *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 505–534. Springer, 2020.

[8] C. Costello, M. Meyer, and M. Naehrig. Sieving for twin smooth integers with solutions to the Prouhet-Tarry-Escott problem. In *EUROCRYPT 2021, Part I*, pages 272–301. Springer, 2021.

[9] J. Couveignes. Hard homogeneous spaces. *IACR Cryptol. ePrint Arch.*, 2006. `http://eprint.iacr.org/2006/291`.

[10] L. De Feo, D. Jao, and J. Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Math. Crypt.*, 8(3):209–247, 2014.

[11] K. Eisenträger, S. Hallgren, K. E. Lauter, T. Morrison, and C. Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 329–368. Springer, 2018.

[12] K. Eisenträger, S. Hallgren, C. Leonardi, T. Morrison, and J. Park. Computing endomorphism rings of supersingular elliptic curves and connections to pathfinding in isogeny graphs. In *ANTS 2020*, volume 4 of *The Open Book Series*, pages 215–232. Mathematical Sciences Publishers, 2020.

[13] L. D. Feo. Mathematics of isogeny based cryptography. *CoRR*, abs/1711.04062, 2017.

[14] L. D. Feo, J. Kieffer, and B. Smith. Towards practical key exchange from ordinary isogeny graphs. In *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 365–394. Springer, 2018.

[15] L. D. Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In *ASIACRYPT 2020, Part I*, pages 64–93. Springer, 2020.

[16] L. D. Feo, A. Leroux, and B. Wesolowski. New algorithms for the Deuring correspondence: SQISign twice as fast. *IACR Cryptol. ePrint Arch.*, 2022. `https://eprint.iacr.org/2022/234`.

[17] S. Galbraith. Constructing isogenies between elliptic curves over finite fields. *Journal of Computational Mathematics*, 2:118–138, 1999.

[18] S. D. Galbraith, C. Petit, and J. Silva. Identification protocols and signature schemes based on supersingular isogeny problems. *J. Cryptol.*, 33(1):130–175, 2020.

[19] S. D. Galbraith and F. Vercauteren. Computational problems in supersingular elliptic curve isogenies. *Quantum Inf. Process.*, 17(10):265, 2018.

[20] D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. D. Feo, B. Hess, A. Jalali, B. Koziel, B. LaMacchia, P. Longa, M. Naehrig, G. Pereira, J. Renes, V. Soukharev, and D. Urbanik. SIKE: Supersingular isogeny key encapsulation. *submission to the NIST's PQC standardization, round 3*, October 2020.

[21] S. Jaques and J. M. Schanck. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 32–61. Springer, 2019.

[22] D. Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California at Berkeley, 1996.

[23] D. Kohel, K. Lauter, C. Petit, and J.-P. Tignol. On the quaternion ℓ-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17:418–432, 2014. Special Issue A: Algorithmic Number Theory Symposium XI.

[24] A. Leroux. A new isogeny representation and applications to cryptography. In *ASIACRYPT 2022*, LNCS. Springer, 2022.

[25] A. Leroux. *Quaternion algebras and isogeny-based cryptography*. PhD thesis, Ecole Polytechnique, 2022.

[26] P. Longa, W. Wang, and J. Szefer. The cost to break SIKE: A comparative hardware-based analysis with AES and SHA-3. In *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 402–431. Springer, 2021.

[27] L. Maino and C. Martindale. An attack on SIDH with arbitrary starting curve (draft). *IACR Cryptol. ePrint Arch.*, 2022. `http://eprint.iacr.org/2022/1026`.

[28] H. Onuki. On the key generation in SQISign. In *NutMic 2021*. Springer, 2022.

[29] J. Renes. Computing isogenies between Montgomery curves using the action of (0, 0). In *PQCrypto 2018*, volume 10786 of *LNCS*, pages 229–247. Springer, 2018.

[30] D. Robert. Breaking SIDH in polynomial time. *IACR Cryptol. ePrint Arch.*, 2022. `http://eprint.iacr.org/2022/1038`.

[31] A. Rostovtsev and A. Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptology ePrint Archive*, 2006. `http://eprint.iacr.org/2006/145`.

[32] A. Udovenko and G. Vitto. Breaking the $IKEp182 challenge. *IACR Cryptol. ePrint Arch.*, 2021. `http://eprint.iacr.org/2021/1421`.

[33] J. Vélu. Isogénies entre courbes elliptiques. *C.R. Acad. Sc. Paris, Séries A.*, 273:238–241, 1971.

[34] J. Voight. *Quaternion Algebras*. Springer, June 2022.

[35] L. Washington. *Elliptic Curves: Number Theory and Cryptography*. CRC Press, 2nd edition, 2008.

[36] B. Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *FOCS 2021*, pages 1100–1111. IEEE, 2022.

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 [11, 25]．また，この方式を改良した Winternitz one-time signature が Merkle [29] 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 [28, 29] 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，ハッシュ関数の各入力に対する出力が無作為に選択されるような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，第二原像を得るのに必要な計算時間は $\Theta(|\mathcal{R}|)$ 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 [16] を用いることにより，第二原像を得るのに必要な計算時間は $\Theta(\sqrt{|\mathcal{R}|})$ 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，SHA-2 [13], SHA-3 [15], Haraka [23] 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．これらのうち，SHA-2, SHA-3 は米国 NIST の指定する標準ハッシュ関数族である．

SHA-2 は Merkle-Damgård 構造 [10, 30] を有するハッシュ関数の族であり，Secure Hash Standard [13] 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 は置換を用いたスポンジ構造 [6] 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．

Haraka はブロック暗号 AES [8, 14] に基づく置換を用いた Davies-Meyer 構造 [33] に基づくハッシュ関数の族であり，Haraka-256, Haraka-512 からなる．Haraka-256, Haraka-512 の末尾の数値は入力の bit 長を表す．出力長はいずれも 256 bits である．Haraka は，ハッシュ関数に基づく署名技術での使用を想定し，短い入力に対して高い処理性能

を達成するよう設計されている．

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 bits の 2 進数表記を表す．
- $\mathbb{B} := \{0,1\}^8$ とする．

7.2 代表的なハッシュ関数に基づく署名方式

7.2.1 Winternitz One-Time Signature

Winternitz one-time signature [29] 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，$0 \le k \le \ell_{\mathrm{m}}-1$ 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. $0 \le k \le \ell-1$ 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. $0 \le k \le \ell-1$ 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. $0 \le k \le \ell-1$ 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，$0 \le k \le \ell_{\mathrm{m}}-1$ について $N'_k \ge N_k$ 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 [12] 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 [24] 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 [28].

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 $j(\ge 0)$ の左から $i(\ge 0)$ 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，マークル木の階層構造による署名方式が提案されている [20]．

この署名方式のマークル木の階層構造の階層数を L とする．この署名方式では，$0 \leq i \leq L-1$ 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，この署名方式は $2^{\sum_{i=0}^{L-1} h_i}$ 個のメッセージに署名できる．

この署名方式で，$0 \leq i \leq L-1$ について，第 i 層のマークル木は $2^{\sum_{j=0}^{i-1} h_j}$ 個存在する．第 0 層（最上層）のマークル木は 1 個であり，その根がこの署名方式の公開鍵となる．したがって，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_0} 個だけ生成すれば良い．$0 \leq i < L-1$ 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 $(L-1)$ 層（最下層）のマークル木は，それぞれ $2^{h_{L-1}}$ 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最下層のマークル木による署名と，$0 \leq i < L-1$ 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 [26] 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．

ビットマスクは，Dahmen ら [9] 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．

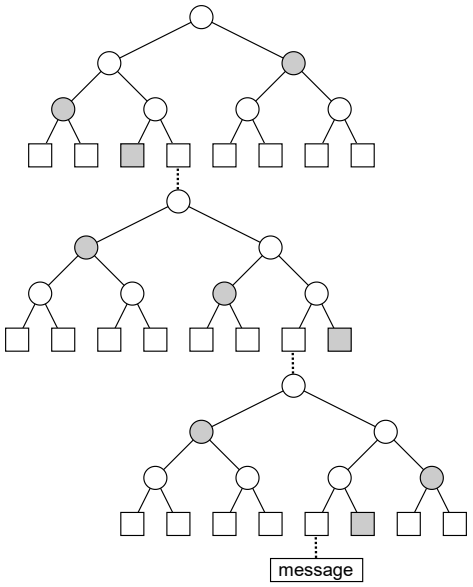


図 7.1: マークル木の階層構造による署名方式

7.3 主要な具体的なハッシュ関数に基づく署名方式

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．

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) [18, 31]			○
SPHINCS$^+$ [1]			○

NIST SP 800-208 [31] 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) [27]
- eXtended Merkle Signature Scheme (XMSS), multi-tree XMSS (XMSS^{MT}) [18]

LMS は Lighton と Micali による署名方式 [26] 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, HSS については，調査報告書で取り上げられている．

SPHINCS$^+$ [1] は 2022 年 7 月に NIST Post Quantum Cryptography Standardization Process で標準化の候補アルゴリズムの一つに選出された．SPHINCS$^+$ はハッシュ関数に基づく stateless な署名方式であり，stateful な方式に求められるような秘密鍵の管理が不要な方式である．SPHINCS$^+$ は SPHINCS [3] の改良版として提案され [4, 5]，その後も NIST Post Quantum Cryptography Standardization Process で改良が行われた．SPHINCS$^+$ 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} とよく似た構造を有することから，以下の SPHINCS$^+$ の

記述では適宜 XMSS を参照する．

7.3.1 XMSS: eXtended Merkle Signature Scheme

XMSS は [7, 20] で提案された方式の改良版 [21] に基づく署名方式であり，Winternitz one-time signature に基づく 1 回署名方式 [17] を用いる．

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 [13] または SHA-3 [15] 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．

$$\begin{aligned}
\mathsf{F}(k, x) &:= \text{SHA-256}([0]_{256} \| k \| x) \\
\mathsf{H}(k, x) &:= \text{SHA-256}([1]_{256} \| k \| x) \\
\mathsf{H}_{\mathrm{msg}}(k, x) &:= \text{SHA-256}([2]_{256} \| k \| x) \\
\mathsf{R}(k, x) &:= \text{SHA-256}([3]_{256} \| k \| x)
\end{aligned}$$

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	(32 bits)
tree address	(64 bits)
type = 0	(32 bits)
OTS address	(32 bits)
chain address	(32 bits)
hash address	(32 bits)
keyAndMask	(32 bits)

(a) OTS ハッシュアドレス

layer address	(32 bits)
tree address	(64 bits)
type = 1	(32 bits)
L-tree address	(32 bits)
tree height	(32 bits)
tree index	(32 bits)
keyAndMask	(32 bits)

(b) L 木アドレス

layer address	(32 bits)
tree address	(64 bits)
type = 2	(32 bits)
Padding = 0	(32 bits)
tree height	(32 bits)
tree index	(32 bits)
keyAndMask	(32 bits)

(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．

■**チェイニング関数**　チェイニング関数 chain の入力は，長さ nBytes 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，

$$Key := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [i+s-1]_{32} \| [0]_{32}), \quad BM := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [i+s-1]_{32} \| [1]_{32})$$

である．なお，ADRS′ は ADRS の上位 24 Bytes であり，例えば，$\mathrm{ADRS}' \| [i+s-1]_{32} \| [0]_{32}$ は図 7.2a の ADRS の hash address, keyAndMask の値をそれぞれ，$[i+s-1]_{32}, [0]_{32}$ 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 $[i]_{32}$ 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

$$Key_j := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [j]_{32} \| [0]_{32}), \quad BM_j := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [j]_{32} \| [1]_{32})$$

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．

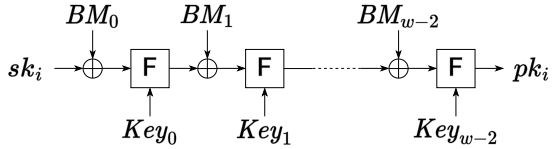


図 7.3: pk_i の計算

■**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ので，詳細は [18] 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，

$$Key := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [0]_{32}), \quad BM_0 := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [1]_{32}), \quad BM_1 := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [2]_{32})$$

である．なお，ADRS$'$ は ADRS の上位 28 Bytes であり，例えば，ADRS$' \| [0]_{32}$ は ADRS の図 7.2 の keyAndMask の値を $[0]_{32}$ とすることを表している．

秘密鍵の生成には [7] 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 $[j]_{32}$，tree index を $[i]_{32}$ 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 $[0]_{32}$，$[0]_{64}$ 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 $[k]_{32}$ 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，上記の 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 $M' := \mathsf{H}_{\mathrm{msg}}(r \| root \| [idx]_{8n}, M)$ とする．ここで，$r := \mathsf{R}(SK_{\mathrm{PRF}}, [idx]_{32})$ 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ので，詳細は [18] 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[*1]．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 $[x]_{32}$，$[y]_{32}$ 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 [18] を参照のこと．

7.3.1.4　パラメータの設定と安全性

Kampanakis と Fluhrer [22] により，LMS と XMSS の比較が論じられている．

Hülsing [21] らは，XMSS について安全性証明を与え，適応的選択メッセージ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 [18] では，上記の結果に基づいて，$n = 32, 64$ 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 [18] 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 [31] と IRTF

[*1] IRTF RFC 8391 [18] では，各層の番号付けが 7.2.3 節の番号付けとは逆順であり，本稿でもそれに従って記述する．

RFC 8391 [18] 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 SPHINCS$^+$

SPHINCS$^+$[1] 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方式である．ただし，XMSS^{MT} とは異なり，stateless な署名方式である．

SPHINCS$^+$ では，以下のような，いくつかの tweakable ハッシュ関数 $\mathbf{T}_\ell$，二つの擬似ランダム関数 $\mathbf{PRF}, \mathbf{PRF_{msg}}$，一つの鍵付きハッシュ関数 $\mathbf{H_{msg}}$ が用いられる．

$$\mathbf{T}_\ell : \mathbb{B}^n \times \mathbb{B}^{32} \times \mathbb{B}^{\ell n} \to \mathbb{B}^n \qquad \mathbf{PRF} : \mathbb{B}^n \times \mathbb{B}^{32} \to \mathbb{B}^n \qquad \mathbf{H_{msg}} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^* \to \mathbb{B}^m$$
$$\mathbf{PRF_{msg}} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^* \to \mathbb{B}^n$$

以下では，$\mathbf{T}_1, \mathbf{T}_2$ について，$\mathbf{F} := \mathbf{T}_1$, $\mathbf{H} := \mathbf{T}_2$ の表記が用いられる．

SPHINCS$^+$ 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．

7.3.2.1 WOTS$^+$

SPHINCS$^+$ でも WOTS$^+$ が用いられているが，XMSS の WOTS$^+$ と以下の点で異なる．

- Winternitz パラメータは $w \in \{4, 16, 256\}$ である．

layer address	(32 bits)
tree address	(96 bits)
type = 0	(32 bits)
key pair address	(32 bits)
chain address	(32 bits)
hash address	(32 bits)

(a) WOTS$^+$ ハッシュアドレス

layer address	(32 bits)
tree address	(96 bits)
type = 1	(32 bits)
key pair address	(32 bits)
$00\cdots0$	(32 bits)
$00\cdots0$	(32 bits)

(b) WOTS$^+$ 公開鍵圧縮アドレス

layer address	(32 bits)
tree address	(96 bits)
type = 2	(32 bits)
$00\cdots0$	(32 bits)
tree height	(32 bits)
tree index	(32 bits)

(c) ハッシュ木アドレス

layer address	(32 bits)
tree address	(96 bits)
type = 3	(32 bits)
key pair address	(32 bits)
tree height	(32 bits)
treen index	(32 bits)

(d) FORS 木アドレス

layer address	(32 bits)
tree address	(96 bits)
type = 4	(32 bits)
key pair address	(32 bits)
$00\cdots0$	(32 bits)
$00\cdots0$	(32 bits)

(e) FORS 木根圧縮アドレス

layer address	(32 bits)
tree address	(96 bits)
type = 5	(32 bits)
key pair address	(32 bits)
chain address	(32 bits)
$00\cdots0$	(32 bits)

(f) WOTS$^+$ 鍵生成アドレス

layer address	(32 bits)
tree address	(96 bits)
type = 6	(32 bits)
key pair address	(32 bits)
$00\cdots0$	(32 bits)
tree index	(32 bits)

(g) FORS 鍵生成アドレス

図 7.4: アドレスの構造

- チェイニング関数は以下のように定義される．

$$\mathsf{chain}(X, i, s, \mathbf{PK}.\mathsf{seed}, \mathbf{ADRS}) := \begin{cases} X & s = 0 \text{ のとき} \\ \mathtt{NULL} & i + s \geq w \text{ のとき} \\ \mathbf{F}(\mathbf{PK}.\mathsf{seed}, \mathbf{ADRS}' \| [i+s-1]_{32}, \mathsf{chain}(X, i, s-1, \mathbf{PK}.\mathsf{seed}, \mathbf{ADRS})) & \text{それ以外のとき} \end{cases}$$

なお，**ADRS** は WOTS$^+$ ハッシュアドレスであり，$\mathbf{ADRS}' \| [i+s-1]_{32}$ は **ADRS** の hash address の値を $[i+s-1]_{32}$ とすることを表している．

■**鍵生成アルゴリズム**　入力は $\mathbf{SK}.\mathsf{seed}$, $\mathbf{PK}.\mathsf{seed}$, WOTS$^+$ ハッシュアドレス **ADRS** である．

1. $0 \leq i \leq \ell - 1$ について，**ADRS** の chain address の値を $[i]_{32}$，hash address の値を $[0]_{32}$ とし，

$$sk_i := \mathbf{PRF}(\mathbf{SK}.\mathsf{seed}, \mathbf{skADRS}) \qquad pk_i := \mathsf{chain}(sk_i, 0, w-1, \mathbf{PK}.\mathsf{seed}, \mathbf{ADRS})$$

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. $pk := \mathbf{T}_\ell(\mathbf{PK}.\mathsf{seed}, \mathbf{wotspkADRS}, pk_0 \| \cdots \| pk_{\ell-1})$ 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 $sk := (sk_0, sk_1, \ldots, sk_{\ell-1})$ である．

■署名アルゴリズム　詳細は [1] を参照のこと．

■検証アルゴリズム　詳細は [1] を参照のこと．

7.3.2.2 HT

SPHINCS$^+$ では，ハイパー木 HT と呼ばれる XMSS^{MT} と同様の XMSS 木の階層構造が用いられる．HT が d 層の XMSS 木からなるとき，第 $(d-1)$ 層と第 0 層はそれぞれ，HT 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．HT の各 XMSS 木の各葉は，7.3.2.1 節の WOTS$^+$ の公開鍵である．各 XMSS 木の葉以外の節点の計算にはハッシュ関数 **H** 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 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 $[x]_{32}$，$[y]_{96}$ である．

HT の鍵生成，署名，検証の各アルゴリズムについての詳細は [1] を参照のこと．

7.3.2.3 FORS (Forest Of Random Subsets)

SPHINCS$^+$ では，メッセージの署名に WOTS$^+$ ではなく，FORS と呼ばれる方式が用いられる．FORS では一組の公開鍵と秘密鍵を用いて複数個のメッセージに署名できる．FORS は，数回（few-time）署名方式 HORS [34] に基づく HORST [3] 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 $\mathbf{SK}.\mathsf{seed}$, $\mathbf{PK}.\mathsf{seed}$, FORS 木アドレス **ADRS** である．

1. $0 \le i < kt$ について，

$$sk_i := \mathbf{PRF}(\mathbf{SK}.\mathsf{seed}, \mathbf{skADRS}) \qquad \mathrm{Node}_{i,0} := \mathbf{F}(\mathbf{PK}.\mathsf{seed}, \mathbf{ADRS}, sk_i)$$

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 $[i]_{32}$ である．また，**ADRS** の tree index の値は $[i]_{32}$ である．

2. $1 \le j \le a$ について，それぞれ，$0 \le i < kt/2^j$ について，

$$\mathrm{Node}_{i,j} := \mathbf{H}(\mathbf{PK}.\mathsf{seed}, \mathbf{ADRS}, \mathrm{Node}_{2i,j-1} \| \mathrm{Node}_{2i+1,j-1})$$

とする．ここで，**ADRS** の tree height の値は $[j]_{32}$，tree index の値は $[i]_{32}$ である．

3. $pk := \mathbf{T}_k(\mathbf{PK}.\mathsf{seed}, \mathbf{forspkADRS}, \mathrm{Node}_{0,a} \| \cdots \| \mathrm{Node}_{k-1,a})$ 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，$\mathrm{Node}_{0,a}, \mathrm{Node}_{1,a}, \ldots, \mathrm{Node}_{k-1,a}$ 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 ka bits のメッセージ M をそれぞれ長さ a bits の k 個のブロック $M_0, M_1, \ldots, M_{k-1}$ に分割する．すなわち，$M = M_0 \| M_1 \| \cdots \| M_{k-1}$ である．さらに，M_i を 2 進数表記の非負整数とみなす．M の署名は $sk_{0\cdot t+M_0}, sk_{1\cdot t+M_1}, \ldots, sk_{(k-1)t+M_{k-1}}$ と，$0 \leq i < k$ について，$\mathrm{Node}_{i,a}$ を根とするマークル木の $\mathrm{Node}_{it+M_i,0}$ の認証パスである．

■**検証アルゴリズム**　詳細は [1] を参照のこと．

7.3.2.4　SPHINCS$^+$

前節までの構成要素を用いて SPHINCS$^+$ の署名が構成される．SPHINCS$^+$ 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）
- Winternitz パラメータ w
- ハイパー木の高さ h と階層数 d
- FORS の木の個数 k と各木の葉の個数 t

メッセージダイジェストの Byte 長は $m := \lfloor (k\log_2 t + 7)/8 \rfloor + \lfloor (h - h/d + 7)/8 \rfloor + \lfloor (h/d + 7)/8 \rfloor$ となる．

■**鍵生成アルゴリズム**　$\mathbf{SK}.\mathtt{seed}, \mathbf{SK}.\mathtt{prf} \in \mathbb{B}^n$ はいずれも無作為に選択される．$\mathbf{PK}.\mathtt{seed} \in \mathbb{B}^n$ は無作為に選択される．$\mathbf{PK}.\mathtt{root} \in \mathbb{B}^n$ は HT の第 $(d-1)$ 層の XMSS 木の根である．秘密鍵は $\mathbf{SK}.\mathtt{seed}, \mathbf{SK}.\mathtt{prf}, \mathbf{PK}.\mathtt{seed}, \mathbf{PK}.\mathtt{root}$ である．公開鍵は $\mathbf{PK}.\mathtt{seed}, \mathbf{PK}.\mathtt{root}$ 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. $\mathbf{R} := \mathbf{PRF_{msg}}(\mathbf{SK}.\mathtt{prf}, opt, M)$ とする．$opt = \mathbf{PK}.\mathtt{seed}$ であるが，opt を乱数とするオプションも用意されている．
2. $digest := \mathbf{H_{msg}}(\mathbf{R}, \mathbf{PK}.\mathtt{seed}, \mathbf{PK}.\mathtt{root}, M)$ とする．$digest$ の最初の $\lfloor (ka+7)/8 \rfloor$ Bytes，次の $\lfloor (h - h/d + 7)/8 \rfloor$ Bytes，その次の $\lfloor (h/d + 7)/8 \rfloor$ Bytes をそれぞれ tmp_0, tmp_1, tmp_2 とする．さらに，tmp_0 の先頭 ka bits を md，tmp_1 の先頭 $(h - h/d)$ bits を idx_{tree}，tmp_2 の先頭 h/d bits を idx_{leaf} とする．
3. HT 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 の署名を生成する．このとき，FORS の **ADRS** の layer address は $[0]_{32}$，tree address は idx_{tree}，key pair address は idx_{leaf} である．さらに，tree height, tree index はともに $[0]_{32}$ である．
4. 上の署名で用いられた FORS の公開鍵への HT による署名を生成する．

M の署名は $\mathbf{R}$，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 HT による署名からなる．

■**検証アルゴリズム**　詳細については [1] を参照のこと．

SPHINCS$^+$ の秘密鍵，公開鍵，署名のサイズはそれぞれ，$4n$ Bytes，$2n$ Bytes，$(h + k(\log_2 t + 1) + d\ell + 1)n$ Bytes である．ここで，$\ell := \ell_1 + \ell_2$ であり，$\ell_1 := \lceil 8n/\log_2 w \rceil$, $\ell_2 := \lfloor (\log_2(\ell_1(w-1)))/\log_2 w \rfloor + 1$ である．

7.3.2.5　パラメータの設定と安全性

Hülsing と Kudinov [19] は，SPHINCS$^+$ が適応的選択メッセージ攻撃に対する存在偽造不能性（EUF-CMA）を満たすことを tweakable ハッシュ関数 $\mathbf{T}_\ell$，鍵付きハッシュ関数 $\mathbf{H_{msg}}$，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．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．

SPHINCS$^+$ については，表 7.5 のパラメータセットが示されている．この表の最左欄のラベルの s と f はそれぞれ，署名長，計算時間について最適化されたパラメータセットであることを示している．ただし，一方の最適化で他方が非実用的にならないよう配慮されている．また，安全性レベルは NIST Post Quantum Cryptography Standardization Process の Call for Proposals に記された安全性強度のカテゴリである．なお，Haraka を用いる場合については，表 7.5 とは異なり，$n = 24, 32$ のときの安全性はレベル 2 とされており，したがって，bit 安全性はおよそ 128 程度となる．Haraka の安全性について，Haraka-512 が（第二）原像攻撃に対して 256 bit 安全性を有しないことが示されている [2] が，これは Haraka を用いた SPHINCS$^+$ の安全性がレベル 2 であることを否定するものではない．

表 7.5: SPHINCS$^+$ のパラメータセットの例．署名長の単位は Byte である．

名称	n	h	d	$\log_2 t$	k	w	bit 安全性	安全性レベル	署名長
SPHINCS$^+$-128s	16	63	7	12	14	16	133	レベル 1	7,856
SPHINCS$^+$-128f	16	66	22	6	33	16	128	レベル 1	17,088
SPHINCS$^+$-192s	24	63	7	14	17	16	193	レベル 3	16,224
SPHINCS$^+$-192f	24	66	22	8	33	16	194	レベル 3	35,664
SPHINCS$^+$-256s	32	64	8	14	22	16	255	レベル 5	29,792
SPHINCS$^+$-256f	32	68	17	9	35	16	255	レベル 5	49,856

7.3.2.6　ハッシュ関数の実現法

SPHINCS$^+$ のハッシュ関数はすべて，SHAKE256，SHA-2，Haraka のうちのいずれかを用いて定義される．なお，tweakable ハッシュ関数については robust と simple の二つの実現が示されている．robust な実現では 7.2.4 節で述べられたビットマスクが用いられるが，simple な実現では用いられない．

SHA-2 を用いた実現では，当初は 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 [32] が示されたため，$n = 24, 32$ については，一部の関数が SHA-512 を用いて実現されることとなった．

SHAKE256 を用いた構成は以下のとおりである．

$$\mathbf{H_{msg}}(\mathbf{R}, \mathbf{PK}.\mathtt{seed}, \mathbf{PK}.\mathtt{root}, M) := \mathrm{SHAKE256}(\mathbf{R}\|\mathbf{PK}.\mathtt{seed}\|\mathbf{PK}.\mathtt{root}\|M, 8m)$$
$$\mathbf{PRF}(\mathbf{PK}.\mathtt{seed}, \mathbf{SK}.\mathtt{seed}, \mathbf{ADRS}) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|\mathbf{SK}.\mathtt{seed}, 8n)$$
$$\mathbf{PRF_{msg}}(\mathbf{SK}.\mathtt{prf}, \mathtt{OptRand}, M) := \mathrm{SHAKE256}(\mathbf{SK}.\mathtt{prf}\|\mathtt{OptRand}\|M, 8n)$$

robust な実現では

$$\mathbf{F}(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M_1) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M_1^{\oplus}, 8n)$$
$$\mathbf{H}(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M_1\|M_2) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M_1^{\oplus}\|M_2^{\oplus}, 8n)$$
$$\mathbf{T}_\ell(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M^{\oplus}, 8n)$$

simple な実現では

$$\mathbf{F}(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M_1) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M_1, 8n)$$
$$\mathbf{H}(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M_1\|M_2) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M_1\|M_2, 8n)$$
$$\mathbf{T}_\ell(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M, 8n)$$

である．ここで，$M \in \{0,1\}^l$ のとき，$M^{\oplus} := M \oplus \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}, l)$ である．

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, SPHINCS$^+$ 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 [18] は NIST の推奨アルゴリズムであり [31]，SPHINCS$^+$[1] は NIST Post Quantum Cryptography Standardization Process で標準化の候補アルゴリズムの一つに選出された．

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, SPHINCS$^+$ 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．XMSS は stateful な署名方式であるが，それを推奨アルゴリズムとする NIST SP 800-208 [31] 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．

SPHINCS$^+$ 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] J.-P. Aumasson, D. J. Bernstein, W. Beullens, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, A. Hülsing, P. Kampanakis, S. Kölbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe, and B. Westerbaan. SPHINCS$^+$. Submission to the NIST post-quantum project, v.3.1, 2022. `https://sphincs.org/resources.html`.

[2] Z. Bao, X. Dong, J. Guo, Z. Li, D. Shi, S. Sun, and X. Wang. Automatic search of meet-in-the-middle preimage attacks on AES-like hashing. In A. Canteaut and F. Standaert, editors, *EUROCRYPT 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 771–804. Springer, 2021.

[3] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, and Z. Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 368–397. Springer, 2015.

[4] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe. The SPHINCS$^+$ signature framework. Cryptology ePrint Archive, Report 2019/1086, 2019. `https://ia.cr/2019/1086`.

[5] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe. The SPHINCS$^+$ signature framework. In L. Cavallaro, J. Kinder, X. Wang, and J. Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019*, pages 2129–2146. ACM, 2019.

[6] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Sponge functions. ECRYPT Hash Workshop, 2007.

[7] J. Buchmann, E. Dahmen, and A. Hülsing. XMSS - A practical forward secure signature scheme based on minimal security assumptions. In B. Yang, editor, *PQCrypto 2011, Proceedings*, volume 7071 of *Lecture Notes in Computer Science*, pages 117–129. Springer, 2011.

[8] J. Daemen and V. Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[9] E. Dahmen, K. Okeya, T. Takagi, and C. Vuillaume. Digital signatures out of second-preimage resistant hash functions. In J. Buchmann and J. Ding, editors, *PQCrypto 2008, Proceedings*, volume 5299 of *Lecture Notes in Computer Science*, pages 109–123. Springer, 2008.

[10] I. Damgård. A design principle for hash functions. In G. Brassard, editor, *CRYPTO '89, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.

[11] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[12] C. Dods, N. P. Smart, and M. Stam. Hash based digital signature schemes. In N. P. Smart, editor,

Cryptography and Coding, 10th IMA International Conference, Proceedings, volume 3796 of *Lecture Notes in Computer Science*, pages 96–115. Springer, 2005.

[13] FIPS PUB 180-4. Secure hash standard (SHS), Aug. 2015.

[14] FIPS PUB 197. Advanced encryption standard (AES), 2001.

[15] FIPS PUB 202. SHA-3 standard: Permutation-based hash and extendable-output functions, 2015.

[16] L. K. Grover. A fast quantum mechanical algorithm for database search. In G. L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 212–219. ACM, 1996.

[17] A. Hülsing. W-OTS$^+$ - shorter signatures for hash-based signature schemes. In A. M. Youssef, A. Nitaj, and A. E. Hassanien, editors, *AFRICACRYPT 2013, Proceedings*, volume 7918 of *Lecture Notes in Computer Science*, pages 173–188. Springer, 2013.

[18] A. Hülsing, D. Butin, S. Gazdag, J. Rijneveld, and A. Mohaisen. XMSS: eXtended Merkle Signature Scheme. IRTF RFC 8391, 2018.

[19] A. Hülsing and M. Kudinov. Recovering the tight security proof of SPHINCS$^+$. Cryptology ePrint Archive, Paper 2022/346, 2022. `https://eprint.iacr.org/2022/346`.

[20] A. Hülsing, L. Rausch, and J. Buchmann. Optimal parameters for XMSS^{MT}. In A. Cuzzocrea, C. Kittl, D. E. Simos, E. R. Weippl, and L. Xu, editors, *Security Engineering and Intelligence Informatics - CD-ARES 2013 Workshops: MoCrySEn and SeCIHD, Proceedings*, volume 8128 of *Lecture Notes in Computer Science*, pages 194–208. Springer, 2013.

[21] A. Hülsing, J. Rijneveld, and F. Song. Mitigating multi-target attacks in hash-based signatures. In C. Cheng, K. Chung, G. Persiano, and B. Yang, editors, *PKC 2016, Proceedings, Part I*, volume 9614 of *Lecture Notes in Computer Science*, pages 387–416. Springer, 2016.

[22] P. Kampanakis and 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`.

[23] S. Kölbl, M. M. Lauridsen, F. Mendel, and C. Rechberger. Haraka v2 - efficient short-input hashing for post-quantum applications. *IACR Trans. Symmetric Cryptol.*, 2016(2):1–29, 2016.

[24] P. Lafrance and A. Menezes. On the security of the WOTS-PRF signature scheme. *Advances in Mathematics of Communications*, 13(1):185–193, 2019.

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

[26] F. T. Leighton and S. Micali. Large provably fast and secure digital signature schemes based on secure hash functions. US Patent 5,432,852, 1995.

[27] D. A. McGrew, M. Curcio, and S. R. Fluhrer. Leighton-Micali hash-based signatures. IRTF RFC 8554, 2019.

[28] R. C. Merkle. *Secrecy, Authentication, and Public Key Systems*. PhD thesis, Stanford University, 1979. `https://www.merkle.com/papers/Thesis1979.pdf`.

[29] R. C. Merkle. A certified digital signature. In G. Brassard, editor, *CRYPTO '89, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238. Springer, 1989.

[30] R. C. Merkle. One way hash functions and DES. In G. Brassard, editor, *CRYPTO '89, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.

[31] NIST Special Publication 800-208. Recommendation for stateful hash-based signature schemes, 2020.

[32] R. A. Perlner, J. Kelsey, and D. A. Cooper. Breaking category five SPHINCS$^+$ with SHA-256. In J. H. Cheon and T. Johansson, editors, *PQCrypto 2022, Proceedings*, volume 13512 of *Lecture Notes in Computer Science*, pages 501–522. Springer, 2022.

[33] J. Quisquater and M. Girault. $2n$-bit hash-functions using n-bit symmetric block cipher algorithms. In J. Quisquater and J. Vandewalle, editors, *EUROCRYPT '89, Proceedings*, volume 434 of *Lecture Notes in Computer Science*, pages 102–109. Springer, 1989.

[34] L. Reyzin and N. Reyzin. Better than BiBa: Short one-time signatures with fast signing and verifying. In L. M. Batten and J. Seberry, editors, *ACISP 2002, Proceedings*, volume 2384 of *Lecture Notes in Computer Science*, pages 144–153. Springer, 2002.

CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）
[CRYPTREC GL-2004-2023]

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
  (技術本部　セキュリティセンター　暗号グループ)
  INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN
  2-28-8 HONKOMAGOME, BUNKYO-KU
  TOKYO, 113-6591 JAPAN

資料３－６

2022 年度暗号技術調査ＷＧ（高機能暗号）活動報告

1. 暗号技術調査ＷＧ（高機能暗号ＷＧ）の活動目的

高機能暗号の研究動向調査をもとに、高機能暗号ガイドラインを 2021 年度、2022 年度において作成する。

2. 委員構成（敬称略）

主査	四方 順司	横浜国立大学 教授
委員	岩本 貢	電気通信大学 教授
委員	大原 一真	国立研究開発法人産業技術総合研究所 主任研究員
委員	勝又 秀一	PQShield Lead Cryptography Researcher
委員	金岡 晃	東邦大学 准教授
委員	川原 祐人	日本電信電話株式会社 主任研究員
委員	国井 裕樹	セコム株式会社 グループリーダー
委員	須賀 祐治	株式会社インターネットイニシアティブ シニアエンジニア
委員	鈴木 幸太郎	豊橋技術科学大学 教授
委員	花岡 悟一郎	国立研究開発法人産業技術総合研究所 首席研究員
委員	濱田 浩気	日本電信電話株式会社 主任研究員
委員	外園 康智	野村総合研究所 上級研究員
委員	山田 翔太	国立研究開発法人産業技術総合研究所 主任研究員
委員	米山 一樹	茨城大学 教授
委員	渡邉 洋平	電気通信大学 助教

3. 2022 年度暗号技術調査ＷＧ（高機能暗号）活動報告の概要

3.1. 背景

公開鍵暗号は、アプリケーションが多様となりその活用が広まっている。その中で、従来の公開鍵暗号よりも機能が向上した高機能暗号を利用してアプリケーションに適用することが有効と考えられている。そこで、2020 年度第 2 回暗号技術検討会において高機能暗号ガイドラインを作成するために高機能暗号ＷＧを設置することが承認された。

3.2. 2021 年度活動概要

検討会により高機能暗号ＷＧを設置することが承認されたことをうけ、2021 年度より高機能暗号ＷＧが設置され、2021 年 7 月 6 日に開催された 2021 年度第 1 回暗号技術評価委員会において、2021 年度の高機能暗号ＷＧの活動として下記 4 点について実施する活動計画が承認された。

(1) 2021-2022 年度において高機能暗号ガイドラインを作成すること
(2) 高機能暗号のスコープの明確化
(3) 高機能暗号技術に関する現状調査

(4) 高機能暗号のアプリケーションに関する調査

この活動計画に沿い、2021 年 8 月 3 日に第 1 回 WG、2021 年 12 月 8 日に第 2 回 WG、2022 年 2 月 8 日に第 3 回 WG の年 3 回の高機能暗号WGを開催した。これらの活動によりガイドラインに記載する高機能暗号、および、目次案が決定された。その概要については、2022 年 2 月 22 日に開催された 2021 年度第 2 回暗号技術評価委員会において報告した。

① 第 1 回高機能暗号WG（2021 年 8 月 3 日）
- （ア） 高機能暗号のスコープの議論
- （イ） 高機能暗号技術に関する現状調査について作業方針・分担を議論
- （ウ） 高機能暗号のアプリケーションに関する現状調査について作業方針・分担を議論
- （エ） 高機能暗号のアプリケーションについて、エンドユーザのヒアリング先の検討

② 第 2 回高機能暗号WG（2021 年 12 月 8 日）
- （ア） 現状調査・アプリケーションに関する中間報告
- （イ） ヒアリングに関する中間報告
- （ウ） ガイドラインの目次案についての議論

③ 第 3 回高機能暗号WG（2022 年 2 月 8 日）
- （ア） 2021 年度高機能暗号WG報告資料の確認
- （イ） 2021 年度調査内容の確認
- （ウ） ガイドラインの執筆方針に関する確認
- （エ） 2022 年度の検討項目の抽出および方針決定

3.3. 2022 年度活動計画概要

2022 年 4 月に行われた暗号技術評価委員会のメール審議により、2022 年度の高機能暗号WGの活動として下記２点について実施する活動計画が承認された。

(1) 2021 年度の作成した目次案に沿ったガイドラインの案の執筆
(2) 高機能暗号のアプリケーションに関するヒアリング調査および調査内容のガイドラインの案への反映

また、高機能暗号が多岐にわたることがあり、新たに 8 名の委員がWGの委員として高木委員長より指名された。

3.4. 2022 年度活動概要

2022 年度活動計画に沿い、年 3 回の高機能暗号WGを開催した。

(1) 第 1 回高機能暗号WG（2022 年 6 月 15 日）
- （ア） 高機能暗号の技術内容に関する執筆分担の議論
- （イ） 高機能暗号のアプリケーションについて、ＮＥＣ様へのヒアリング

(2) 第 2 回高機能暗号WG（2022 年 11 月 9 日）
- （ア） ガイドラインの案の執筆状況の中間報告および内容確認・議論
- （イ） 高機能暗号のアプリケーションについて三菱電機様へのヒアリング

(3) 第 3 回高機能暗号WG（2023 年 2 月 10 日）
- （ア） 高機能暗号ガイドラインの案の内容の最終確認

第 3 回 WG でのコメント修正版を 2 月 17 日に完成し､WG でのガイドラインの案の完成版とした。

4. 高機能暗号ガイドラインの案の執筆

2021 年度の WG の活動により、高機能暗号を“守秘”、“認証・署名”、“その他”（表１の第１列参照）に分類した。そして、ガイドラインに記載すべき高機能暗号の対象を

守秘： ID ベース暗号、属性ベース暗号、放送型暗号、準同型暗号、プロキシ再暗号化

認証・署名：ID ベース署名、集約署名・MAC・マルチ署名、グループ署名、リング署名、しきい値署名

その他：秘密分散、マルチパーティ計算－秘密分散ベース、マルチパーティ計算－Garbled Circuit ベース、ゼロ知識証明、検索可能暗号、Private Information Retrieval、Oblivious RAM

の１７項目として、執筆を分担した。執筆分担を表１に示す。

表１　作業（執筆）分担表（敬称略）

（各章のタイトルは略称）

章			章タイトル	執筆担当
第１章			はじめに	四方、事務局
第２章	2.1		高機能暗号とは	四方、事務局
	2.2		どこに使えるか、その有用性	四方、事務局
	2.3		種類と分類	四方、事務局
	2.4	2.4.1	守秘関連の活用事例と標準化	四方、金岡、国井、花岡、外園、米山、事務局
		2.4.2	認証・署名関連の活用事例	四方、国井、須賀、米山、事務局
		2.4.3	その他の活用事例	金岡、国井、須賀、花岡、外園、米山、事務局
		2.4.4	活用事例からみた利用方法	外園、事務局
第３章	3.1	3.1.1	ID ベース暗号	金岡、渡邉、四方
		3.1.2	属性ベース暗号	川原、山田、四方
		3.1.3	放送型暗号	山田、花岡
		3.1.4	準同型暗号	山田、四方
		3.1.5	プロキシ再暗号化	花岡、勝又
	3.2	3.2.1	属性ベース署名	勝又、四方
		3.2.2	集約署名／MAC、マルチ署名	勝又、花岡、四方
		3.2.3	グループ署名	米山、鈴木
		3.2.4	リング署名	鈴木、米山
		3.2.5	閾値署名	国井、米山
	3.3	3.3.1	秘密分散	花岡、大原、岩本
		3.3.2	MPC-秘密分散ベース	濱田、大原、花岡
		3.3.3	MPC-Garbled Circuit ベース	大原、花岡
		3.3.4	ゼロ知識証明	外園、川原
		3.3.5	Oblivious RAM	米山、濱田
		3.3.6	PIR	岩本、須賀
		3.3.7	検索可能暗号	渡邉、米山、岩本
第４章			おわりに	四方、事務局

5. 高機能暗号のアプリケーションに関するヒアリング調査

ＮＥＣ様と三菱電機様にＷＧに参加していただき、ヒアリングを実施した。ヒアリング内容については、ガイドラインの案の“2.4.4 章活用事例から見た高機能暗号の利用方法”に反映させた。

5.1. ＮＥＣ様ヒアリング（2022 年 6 月 15 日）

個々のデータのプライバシを強化するために、秘密計算を利用する事例、ゲノム解析、創薬等の報告をいただき、質疑を行った。

5.2. 三菱電機様ヒアリング（2022 年 11 月 9 日）

企業が持つデータベースのデータを共有するにあたり、パブリッククラウドを使用してコストを削減しつつ、不要なデータが他企業に渡る（漏洩する）ことを防ぐために、検索可能暗号や属性ベース暗号を利用する事例、組織内文書管理、委託作業等の報告をいただき、質疑を行った。

6. 今後の予定（ガイドライン発行に向けた手順）

ＷＧで作成したガイドラインの案を本日の資料３－７別紙として配布。

2023 年 3 月 31 日までに第 1 版高機能暗号ガイドラインを完成させ、四月上旬に CRYPTREC ホームページで公開する。

以上

資料3－7

CRYPTREC 暗号技術ガイドライン（高機能暗号）

CRYPTREC 暗号技術調査ワーキンググループ
（高機能暗号）

2023 年 3 月

表１　CRYPTREC 高機能暗号 WG 委員構成

主査	四方 順司	横浜国立大学
委員	岩本 貢	電気通信大学
委員	大原 一真	国立研究開発法人産業技術総合研究所
委員	勝又 秀一	ＰＱＳｈｉｅｌｄ　Ｌｔｄ．／国立研究開発法人産業技術総合研究所
委員	金岡 晃	東邦大学
委員	川原 祐人	日本電信電話株式会社
委員	国井 裕樹	セコム株式会社
委員	須賀 祐治	株式会社インターネットイニシアティブ
委員	鈴木 幸太郎	豊橋技術科学大学
委員	花岡 悟一郎	国立研究開発法人産業技術総合研究所
委員	濱田 浩気	日本電信電話株式会社
委員	外園 康智	株式会社野村総合研究所
委員	山田 翔太	国立研究開発法人産業技術総合研究所
委員	米山 一樹	茨城大学
委員	渡邉 洋平	電気通信大学

目次

第 1 章

はじめに

ネットワークの高速化、広帯域化、さらには、すさまじい勢いでのコンピュータの高性能化に伴い、サイバー空間とフィジカル空間が融合したいわゆる Society5.0 の世界が現実的なものになりつつある。特に、クラウドサービスの充実、人工知能（Artificial Intelligence，AI）技術の進歩等もあいまって、多くの分野でオンラインサービスの需要が高まっている。また、暗号資産の出現により、経済も変化しつつある。さらに、2020 年に訪れたコロナ禍においては、すべての対応をオンライン化しなければならない状況が生じ、在宅勤務を含むリモートワークや、リモート会議等のオンラインサービス、それに伴う技術が著しく進歩し、いわゆるニューノーマルな社会を実現するための環境作りが進んできた。

以前より、これらの技術の進歩、社会の変化に伴った現状のサービスの改善や、将来のサービス変化を見越して、既存の暗号技術よりも効率的で高機能な方式の研究開発が進められている。従来技術より、低コスト、高機能の特徴をもつ「高機能暗号」は、ゲノムのような重要な個人情報を扱う医療データの活用、AI 用の秘密計算、DB の秘匿検索をはじめとするさまざまな用途での利用が期待されている。しかし、どの高機能暗号を選べばよいのか、選ばれた高機能暗号方式は従来の暗号方式よりもどのような点で優れているのか、運用時にはどのようなことに注意することが必要か等、実際に利用する際には専門家以外では、判断基準がわからず、実運用の判断に困難な場合も多い。

CRYPTREC では、主として電子政府で利用する暗号技術について検討を行っているが、それに加えて、今後さまざまな領域で利用が想定される暗号技術について技術調査を行い、社会に役立つ形で情報提供を行うことを目指している。特に、高機能暗号技術が求められる製品やサービスにおいて、利用者が適切な暗号方式を選択でき、容易に調達できることを目指し、2021 年度より CRYPTREC 暗号技術評価委員会の下に、“暗号技術調査ワーキンググループ（高機能暗号）”（以下、

高機能暗号 WG）が設置された。本ガイドラインは、高機能暗号の方式を選択・利用する際の技術的判断に資すること、今後の利用促進を図ることを目的として、高機能暗号 WG が作成したものである。主たる読者として、セキュリティ技術の標準化等を目指すコンソーシアムや団体、情報システムのセキュリティ機能の設計・開発・実装において暗号技術を活用する技術者や、セキュリティ機能を搭載した情報システムの導入を推進する企業等を想定しているが、高機能暗号技術に興味を持つ方に広く読んで頂ければ幸いである。

1 章は、本ガイドラインの総説である。2 章では、高機能暗号の概説をまとめている。まず、対象とする高機能暗号と、その活用例を示し、その上で、高機能暗号を実際に活用する際の手引きを示している。特に、高機能暗号の特徴、代表的なユースケース、方式の選択方法等を記載している。3 章では、代表的な高機能暗号のアルゴリズムと性能等を示している。多くの高機能暗号方式が提案されている中、優位な特徴がある方式を取り上げ、アルゴリズムを紹介するとともに、その優位性を示している。

本ガイドラインを読むにあたり、どのような暗号があるかを知りたい読者は、第 1 章、第 2 章までを読み、より暗号技術の詳細内容まで知りたい読者は、第 3 章まで読むことをお勧めする。

本ガイドラインで紹介している高機能暗号技術は、執筆時点までに、主要国際学会で発表されており、有力な攻撃法が発見されておらず、かつ、十分な性能を持つと考えられる方式を選んでいる。このため、本ガイドラインで紹介する高機能暗号は、CRYPTREC 暗号リストの暗号方式とは異なり、CRYPTREC において安全性評価を行った方式ではなく、国際会議等の発表により脆弱性が発見されていないことを安全性の根拠としている。そして、可能な限り最新の情報に基づき安全性や、性能比較、標準化動向等を紹介している。しかしながら、高機能暗号の研究開発は今まさに盛んに行われており、年々新たな方式や評価結果が出ているところであり、記載内容が執筆時点のものであることに留意いただきたい。また、本ガイドラインで紹介する高機能暗号は、その機能の優位な特徴を明確に示す暗号を掲載している。このため、アプリケーションに本ガイドラインに記載されている暗号方式を実装することで、その暗号方式が有する優位な特徴を利用することはできるものの、必ずしもそのアプリケーションにとって最適な方式とは限らないケースもある。この場合、最適な方式を選択するためには、暗号方式の詳細を理解し、改良したり、別の暗号方式を利用する等の検討が必要となることがあることをご留意いただきたい。

本ガイドラインは表 1.1 に示す高機能暗号 WG 委員および CRYPTTREC 事務

局で執筆・編集を行った。所属は、WG 委員、事務局を担当した期間におけるものである。また、本ガイドラインを作成する上で、活用事例からみた高機能暗号の利用方法において、日本電気株式会社の横田治樹氏、藤井了氏、三菱電機株式会社の平野貴人氏、川合豊氏、反町亨氏、早坂健一郎氏に多くの有益な情報をご提供いただいた。

表 1.1　高機能暗号ガイドライン執筆者リスト

主査	四方 順司	横浜国立大学
委員	岩本 貢	電気通信大学
委員	大原 一真	国立研究開発法人産業技術総合研究所
委員	勝又 秀一	PQShield Ltd. ／国立研究開発法人産業技術総合研究所
委員	金岡 晃	東邦大学
委員	川原 祐人	日本電信電話株式会社
委員	国井 裕樹	セコム株式会社
委員	須賀 祐治	株式会社インターネットイニシアティブ
委員	鈴木 幸太郎	豊橋技術科学大学
委員	花岡 悟一郎	国立研究開発法人産業技術総合研究所
委員	濱田 浩気	日本電信電話株式会社
委員	外園 康智	株式会社野村総合研究所
委員	山田 翔太	国立研究開発法人産業技術総合研究所
委員	米山 一樹	茨城大学
委員	渡邉 洋平	電気通信大学
事務局	野島 良	国立研究開発法人情報通信研究機構
事務局	青野 良範	国立研究開発法人情報通信研究機構
事務局	伊藤 竜馬	国立研究開発法人情報通信研究機構
事務局	大久保 美也子	国立研究開発法人情報通信研究機構
事務局	金森 祥子	国立研究開発法人情報通信研究機構
事務局	黒川 貴司	国立研究開発法人情報通信研究機構
事務局	篠原 直行	国立研究開発法人情報通信研究機構
事務局	吉田 真紀	国立研究開発法人情報通信研究機構
事務局	小川 一人	国立研究開発法人情報通信研究機構

第 2 章

高機能暗号技術とその活用法

2.1 高機能暗号とは

近年、従来の守秘、認証、署名の機能だけではなく、様々な機能を持つ「高機能暗号」（Advanced Cryptography）の研究開発が進んでおり、学会等で提案されている。高機能暗号は日本が強みを持つ分野であり、日本での研究開発が盛んである。そして、日本発の高機能暗号が、多くの著名な国際会議で発表されている。また、ISO/IEC 等において標準化が進められている高機能暗号技術もある。

様々な機能を有する高機能暗号技術は、今後の発展が予想される 5G および Beyond5G の社会における種々のサービスで利用できる可能性があり、Internet of Things（IoT）や AI にとどまらず、Society5.0 における基盤となる Cyber Phisical System（CPS）といった次世代のネットワークサービスを構築する上でも有効なセキュリティ技術の一つとなることが期待されている。

一方で、これまで提案されてきた高機能暗号技術には、様々な機能を有する方式が存在する。機能に応じて、それぞれ暗号技術が発展しており、それらを一つの指標で比較することは困難である。このため、国際的に「高機能暗号」に対して一般的に合意されている定義はない。ただし、ガイドラインを読み進めるにあたり、高機能暗号を定義しておくことは読者の理解の支えになると考えられる。以上の状況を鑑み、本ガイドラインでは、

「従来の暗号技術に対して、機能が追加・向上される等の優位性を主張する暗号技術、および、従来の暗号技術では困難であった事象を解決できる等の新規機能を有することを主張する暗号技術」

をスコープとし、従来の暗号技術に対して、なんらかの機能・性能で優位性（高機能性）を持つ暗号技術を高機能暗号における主な対象とする。ここで述べた従来の暗号技術とは、CRYPTREC 暗号リスト [1] に挙げられているような暗号基礎技術を指す。そして、この高機能性については、各項目において明示することにする。

2.2 高機能暗号はどこに使えるか、その有用性

2.2.1 CRYPTREC 暗号リストの暗号方式との違い

CRYPTREC 暗号リスト [1] には、公開鍵暗号、共通鍵暗号、ハッシュ関数、暗号利用モード、メッセージ認証コード、認証暗号、エンティティ認証が記載されて

いる。さらに、公開鍵暗号は、署名、守秘、鍵共有に分かれている。また、共通鍵暗号は、64 ビットブロック暗号、128 ビットブロック暗号、ストリーム暗号に分かれ、暗号利用モードは、秘匿モード、認証付き秘匿モードに分かれている。

本ガイドラインで紹介する高機能暗号方式は、これらの暗号方式単独では実現できない機能を持つ暗号方式、もしくは、これらの暗号方式で実現するには非効率である事象を効率的に実現する暗号方式である。前者の例としては、CRYPTREC 暗号リストにおける公開鍵暗号では鍵生成においてランダムに出力される値が公開鍵として使用される。これに対し、任意の記号列、例えば、人名やメールアドレス、を公開鍵として利用することを可能とした高機能暗号が開発されている (第 3.1.1 章 ID ベース暗号参照)。また、後者の例としては、一つの平文を複数の利用者に送る際に、復号権限のある利用者の人数分の暗号文を生成し全利用者に暗号文を送ることに比べて、さらにコンパクトな暗号文のサイズで同じ機能を達成する高機能暗号も開発されている（第 3.1.3 章放送型暗号参照)。

第 1 章でも述べたように、本ガイドラインで紹介する高機能暗号方式は、CRYPTREC により安全性評価を行った方式ではない。これは、高機能暗号は発展途上にあるが、できる限り最新の情報を提供することを目指したためである。発展途上にある最新の技術を CRYPTREC で評価をするためには、多大な時間を要することになり、逆に、最新の技術が掲載できなくなる。とはいえ、ガイドラインに掲載するためには、安全性が保障されない技術を記載するわけにはゆかない。そこで、主要な国際会議等の発表により脆弱性が発見されていないことを安全性の根拠とし、世界の研究者を評価者と考えることで、安全性を担保する方針とした。

2.2.2 高機能暗号の有用性

本ガイドラインで扱う高機能暗号は、従来の暗号技術と比べ、

- ランダムな暗号化鍵ではなく、良く知っているメールアドレスや人名を暗号化鍵に利用できる。
- 属性、例えば課長以上の役職の職員、を指定した、ファイル、ドキュメントへのアクセス制御ができる。
- 復号鍵の漏洩対策として、複数人の復号情報が集まらなければ、もとのデータに復号できない設定にできる。
- 暗号化したまま論理演算や算術演算ができる。
- 個人情報を秘匿したまま、あるグループに所属しているメンバーシップを認

証できる。
- 個人が実際に所有する秘密情報を公開することなく、その秘密を持っていることを他者に証明することができる。
- DB へのアクセスパターンを秘匿することができる。
- 管理者等によるコンピュータの RAM の解析を防ぐことができる。
- DB 管理者等に、どんなデータを検索したかを秘匿したまま、所望のデータを検索できる。

等の特徴を持つ暗号技術であり、これらの特徴が従来の暗号技術よりも優位な点となる。

このように様々な高機能暗号があるが、その優位性は種別毎に異なっている。このため、高機能暗号を使用するにあたり、その目的にあった高機能暗号の選択が必要となる。

2.3 高機能暗号の種類と分類

2.2.2 章で述べたように、高機能暗号は、その使用目的にあった選択が必要となる。そこで、本ガイドラインでは、高機能暗号の目的に応じて、“守秘”、“認証・署名”、“その他” の 3 つに分類した。

守秘は、第 3 者に秘匿する文書等を暗号化するための技術である。認証・署名は、文書等を作成する人、文書等を送る人、および、通信している機器の認証を行う、もしくは、それらに署名を付与するための技術である。その他については、上記の守秘、認証・署名には属さない新たな技術である。例えば、複数の人やサーバが協調しつつ各利用者が持つ秘密情報は秘匿して計算を行う方式や、秘密情報を提示することなく個人が秘密情報を持っていることを提示する方法等の技術である。

守秘と認証・署名は CRYPTREC 暗号リストにも同じ分類があるが、その他については、CRYPTREC 暗号リストの分類では、どの分類にも当てはまらない暗号技術となる。表 2.1 に、本ガイドラインで扱う暗号方式の一覧を示す。

2.4 高機能暗号の活用例と標準化動向

以前は、セキュリティ技術は付加機能であり、開発コストが高く、敬遠され、進んで導入する組織は多くなかった。現在では、企業からの情報漏洩は、企業のイメージを落とすことにもつながるため、たとえ一部の情報であっても漏洩することがあってはならない状況である。そして、サイバー攻撃等による情報漏洩の危

表 2.1　本ガイドラインで扱う高機能暗号一覧

分類	高機能暗号	特徴
守秘	ＩＤベース暗号	任意の記号列を公開鍵にすることが可能
	属性ベース暗号	受信者の属性に応じた復号権限の付与が可能
	放送型暗号	多数の受信者に対して放送用のコンテンツの効率的な復号権限の制御が可能
	準同型暗号	暗号化した状態での演算が可能
	プロキシ再暗号化	サーバにおいて暗号化鍵を変更することが可能
認証・署名	属性ベース署名	個人ではなく、属性に応じた署名権限の付与が可能
	集約 MAC、マルチ MAC、集約署名、マルチ署名	複数の MAC ／署名をコンパクトに圧縮することが可能
	グループ署名、リング署名	グループを構成するメンバの誰かが匿名で署名をすることが可能
	しきい値署名	一定数以上の署名者がそろうことで、一つの署名生成が可能
その他	秘密分散	秘密データを複数に分割し秘匿する方法
	マルチパーティ計算	複数のサーバが入力情報を秘匿したまま協調的に計算して結果を得ることが可能
	ゼロ知識証明	証明者が検証者に命題の正しさを証明でき、それ以外の情報を漏らさないことが可能
	Oblivious Random Access Machine （ORAM）	サーバ内のデータアクセスにおいて、データの秘匿に加えてデータへのアクセスパターンをサーバに対して秘匿することが可能
	Private Information Retrieval （PIR）	サーバ内のデータアクセスにおいて、どのデータにアクセスしたかをサーバに秘匿することが可能
	検索可能暗号	サーバ内のデータへのキーワード検索において、検索キーワードを秘匿したままデータ検索が可能

険性もあり、セキュリティ対策を堅実に行うことが組織のアピールにもつながり、セキュリティ技術を積極的に導入する組織が増加している。

さらに、社会環境も変化し、種々多様なサービスが発展してきている。この変化に伴い、ネットワークや、データの保護方法も変化し、高機能暗号が期待され、提案されている。また、すでに導入、活用され、実用化の段階に入っている高機能暗号も増加している。以下では、個々の高機能暗号に対し、どのような活用例が

考えられているかを示すとともに、使用可能なライブラリや、標準化に向けた活動がある方式については標準化動向を示す。その後、活用事例に対し、高機能暗号を利用した実例を紹介する。

2.4.1 守秘関連の活用事例と標準化動向

■ID ベース暗号 ID ベース暗号は、利用者の ID をはじめとする任意の文字列を公開鍵に利用できる公開鍵暗号方式の一種である。暗号文を受け取る相手の名前、メールアドレス、ID、企業名、部署名等、よく知っており、一目で相手がわかる情報を公開鍵として利用することができるため、相手とは全く無関係の公開鍵を利用した従来の暗号方式よりも利便性に優れている。

具体的には、メールの本文や添付ファイルを暗号化するサービス等が商用化されている。そのメールサービスでは、個々の利用者の公開情報であるメールアドレスを公開鍵として、それに対する秘密鍵がサービス事業者により生成され、秘密鍵はそれぞれの利用者に送られ保持される。メールの送信者は、メール送信先のメールアドレスを公開鍵（暗号鍵）として、メール本文を暗号化する。メール受信者は、メールアドレスに対する秘密鍵（復号鍵）を持っているので、このメールを復号して読むことができる。秘密鍵は、メールアドレスの受信者しか持っていない情報であるため、メールの守秘性が保たれる。この方式では、送信者はメール受信者の公開鍵を知る必要はなく、メールアドレスだけを知っていれば、受信者が復号できる暗号文を作ることができるため、従来の公開鍵暗号のように受信者の公開鍵を取得するための通信等の必要がない。

ID ベース暗号では、複数のライブラリがオープンソースソフトウェアとして公開されるなどソフトウェアが充実しており、パフォーマンスも高い [2]。まず、ID ベース暗号に限らず近年の多くの公開鍵暗号方式で使われている Pairing と呼ばれる基盤技術がある。この Pairing をベースに最初に作られたライブラリが PBC ライブラリ（Pairing-based Cryptography Libraly）[3] である。PBC ライブラリの中には ID ベース暗号関連のソフトウェアも含まれている。PBC ライブラリ以外にも ID ベース暗号関係のライブラリは存在するが、PBC ライブラリが最も参照され、使用されている。商用のサービスとしては Voltage 社による電子メールの暗号化ソリューションである Voltage SecureMail がある [4]。また Cloudflare 社が提供する CDN 顧客の秘密鍵管理のソリューションである Geo Key Manager において ID ベース暗号の発展である Identity-based Broadcast Encryption と Identity-based Revocation が採用されている [5]。

また、標準化活動も活発である。IETF では 2007 年 12 月に RFC 5091 として

Boneh-Franklin の手法と Boneh-Boyen の手法が標準化された [6] ことを皮切りに、ID ベース暗号関連の標準化がすすめられた。RFC 5091 をもとにデータ構造等が RFC 5408、5409 として 2009 年 1 月に標準化された [7, 8]。2011 年に入り、ID ベースの認証鍵交換（ID based Authenticated Key Exchange，IDAKE）手法が RFC 6267 として標準化され [9]、2012 年に、IDAKE 自体が RFC 6539 [10]、ID ベース暗号向けの Certificateless Signature 手法が RFC 6507 [11]、境-笠原の手法が RFC 6508 として標準化された [12]。IEEE では P1363.3 として ID ベース暗号だけではなくより広い Identity-based Cryptography（IBC）として標準化の活動が行われ、2013 年に IEEE Std 1363.3-2013 として標準化されている [13]。そこでは境-笠原の Key Encapsulation Mechanism（KEM）、Boneh-Boyen の KEM、Boneh-Franklin と Boneh-Boyen の ID ベース暗号方式が Identity-based encryption scheme として標準化されている。ISO/IEC でも ID ベース暗号だけではなくより広い範囲で Identity-based ciphers として ISO/IEC 18033-5:2015 が策定されている。そこでは IEEE Std 1363.3 と同様に Boneh-Franklin、Boneh-Boyen、境-笠原の手法が標準仕様として掲載されている [14]。

ID ベース暗号の活用に関する学術の研究としては、2003 年に ID ベース暗号のパラメータ配付に DNS を利用する方式が Smetters らにより提案されている [15]。また 2013 年に Ruoti らによる電子メール暗号化の高いユーザビリティを実現する Private Webmail（PWM）において ID ベース暗号が採用されている [16, 17]。Miklejohn らは交通系の料金収受システムの提案内において Blind Identity-based Encryption を利用し、被発行者 ID を利用した暗号化を行っている [18]。機器認証において ID ベース暗号を用いる手法は Software Defined Network（SDN）の研究において 2012 年の Veltri らの研究を筆頭に、2010 年代半ばに複数研究されている [19, 20, 21]。

■属性ベース暗号　属性ベース暗号は、暗号文の復号の際に、暗号文の受信者の肩書、会社名、地域名等の属性に応じて復号権限を変更することができる公開鍵暗号方式の一種である。より正確には、暗号文または復号鍵（秘密鍵）に属性に対応したアクセス構造を持たせることにより、当該暗号文の復号権限を一括で制御できる。従来の公開鍵暗号方式により属性毎のアクセス構造を持たせる場合、同じ属性を持つ利用者の公開鍵を用いて、利用者数に応じた暗号文を作ることになる。これに対し、属性ベース暗号では、暗号文は一つでアクセス構造を持たせることができ、効率性において従来の公開鍵暗号方式よりも優れている。

属性ベース暗号の活用事例としては、クラウド環境を介したファイル交換サービスにおける暗号化されたファイルへのアクセス制御の仕組みをセキュアかつ効

率よく実現するために属性ベース暗号を利用しているソリューションが提供されている [22]。

具体的には、ある企業内のファイルにおいて、部長、課長は閲覧可能であるが、他の職員は閲覧できないようにするソリューションである。そのソリューションでは、部長、課長が持つ秘密鍵には、部長、課長に対応する属性情報が埋め込まれている。他の職員には、部長、課長の属性情報が埋め込まれていない秘密鍵が配布されている。ある企業内の機密ファイルに、部長、課長の属性をもつ復号鍵であれば復号可能となるように、公開鍵（暗号化鍵）と部長、課長という属性を用いて暗号化し、クラウドに保存する。クラウドからファイルの取得は誰でも可能であるが、復号時に、復号者の秘密鍵に埋め込まれた属性が適合しなければ復号できない。すなわち、属性に応じた復号のアクセス制御が可能となる。

一般に、公開鍵暗号においては、暗号文生成時においての復号権限をもつエンティティは唯一に指定されるため、複数エンティティに復号権限を与える場合にはその数に比例する暗号文の生成が必要である。一方、属性ベース暗号では上記の機能を単一の暗号文で実現可能となる。属性ベース暗号は、暗号文ポリシー型と鍵ポリシー型に大別され、それぞれ暗号文と秘密鍵にアクセス構造を埋め込む技術である。利用者は、この特徴に応じて、どちらかの型を選択することになる。すなわち、ユースケースにおける暗号文と秘密鍵の更新頻度を比較し、相対的に更新頻度が少ない方へアクセス制御を埋め込む型を選択する方が効果的である。

そして、属性ベース暗号の様々な実現手法を実装したライブラリがオープンソースソフトウェアとして公開されている [23, 24, 25]。[23] は、ID ベース暗号の項で述べた PBC ライブラリを用いて作成された暗号文ポリシー型属性ベース暗号のソフトウェアである。[24] は Garg らによる GCH+13 方式 [26] の属性ベース暗号を用いて作成された Java ベースのライブラリである。[25] は Goyal らの属性ベース暗号 [27] をもとにして作成されたライブラリである。

■放送型暗号　放送型暗号とは、一つの暗号化鍵を用いて暗号化されたコンテンツを、複数の利用者が同時に復号できるようにする方式であり、暗号化鍵と復号鍵が異なる公開鍵型放送型暗号と、暗号化鍵と復号鍵が等しい共通鍵型放送型暗号がある。複数の利用者に同じコンテンツを送る場合であっても、暗号化鍵は一つしかなく、すなわち、暗号化は一度でよい、という点で効率性に優れている。

公開鍵型放送型暗号は、利用者毎に異なる復号鍵を持つが、暗号化のための公開鍵は一つである。暗号化鍵は公開情報であり、公開情報のみで誰でも暗号化が可能だが、この性質を積極的に活用可能な応用先が乏しく、現時点では具体的な社会実装例は見当たらない。

共通鍵型放送型暗号は、暗号化鍵と復号鍵が同一であり、また、利用者すべてが同じ復号鍵を持つ方式である。誰でも暗号化を行うことができる公開鍵暗号型放送型暗号とは異なり、暗号化を行うことができる機関が暗号化鍵を共有したエンティティに限定されるものの、大規模コンテンツ配信システム等においては、暗号化するエンティティがごく少数であるため、そのような制約はそれほど問題とはならず、大規模コンテンツ配信の実用システム上では共通鍵型放送型暗号方式が利用されている。また、Blu-ray 等のメディア用にも使用されており、具体的には、Blu-ray 等のコンテンツ保護に用いられる Advanced Access Content System（AACS） [28] は、Naor らにより開発された放送型暗号である SD 法 [29] と共通鍵暗号を組み合わせた共通鍵型放送型暗号に基づき構成されている。AACS については、デジタル家電業界、パソコン業界、映画業界の主導により、規格策定団体 AACS LA [30] が設立され、IBM、 Intel、 Microsoft、 Panasonic、 SONY、 Disney、 Toshiba、 Warner Brothers 等により運営されている。

■準同型暗号　準同型暗号は、データを暗号化したまま演算処理（四則演算等）ができる暗号技術である。可能な演算は暗号方式により異なっており、加算のみ、乗算のみ、加算と乗算の両方が可能な方式等がある。暗号化状態での演算であるため、演算の途中でデータが漏洩したとしても、生のデータが解読されることはなく、データを処理速度の速いコンピュータに預けて演算を行う分析等の応用例が具体化されている。具体的には、機械学習の分野で準同型暗号を活用することにより、学習データのプライバシを確保しつつ機械学習モデルを生成する手法に関する研究開発等がある。

すでに実証実験に至っている事例もある。複数の金融機関の取引データを利用して不正取引検知の精度を向上することを目的とする実証実験である。この実験では、複数の金融機関が持つ取引データを持ち寄り、不正取引検知用の AI モデルを構築する方法が提案されている [31]。複数の金融機関のデータは、他の金融機関や中央サーバには秘匿すべきデータとして扱う。そして、それらを AI の連合学習に利用する。深層学習を行う際には、DeepProtect と呼ばれるプライバシ保護深層学習技術が使われている [32, 33]。DeepProtect では、準同型暗号により学習中のパラメータ（勾配情報）を暗号化して中央サーバに送り、中央サーバでは、暗号化したまま、学習モデルのパラメータの更新を行う。

鉄道会社の IC カードの分析に秘密計算を取り入れる検証等も実施されている [34]。そして、IC カードの利用履歴、乗降履歴や購買履歴等のセキュアな利用に取り組んでいる。この検証では、加算型の準同型暗号の一つである Paillier 暗号 [35] が使用されている。また、暗号化前と暗号化後の順番が変化しない順序保

存型暗号（Order Preserving Encryption）も活用しているとされている。

準同型暗号については、ライブラリ、ソフトウェアが数多くオープンソースソフトウェアとして公開されている。Brakerski らによる準同型暗号 BGV12 方式 [36] と Cheon らによる準同型暗号 CKKS17 方式 [37] を基に開発されたライブラリ HElib [38]、Brakerski による準同型暗号 Bra12 方式 [39]、Fan らによる準同型暗号 FV12 方式 [40]、CKKS17 方式に基づくライブラリ SEAL [41]、 CKKS17 方式 に基づくライブラリ HEAAN [42]、Bra12 方式、 FV12 方式、BGV12 方式、CKKS17 方式、Ducas らによる準同型暗号 DM15 方式 [43]、 Chillotti らによる準同型暗号 CGGI16 方式 [44] に基づくライブラリ PALISADE [45]、Bra12 方式、FV12 方式、CKKS17 方式に基づくライブラリ Lattigo [46] 等がある。さらに、HEAAN を用いたプラットフォーム等の製品 [47] が商用化されている。

準同型暗号の標準化も行われており、標準化団体 Homomorphic Encryption Standardization [48] は準同型暗号のセキュリティ、API &アプリケーションの標準化を目指している。この標準化では、記法と準同型暗号化のセキュリティプロパティ、完全準同型暗号スキーム、推奨パラメータ値、標準的な攻撃方法とその耐性、推定実行時間だけでなく、将来を見越した耐量子計算機対応としてラティス攻撃、LWE に対する Arora-Ge 攻撃、RLWE に対する代数攻撃方法とその耐性が標準化の対象となり、今後も、API 設計、アプリケーション例が追加される予定となっている。さらに、ISO/IEC では、単一演算型の準同型のアルゴリズム（Exponential ElGamal encryption、Paillier encryption）の暗号処理にかかるプロセスを規定している [49]。

■プロキシ再暗号化　プロキシ再暗号化は、ある暗号文を復号せずに、暗号文の作成者とは異なるエンティティ（プロキシ）が、平文が同じである別の暗号文に変換する暗号方式である。入力された暗号文と出力される暗号文では、秘密鍵が異なることが一般的であり、このため、秘密鍵の更新が必要となるアプリケーションや、所有者変更等のアプリケーションでの応用例が報告されている。

例えば、復号を実行せずに動的に受信者を変更できるため、動的にアクセス権限の変更をすることを想定するクラウドストレージ等において特に有用であると考えられる。そして、NuCypher 社は、Umbral PRE と呼ばれるプロキシ再暗号化方式を開発し、これを用いた事業展開を進めている [50]。ここでは、サーバにおかれた暗号化データを利用者に配布する場合、その利用者の秘密鍵で復号できるようにアクセス制御する方式等が提供されている。また、東芝も独自に開発したプロキシ再暗号化方式を用いて、デジタル貸金庫と呼ばれるサービスを 2021 年まで展開していた [51]。デジタル貸金庫では、サーバに蓄えられたデータを、複数の

デバイスで利用できるようになっている。ただし、個々のデバイスの復号鍵は異なっている。この個々のデバイスで利用するために、サーバに蓄えられた暗号文から、利用するデバイスの復号鍵で復号できる暗号文にプロキシ再暗号化を利用して変換していた。

2.4.2 認証・署名関連の活用事例と標準化動向

■属性ベース署名 属性ベース署名とは、属性ベース暗号の署名版と考えて差支えない。属性ベース暗号の場合は、復号する利用者の属性に応じて、より正確には、復号する利用者が持つ復号鍵に応じて、暗号文へのアクセス制御を行っている。これに対し、属性ベース署名では、署名者の持つ署名鍵が、署名者の属性に応じて異なっており署名者の属性集合が埋め込まれた署名鍵になっている。そして、ある文書に対して署名鍵を用いて署名を生成するわけであるが、その署名の中に属性に関するアクセス制御構造が埋め込まれる。署名検証において、署名者の持つ属性集合が、アクセス制御のために埋め込まれた構造にマッチしている場合は、署名検証をパスすることができる。すなわち、署名者の属性に応じて、検証判定ができる署名方式である。

この属性ベース署名を使うことで、社内に送付する文書において、名前ではなく、文書作成者の属性（役職、部局等）等に応じた署名を付与することができる。ある社の A さんが作成した文書を、その課長、部長等が押印して回付することはよくあることである。紙媒体の場合、A さんの印、課長印、部長印が押印される。この課長印、部長印は、課長もしくは部長の権限を持つ人が、課長／部長の責任に応じた役割を果たし、文書を確認したことを、文書の受領者に示すものであり、課長、部長の名前よりも、その役職に意味がある印である。さらに、詳細に、総務部の課長印のように、部署名も必要なこともある。この押印を電子的に行う場合、属性ベース署名が役に立つ。

総務部の課長 B さんは、”所属：総務部”、”役職：課長”の属性を持つ。このため、B さんの署名鍵（秘密鍵）は、”所属：総務部”、”役職：課長”に応じた署名鍵となる。B さんが署名を生成する際に、署名を受理するためのアクセス条件を付与する。この例の場合、”総務部”かつ”課長”であれば、受理するという条件になる。文書の受信者は、署名がアクセス条件にあっているかどうかを確認することになる。B さんの署名は、”総務部”かつ”課長”の属性であるため、署名検証が通る。もし”総務部”かつ”課長”ではない第 3 者が署名を偽造したとしても、署名検証でエラーとなる。

この例で示す通り、名前による確認ではなく、役職や所属部署名による確認を

行うことができる。もちろん、企業によっては、課長印、部長印以外に、名前が必要な場合もある。この場合は、”名前：B さん”または”名前：C さん”または”名前：D さん”等をアクセス条件に入れることで対応ができる。

また、属性の作り方を工夫することで、多くの活用方法が考えられ、例えば、ブロックチェーンやユーザ認証において匿名性を確保するために属性ベース署名を活用する手法の研究が行われている [52, 53]。

■集約 MAC・マルチ MAC・集約署名・マルチ署名　集約 MAC は、異なる複数の文書に対し、複数もしくは一つの秘密鍵でメッセージ認証符号（Message Authentication Code，MAC）を付与する場合を効率よくするための技術である。例えば、企業内で文書に対し、文書の作成者とその上司、および、別の部の担当者等、複数の人が異なる秘密鍵により MAC を付与する場合である。通常の MAC 方式であれば、MAC を付与する人数が多くなると、人数に比例して MAC 部分のサイズが大きくなる。これに対し、集約 MAC では、複数の MAC を単一の MAC に集約するため、MAC サイズが小さくなるという優位性を持つ。

マルチ MAC は一つの文書に対し、複数の異なる秘密鍵で生成された MAC を統合し、集約する技術である。MAC を付与する文書が共通であり、集約 MAC の特殊ケースとして捉えることができる。そして、集約 MAC と同じように、MAC サイズが小さくなる優位性を持つ。

集約署名は集約 MAC の電子署名版と考えることができる。複数の文書に対し、複数人が署名するような場合を効率よくするための技術である。通常の署名方式であれば、署名を付与する人数が多くなると、人数に比例して署名部分のサイズが大きくなる。これに対し、集約署名では、複数の署名を単一の署名に集約するため、署名サイズが小さくなる、という優位性を持つ。また、署名を検証する人は、これを効率よく検証できる提案もある。すなわち、集約された署名の人数分の検証を行うのではなく、一回の検証処理で行う方式である。

マルチ署名は、一つの文書に対し、複数人が署名を行う場合、この署名を単一の署名にまとめる技術である。集約署名と同じように、署名サイズが小さくなる点がメリットである。

集約 MAC および集約署名については、現時点で具体的なサービス等の提供は行われていないが、ITU により廣瀬らによる方式 [54]、佐藤らによる方式 [55] の技術内容の標準化が行われている [56]。また、この技術を用いた認証関連のデータ量の削減効果を考慮し、認証に係るインターネット上のトラフィック量削減を目的とする活用に関する研究が行われている [57]。

マルチ署名は、暗号資産 Bitcoin やブロックチェーンプラットフォームである

Ethereum におけるマルチシグウォレットにおいて活用されている。マルチシグウォレットは、複数のユーザが共同利用することを前提とし、取引の実行には全てのユーザの同意が必要となるものであり、同意形成の仕組みとしてマルチ署名が利用されている。例えば、信頼できる第三者に署名生成鍵を委託することにより、他のユーザによる不正送金等を防ぐことを目的に利用される [58, 59]。

■グループ署名　グループ署名は、署名者があるグループの一員であることは明示的に分かるが、グループ内の誰であるかまでは分からない匿名性を保証することができる署名方式である。一般の署名方式では不可能であった、署名者のプライバシ保護、匿名性が必要となるアプリケーションにおいて需要がある。

例えば、プログラム開発での確認作業、いわゆる Attestation において、2000 年代からグループ署名が利用されるようになった。ここでの Attestation とはプログラムが製造者やユーザの意図した通りに動作しているかを確認するための処理であり、ブート結果や周辺機器の接続状況等の予め想定された処理結果に対してデジタル署名をつけて正当性を確認できるようにするものである。集中管理するケースにおいては通常のデジタル署名や PKI を利用した場合、Attestation をしたエンティティの活動に対するプライバシ侵害が起こりうるため、グループ署名を利用してそれを回避する仕組みが取られることが多い。

また、ISO/IEC において、標準化も行われている [60, 61]。

■リング署名　リング署名では、署名者があるグループの一員であることは明示的に分かるが、グループ内の誰であるかまでは分からない匿名性を保証する署名方式である。一般の署名方式では不可能であった、署名者のプライバシ保護、匿名性が必要となるアプリケーションにおいて需要がある。この意味では、グループ署名と同じ効果を有する署名方式である。ただし、グループ署名とは異なり、グループ管理者を置く必要がなく、署名生成時に柔軟にグループを選ぶことができるという点で、グループ署名とは異なっている。

グループ管理者を置かないことにより、グループ生成を柔軟にできることはメリットであるが、逆に、署名者の誰かが不正を行った場合に、不正を行った署名者が誰かを特定することは困難である、というデメリットもある。グループ署名を使用するか、リング署名を使用するかは、その利用環境に応じて決められるべきことである。例えば、署名を生成できるユーザを管理者が認可したユーザだけに絞りたい場合はグループ署名を使用する。この場合、管理者に対して、署名者の匿名性を担保することは困難となる。これに対し、リング署名の場合は完全な匿名性を担保したい場合に使用する。グループ署名とは逆に、署名者を管理する管理

者がいないため、誰でも署名を生成することができる。

リング署名は、実用事例は少ないが、活動が盛んになっている暗号資産において使われている [62]。

■しきい値署名　しきい値署名は、「シェア」と呼ばれるある情報の分散情報を利用する。まず、一つの平文に対し、複数の署名鍵のシェアを用いて署名のシェアを生成する。この署名のシェアをあるしきい以上集めて、結合することで真の署名を生成する方式である。検証鍵は一つだけであり、各署名シェアではなく、真の署名の検証時に利用する。平文の正当性を一人の署名だけで確認するのではなく、しきい値以上の複数人の署名、すなわち、複数人が個々に平文を確認し、署名することにより、複数人が確認した平文となる。しきい値署名では、この複数人が生成した署名を、一度に検証することができる。

暗号資産における鍵漏洩への対策としてしきい値署名が使われることが多くなっている。例えば、しきい値署名の利用例として Bitcoin 等での利用が挙げられる。Bitcoin ではマルチ署名のような複数の署名者によるトランザクションの生成が可能である。しかし、この方法では複数の署名を必要とするため、ストレージ効率が悪いことや、署名者の構成が公開されてしまう等のデメリットがある。一方、しきい値署名はオフチェーンで利用することが前提となるが、複数の署名者がしきい値署名を使って署名の断片を作り、その署名を集めて一つの有効な署名としてトランザクションを生成する。この署名は単一の署名値でありマルチ署名に比べてストレージ効率が良く、署名者の構成が署名値からは分からない。ここでのオフチェーンとは直接ブロックチェーンの処理に組み込まれることのない処理のことを指している。

そして、このオフチェーン処理の署名について、特に Bitcoin が利用する署名方式を ECDSA からシュノア署名に置き換わるにあたり、しきい値署名の利用が推進される可能性が種々報告されている [63]。また、暗号資産 quredo のイエローペーパーではしきい値署名が使用され [64]、フレセッツ社では HSM を利用したしきい値署名 [65] による Ethereum 秘密鍵保護方式 [66]、Coinbase 社では DApp Wallet [67] と呼ばれる暗号資産用のアプリで採用しているようであり、分散型のブロックチェーンを開発するプロジェクトである DFINITY ではスマートコントラクト canister が保持する Bitcoin や Ethereum の鍵をしきい値 ECDSA にする提案（V.Shoup） [68] 等の報告がある。

これに応じて、標準化活動もおこっている。アメリカ国立標準技術研究所 (National Institute of Standards and Technology, NIST) では、2019 年にしきい値暗号に関する NIST の標準化の目的等について記述された NISTIR 8214 [69] が

発行され、さらに 2020 年にそのロードマップとなる NISTIR 8214A が発行された [70]。そして、2021 年 9 月にフィードバックの締切があり、2022 年 8 月に開発者向けガイドラインと推奨の 8214B が発行された [71]。また、IETF では、標準化提案が行われ、標準化プロセスが始められている [72]。

2.4.3 その他の高機能暗号の活用事例と標準化動向

■秘密分散 秘密分散法は、その名の通り秘密情報を何らかのグループのメンバー間で分散する暗号技術の一種である。秘密鍵などの秘密情報を保管する際は、その秘密情報の紛失、破損、盗難、漏洩といった脅威を考慮する必要がある。破損や紛失については情報のコピーを作ることで対策できるが、コピーが増えれば増えるほど盗難や漏洩のリスクが高まる。これを解決するため、秘密分散法は秘密情報を「シェア」と呼ばれる複数のデータに分割し、これを複数の当事者間で分散して保管する。このシェアは、ある定められた組み合わせを揃えたときのみ元の秘密情報が復元でき、また、復元の条件を満たさないシェアを揃えても、元の秘密に関する情報が漏れないように作られる。最もシンプルでよく用いられる「組み合わせ」の例はしきい値型である。すなわち、n 個のシェアのうち、k $(\leq n)$ 個のシェアを集めることで秘密が復元できるような方式である。

上述の特徴の通り、秘密分散法の直接的な応用は分散ストレージサービスである。秘密分散法を用いて分散保管することで、データの秘匿性を保証しながら冗長性を確保することが可能となる。そして、この冗長性の確保により、一部のストレージに障害が起きた場合への耐性を取ることが可能となる。具体的なアプリケーションとしては、文書管理システムやコンテンツ配信システムなどへの適用事例がある [73]。自然災害の多い日本国においては、事業継続計画（Business Continuity Plan，BCP） 対策は重要な課題であると認識されており、とりわけ東日本大震災の直後は、Azure 等のクラウドストレージサービスを利用しながら世界中のサーバにデータを秘密分散するサービス [74] が登場するなど、セキュリティと災害対策を両立する技術として活用されている。

また、計算量的な秘密分散法として知られる All-Or-Nothing Transform（AONT）を利用したものでは、Shin らによる AONT [75] を用いたサービスが ZenmuTech 社によって実装され、近年実用化されている [76]。

秘密分散法の応用は分散保管にとどまらず、様々な暗号プロトコルの要素技術としても用いられている。特に、複数のエンティティで一つの計算を行うマルチパーティ計算との関連が深い。近年ではブロックチェーンと組み合わせた秘密鍵の分散管理システムなどへの応用もあり、マルチパーティ計算と組み合わせるこ

とで暗号資産等の秘密鍵を安全に運用する手段として注目されている [77]。

標準化も進められており、ISO/IEC SC27 WG2 においては、秘密分散法の定義、記法、代表的な方式の標準規格が発行されている [78]。現在の規格では、5 つの代表的な方式が記載されている。1 つ目は、しきい値型の完全秘匿性を達成する Shamir の秘密分散法 [79] であり、今日においては最もよく知られた方式である。完全秘匿性とは、攻撃者が無限の計算能力を持っていたとしても破ることができな秘匿性のことである。2 つ目は、完全秘匿性の要件を緩和する代わりに、シェアのサイズを秘密情報のサイズより小さくするランプ型秘密分散法（Ramp Secret Sharing） [80, 81] であり、Shamir の秘密分散法の一般化として構成される。3 つ目は、加法型秘密分散法（Additive Secret Sharing）と呼ばれる手法に基づく、一般的なアクセス構造を実現する秘密分散法 [82] である。これは、秘密情報へのアクセスを許されるシェア集合と、許されないシェア集合を指定可能な秘密分散法であり、しきい値型よりも柔軟なアクセス構造を導入することが可能である。4 つ目は、複製型秘密分散法（Replicated Secret Sharing）と呼ばれる手法に基づく、一般的なアクセス構造を実現する秘密分散法 [83] である。この方式も、任意のアクセス構造を実現することが可能である。こちらの方式はシェアのサイズがパーティ数に対して大きいが、方式の特殊ケースは近年の効率的なマルチパーティ計算にも利用されている [84, 85]。5 つ目は、計算量的安全な秘密分散法 [86, 87] である。これは、情報理論的に安全な秘密分散法と共通鍵暗号を組み合わせることで実現でき、情報理論的な秘密分散法と比較してシェアサイズが大幅に削減できることが知られている。

■マルチパーティ計算　マルチパーティ計算（Multi-Party Computation，MPC）は、複数の参加者（マルチパーティ）が自身の秘密情報を秘匿しながら、それらの秘密情報を入力とする種々の関数をパーティ間で協調計算することを可能とする技術である。MPC には、大きく分けて秘密分散法をベースにしたものと、Garbled Circuit をベースにしたものがある。前者は後者に比べて参加者間で複数回の通信を必要とする代わりに、計算量・総通信量が小さい傾向にある。また、両者ともに参加者のうち一定数が結託しないという仮定の下で安全性が保証される。そして、標準的な公開鍵暗号では困難であった暗号化しながらの計算処理が、MPC では可能となる。また、準同型暗号と比較した場合、通信量や計算量に優れるため、処理のスループットが高いという特徴がある。

秘密分散法をベースにした MPC 方式の活用事例として、個人の秘密情報を秘匿したまま、統計情報の計算・機械学習等のデータ分析を行うサービス提供がある [88]。例えば、病院の臨床データの分析のために、ディープラーニングをする

際、入力に対してマルチパーティ計算させることで、平文の入力データを誰にも知られることなく、分析結果を得ることができる。秘密分散法がベースであるため、前述の秘密分散の項で説明したように、入力データはシェアと呼ばれるデータに分割され、複数の入力データシェアが作成される。その後、マルチパーティ計算を行う複数のサーバにそれぞれの入力データシェアが入力される。入力シェアを得たサーバは、ディープラーニング用にサーバ独自の計算を行うとともに、別のサーバと通信をすることで、分析結果データを分割したものとなる出力シェアを生成する。この出力シェアを全部集めて合成することで、分析結果を得る方法である。

さらに、複数の医療機関が保持するゲノム情報や診療情報を統合して解析することにも応用されている [89]。このゲノム情報は非常に機微性の高い情報であり、漏洩は絶対に許されない。そこで、MPC を用いてデータ秘匿性を守りながらデータを収集し、医療・医学に活用する方法としている。

さらに、国外でも、政府の持つ TAX Records と STUDENT Records より、教育と年収の関係、大学在学中のアルバイトと留年の関係、大学の学問分野とアルバイト量の関係を分析 [90] したり、エネルギー消費の最適化を行うために大手企業の産業データを秘匿化して収集し分析するエネルギー供給の最適化 [91] 等、多くの応用例がある。また、国内外で多くのスタートアップ等による活用が進んでおり、コミュニティの形成を通じた知見の共有、企業間の連携が進んでいる [92]。さらに、秘密計算技術が広く社会実装され、クラウドサービスのデータ保護に対する不安の払拭や、組織や企業の枠を超えたデータ利活用により新たな価値が創出されることを目的とし、様々な秘密計算方式を俯瞰した実用的かつ客観的な安全性基準や、ユーザが秘密計算を活用する際の参考となる指針の検討、ホームページやイベントを通じた、上記基準や指針の検討状況、方式の性能等に関する技術資料や先端事例の情報発信 [93]、秘密計算を中心としたプライバシ保護関連技術に関連する情報発信、および、個人情報保護法の改正を始めとしたデータ活用とプライバシ保護がコンフリクトしている現状に対応すべく、法令遵守したデータ活用やプライバシ保護テクノロジの勉強会や情報発信を行っている団体もある [94]。さらに、NIST はプライバシに関するプロジェクト Project on Privacy-Enhancing Cryptography（PEC）の中で、MPC の普及に務めている [95]。

標準化も進められており、現在、ISO/IEC SC27 WG2 において、秘密分散型 MPC の定義、記法、代表的な方式の標準化に向けたドラフト作成が進められている [96]。

Garbled Circuit をベースにした方式では、2 者計算の様々なシーンで利用されるが、近年では鍵管理の文脈において、秘密鍵を分散して保護しながら、それを

Garbled Circuit によって復元することなく暗号化・復号・署名等の計算を可能とする Hardware Secure Module（HSM）の代替としての応用が存在する。特に、暗号資産における電子署名の鍵管理手法として注目を集めており、従来の取引所等が集中的に鍵を管理する形態と比べて、鍵管理を集中して行う必要がなくなり、取引所等の単一障害点が排除されており、秘密情報を含む情報漏洩に対しても耐性があり、安全性が高い利点がある [97]。

Garbled Circuit ベースの MPC についての標準化の動向はないが、先に紹介した NIST の PEC プロジェクトにおいては、PEC 技術の一種として Garbled Circuit に関するワークショップ等での報告が行われている [95]。

■ゼロ知識証明　ゼロ知識証明とは、ある人が他の人に特定の命題を証明したいときに、証明したいこと以外の知識を与えることなく証明する手法であり、以下に示す完全性、健全性、ゼロ知識性の 3 つの性質をもつ。

- 完全性：証明者の命題が真ならば、検証者は真であることが必ずわかること。
- 健全性：証明者の命題が偽ならば、検証者はかなり高い確率で、偽であることを見抜けること。
- ゼロ知識性：あらゆる場合で、検証者が証明者から何らかの知識を得ようとしても、証明者の命題が真であること以上の知識は得られないこと。

ここで証明したい命題には、ある値が規定の範囲内に存在するかの「範囲の証明」や、任意の演算が正しく実行されたか「演算の証明」がある。一番の特徴は、証明したいこと以外の知識を与えないため、安全性が高いことである。

この特徴を利用して、いろいろな場面での活用が考えられている。例えば、不動産会社と顧客と金融機関の間での取引において
「不動産会社は、顧客に対して『住宅を借りるために十分な所得があるか』を証明してほしいと考えている。顧客は、正確な所得金額を明かしたくない。顧客は、命題『一定金額以上の所得がある』ことを、金融機関を通して、ゼロ知識証明により証明する。」
より具体的には、銀行において、顧客の本人確認を行う時に、正確な数字を明らかにすることなく、給与が特定の範囲内にあることを証明したり（ZKRP）、特定の地域の住人であることを具体的な住所を明かすことなく居住地区の範囲を証明することをゼロ知識証明により行う。そして、ZKRP をオープンソースソフトウェアとしてリリースしているケースもある [98]。

また、匿名性のある暗号資産の Zcash においては、zk-SNARK と呼ばれるゼロ知識証明を用いて、トランザクションを生成することにより、送信アドレスと受信

アドレス、送金額を秘匿することができる [99]。

さらに、パブリックブロックチェーンの一番の課題は、BtoC もしくは BtoB でやりとりされるトランザクションの匿名性の確保である。そこで、パブリックチェーンの Ethereum 上でトークンのプライベートトランザクション（匿名送金）を可能にするゼロ知識証明を用いたプロトコルが開発されている [100]。ただし、プライベートトランザクションはデータサイズが大きくなり、Ethereum のシステムを稼働させるための手数料にあたる、いわゆる“gas 代”の負担が高くなる傾向にあるが、複数の証明のバッチ化の工夫により、コスト負担を軽減させている。

また、標準化を目的とした団体があり、その中には NIST も参加している団体もある。標準化では、ゼロ知識証明を用いたデータプライバシ製品やアプリケーションのセキュリティ保証と相互運用性を向上させることを目的としている。そして、作成されたドキュメントでは、理論から実装までの情報がカバーされている [101]。

■Oblivious Random Access Machine （ORAM） サーバに暗号化データを格納し必要に応じて読み込み/書き込みを行う場合に、データは秘匿されているが、暗号化データへのアクセス頻度等のアクセスパターンの情報からサーバに統計的な部分情報が漏れてしまう。Oblivious Random Access Machine（ORAM）においては、サーバにアクセスパターンを隠しながら、暗号化データの読み込み/書き込みが可能となる。具体的には、ORAM 上で暗号化データに対して読み込み/書き込みを行うたびに、格納位置をシャッフルするとともに再暗号化することで、各アクセス間の関係を秘匿する。

ORAM を利用して、プライバシ保護を可能としたデータベースサービスが開発されている [102]。このサービスでは、プライバシ保護の安全性指標である差分プライバシの概念に従ったプライバシ保護技術を利用して個別のデータを明らかにすることなくデータ分析を可能とし、ORAM によりアクセスパターンの漏えいを防ぐことができる。また、暗号資産のトランザクションにおいて、マイナーによりトランザクションの順番を、マイナーに有利な順番に操作する攻撃が考えられている。このような攻撃を防ぐために、ORAM を利用した仕組みが開発されている [103]。この攻撃では、マイナーはユーザのアクセスを監視して攻撃を行うが、ORAM を用いることによってアクセスパターンをマイナーに秘匿できることになる。

■Private Information Retrieval （PIR） ユーザがデータベース（DB）を検索し、ユーザに必要な情報を取得するサービスは多数ある。この DB 検索を行うためには、

DB に対してクエリと呼ばれる質問を入力し、その質問に対する回答として、DB からファイルやデータ等の検索結果を受け取っている。この際、ユーザは、“質問内容も個人情報であり、この内容を DB の管理者であっても知られたくない”という場合もある。このような情報秘匿を行うために創出されたプロトコルが Private Information Retrieval（PIR）である。そして、PIR は暗号方式そのものではなく、暗号方式を用いて、ユーザのプライバシを保護した情報検索法である。

具体的には、ある DB が n 個のデータ $D = \{D_1, \ldots, D_n\}$ を持っている時に、ユーザは質問 i にある処理を施しクエリ q_i を作成し、q_i を DB に送る。この q_i からは i の情報は全く漏洩しないように工夫されている。そして、DB は i に関する情報を何も得られないが、q_i を用いてある演算をすることで、検索結果を含む複数の候補を出力することができる。ユーザは、受け取った複数の候補から、所望の D_i を抽出する。全体として、DB が得られる情報から i については何の情報も得られないプロトコルとなっている。

具体的な提案としては、Wang らは安全で信頼できるハードウエアの存在を前提として、データベースのアウトソーシングに関する考察を行なった [104]。ここでは信頼できる第 3 者機関（Trusted Third Party，TTP） の存在、非存在の両方について議論されている。当時の解決方法としては、AMD，HP，Intel，Microsoft 等が中心となっている Trusted Computing Group（TCG）の仕様に準拠したセキュリティチップ Trusted Platform Module（TPM）や Intel 社による LaGrande Technology などの採用を検討していたことが論文からは窺える一方で速度の面から効率的な実装は少なかったと判断される。

コンピュータシステム・ストレージ系の書籍 [105] において PIR を用いた活用事例でのセキュリティとプライバシを考慮したデータマイニングが実利用される場合の懸念点が示されている。１）コントロール外のデータの扱い、２）技術的に担保されていたとしても非匿名化の可能性、３）サービスに対する法域外の扱い、４）法律と現在の技術の対応が未熟、という 4 点である。PIR を適用するデータはセンシティブな扱いを要することが容易に想像できるか、上記のような課題が存在していることに留意しなければならない。

近年では 2020 年に Mozaffari らによる PIR の改良方式 [106] が広く知られることとなった。この論文の中で以下のようなアプリケーションが列挙されており、幅広い適用が検討されていることが見て取れる。

- インターネット ドメインの登録
- Tor サーバのリレー情報の取得
- プライベート メディア配信

- プライバシを保護する電子商取引アプリケーション
- オープン アクセス電子論文リポジトリでのプライベート クエリ
- メッセージング アプリケーション
- プライベート オンライン通知
- プライベート ファイル共有アプリケーション

また日本でもブロックチェーン技術への応用に関するプレスが 2020 年 6 月に発表されている [107]。Bitcoin 等で利用されている Unspent Transaction Output（UTXO）と呼ばれるトランザクション（価値の移動を表現するフォーマット）のデータベースが運用されていると仮定して、この DB に対して検索を行う際の検索クエリが流出するというプライバシ問題を解決するために PIR を利用するというアナウンスであった。ただし、Bitcoin 関連のイベントに合わせたプレスであることや、その後、このプレスを発行した組織は買収されるなどして、その後の動きなどはなく実際には実用化されていないと考えられる。

また、暗号資産のアクセスの秘匿化に使用する実装が報告され、実用化されているている [107]。この実装では、Microsoft から公開されている PIR のオープンソースソフトウェア SealPIR [108] が使われている。

■検索可能暗号　一般的な共通鍵暗号や公開鍵暗号では、サーバに暗号化文書を格納した場合に、暗号化すると平文が秘匿されるため、ある暗号文に対応する平文があるキーワードを含んでいるか検索することができない。検索可能暗号においては、サーバにキーワードの暗号文を一緒に格納することにより、検索キーワードを含むかどうか以外の平文の情報をサーバに漏らさずに、暗号文のままキーワード検索が可能となる。

検索可能暗号がもっとも活用される分野として注目されているのはデータベースと言って良い。この場合、データベースには、暗号化されたデータと、暗号化されたキーワードが保存されている。利用者はデータベースに暗号化されたキーワードのクエリを送る。データベースは復号鍵を持っていないので、平文のクエリを知ることはできない。データベースでは、その暗号化されたクエリと保存されている暗号化キーワードのマッチングを行い、検索結果を得る。検索結果も暗号化されたデータであり、その平文データをデータベースに知られることはない。

暗号化されたクエリにより検索を可能とするデータベースは MIT による CryptDB [109] を代表に多くの事例がある。CryptDB は Paillier 暗号等、複数の暗号技術を用いて暗号化データベースを実現しており、Cipherbase や SEED、Microsoft 社のシステム等、複数の商用サービスに採用されている。この他にも、

学術研究をもとにした著名なオープンソースソフトウェアとして KafeDB [110] 等がある。国内の商用のソリューションとしては、日立製作所はクラウドサービスとして検索可能暗号技術を提供しており [111]、三菱電機もソフトウェアを開発している [112] 等、さまざまなアプローチで活用されている。また、Intel SGX 上で検索可能暗号を実現する技術 [113] 等、多くの実用化研究が行われている。

検索可能暗号の標準化については、IETF において過去に準同型暗号についてのプレゼンテーション等で検索可能暗号についれ触れられていたことがあったものの、標準仕様として議論されるには至っていない。

2.4.4 活用事例からみた高機能暗号の利用方法

ここまで、高機能暗号技術がどのようなものであるか、という視点に立ち、その活用例、実施例、標準化活動を紹介した。本章では、どのようなサービス、応用方法があるか、という視点に立ち、そのサービス、応用にとって、どのような暗号方式で有用であり、実用化されたか、を紹介する。

■プライバシ強化 種々のサービスにおいてプライバシを強化する目的で、秘密計算を利用した暗号を社会実装したケースがある。秘密計算とは、計算を行うサーバ等の装置に対し、ある加工されたデータを渡し、そのサーバ等の装置において計算の元となる値を知られることなく、その結果を計算する方法の総称である。代表的な方式として、準同型暗号（3.1.4 章参照）を用いた方式、秘密分散法（3.3.1 章参照）を用いた方式、マルチパーティ計算（Multi-Party Computation，MPC）（3.3.2 章、3.3.3 章参照）を用いた方式、装置内に組み込まれている信頼できる秘匿されたハードウェア（Trusted Execution Environment，TEE）を用いる方式等が広く知られている。表 2.2 に示す事例は、この秘密計算が有用であると考えられる事例である。これらの中から、数例を以下に紹介する。

例 1. ゲノム解析による実証実験

ゲノム解析に関し、秘密計算を利用した企業と大学の共同実験が行われている [114]。ゲノムは、個人の遺伝子情報の全体であり、究極のプライバシ情報と言える。これまで、ゲノム情報は個別の医療機関で所有しているが、医療機関を超えた連携は、プライバシ情報の漏洩につながるため容易ではなかった。より多くのデータを集めることで、正確でより広範なゲノム解析が可能となるが、このプライバシ保護のハードルを超える必要があった。そこで利用したのが、秘密計算である。

実験の概要を図 2.1 に示す。ここでは、各医療機関が持つデータを加工し、複数

表 2.2　秘密計算の応用事例

領域	説明	適用例
統計処理	典型的な統計計算（合計、平均、最大値、最小値）	売上集計、ログの分析、ゲノム分析
突合	複数のデータソースから同じキーを持つレコードを突合	マーケティング、ゲノム分析
スコアリング／ AI 推論	入力および AI モデル等を秘匿しながら計算	AML（不正送金）
マッチング	要求条件を秘匿しながら、条件の合うレコードを検索	秘密オークション、場外の証券取引
連合学習	連合学習時の AI モデルの保護	特に機微なデータを元にした AI モデルの保護
投票	投票者を秘匿しながら集計	選挙

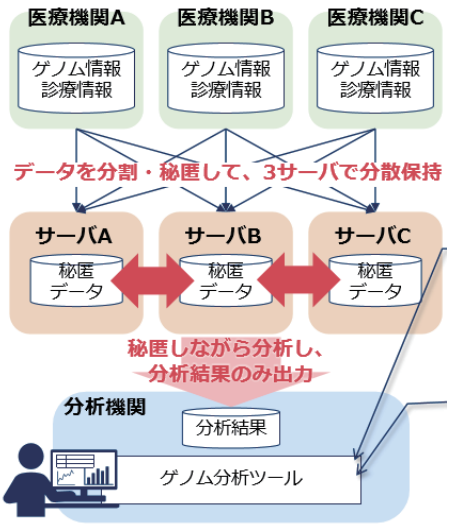


図 2.1　実証実験１：ゲノム分析（[114] から引用）

の別のデータに変換した。具体的には、個々の医療機関が持つゲノム情報や、診療情報のデータを３分割し、秘匿化する。この分割されたデータから、元のゲノム情報や診療情報のデータは復元できないようになっている。いわゆる秘密分散法を用いた方式である。そして、これらの分割され秘匿化されたデータを、それぞれ異なる３台のサーバに分散して保存する。それぞれのサーバでは、受け取ったデータを利用するとともに他のサーバと通信しつつ、あらかじめ定められた演算を行う。これは、MPC 技術の一種であり、入力データの加工方法である秘密分

散法と組み合わせ、秘密分散ベース MPC と呼ばれることもある。秘密分散ベース MPC では他のサーバとの通信を行うが、各サーバのデータが漏洩しないように、通信路の暗号化を追加する等、実運用時にはさならる安全性を担保する。

MPC では複数のサーバが必要となる。近年の個人情報漏洩のインシデントを見る限り、何らかのオペレーション時の個人の操作ミスによるものが多い。従って、複数のサーバの管理者を分けるだけでもインシデント対策として効果はある。さらに、サーバ毎に異なる組織、事業者などで管理することで、より安全なシステムが構築できる。

演算を終えた各サーバは、演算結果を分析機関に送る。分析機関では、すべてのサーバからの演算結果を統合する。この統合方法は、入力データの加工方法と、各サーバに定められた演算によって決まるが、秘密分散の逆演算とみなすことができる演算を行う。この秘密分散の逆演算を行うことにより、分析機関では、秘匿化されたデータの可読化を行う。

実験では、この可読化されたデータに対しゲノム分析ツールを用いて、典型的ゲノム分析、非典型的ゲノム分析の 2 種類の分析を行った。

- 典型的ゲノム分析（ゲノム変異分析）：ゲノム変異が起こる割合を年代や性別ごとに集計グラフ化する。
- 非典型的ゲノム分析（連鎖不平衡ブロック生成）：世代間の遺伝のし易さを変異場所のブロックで示す分析手法を利用。

これらの実験を通じ、秘密計算の処理の高速性と、分析に秘密計算が適用できることが実証された。

例２．創薬向けの実証実験

創薬向けの予測モデルの構築において、秘密計算を利用した方法の実証実験も行われている [115]。創薬においては、企業毎に種々の原料が使われ、その構造データは各企業の機密情報である。このため、データを企業から持ち出すことや、企業間で共有し、新たな創薬を行うことは困難であった。また、最近では、創薬にも機械学習が多く用いられている。機械学習では、学習によるモデルの構築が必要となるが、学習データが多いほど精度の高い学習が可能となる。このため、図 2.2 に示すような、秘密計算を用いた連合学習を行い、モデル構築の実証実験が行われた。

実証実験では、モデル構築に参加する複数の組織を想定している。各組織で個別のデータを用いて機械学習を行う。機械学習により出力されたモデルのパラメー

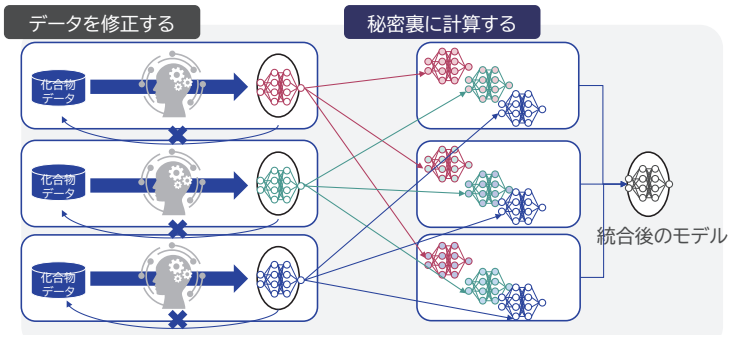


図 2.2　実証実験２：創薬向け予測モデル（NEC 提供）

タを、秘密分散し、とりまとめを行う複数のサーバに送る。秘密分散を行う前のデータは、ある意味統計加工されたデータとなっており、ここから元データを知ることは困難である。複数のサーバでモデルを統合するが、その際に秘密計算を利用して統合モデルを作成する。サーバでモデルを統合する場合には、センターと各組織の間で通信するデータはモデルそのものではなく、差分値だけを扱うことで盗聴耐性を持たせることも可能である。

この実証実験では、データを暗号化したまま計算処理ができる秘密計算を用いた複数組織間のデータ統合の有効性を検証した。

例１において、サーバへのデータ入力、データの加工方法はデータを分割する方法が記されているが、加工方法は応用事例に応じて多種多様である。特に、個人のデータを利活用する場合、個人の同意を得ることが、個人情報保護法で求められている。ただし、個人のデータに対しある加工を施すことで、制限はあるものの個人のデータの利活用が可能となっている。この加工技術が、匿名加工や、仮名加工である。

- 匿名加工：　各医療機関が、複数のデータを一つにまとめ、外部への提出用データを作成する。データはまとめられ、加工データから、元データを知ることは不可能なため匿名加工したこととなる。
- 仮名加工：　各医療機関では、実名と匿名の対応表を作り、外部への提出用に匿名のデータ表を作成する。匿名化されており、実名を知ることは不可能なため、仮名加工したこととなる。

これらは、個人情報保護法に対応した個人データの利活用をするためにも、必須の技術となっている。

また、商用に供するために、各社で秘密計算のライブラリを整えている。MPC、準同型暗号を用いた方式、TEE を利用した方式など、利用者に合わせて利用できる。これらのライブラリを用いて、ゲノム解析、創薬の実証実験以外にも、

- 商店街の活性化のイベントを行う際に、その費用対効果をしるために、各店舗の売上情報を秘密計算を用いて統合分析する統計値処理
- 銀行における不正取引リスクを分析するために、各銀行が持つリスクデータを統合して分析するリスク分析の AI
- プラントのログ解析

等、多くの応用が検討されている。第 3 項目のプラントのログ解析は、個人情報保護とは直接は関係がない項目であるが、同じ要素技術で解決策を得られる項目例として、ここに挙げている。

■データへのアクセス制御－セキュアなパブリッククラウドの利用　企業、団体、組織（以後、組織）からの種々のサービス提供のみならず、組織内における電子データのやりとりにおいては、電子データは的確に保護されなければならない。これらの目的のために、属性ベース暗号や検索可能暗号と呼ばれる高機能暗号技術を利用して、データの保護、データへのアクセス制御を社会実装したケースがある。これらの事例を以下に紹介する。

例１．組織内文書管理－属性ベース暗号技術の活用例

組織内の共有スペースに電子データを置き、データの共有をはかることはよく行われている。但し、電子データの中には、一部のメンバだけで共有したいデータもある。この場合、共有するデータにアクセス制御をかける必要がある。さらに、暗号化により他のメンバーに閲覧できないようにすることで、強固なアクセス制御が可能となる。以下に紹介する社会実装例では、属性ベース暗号（3.1.2 章参照）を利用して、強固なアクセス制御を実現している。

属性ベース暗号とは、暗号文を復号するための条件を設定し、復号できるユーザを制御できる暗号技術である。復号鍵を所有しないユーザは、当然、暗号文を復号することはできない。さらに、復号鍵を持っているユーザであっても、条件に適合しない場合は復号できない。すなわち、暗号化とデータへのアクセス制御が一体化した技術である。

属性ベース暗号の論文での提案時より、活用例として、企業内や学校内での文書管理が記載されている。具体的には、企業内の役員だけが閲覧できる文書、部長以上の役職であれば閲覧できる文書、ある部署（例えば総務部のみ）の職員だけ

が閲覧できる文書、社員であれば誰でも閲覧できる文書など、文書の閲覧には閲覧できるユーザに条件がつけられていることが多い。ユーザの役職、所属部署などを属性と見て、職員 A さんは属性（人事部、課長）であり、所属 B さんは属性（総務部、部長）を持つとする。そして、職員 A さんには、属性（人事部、課長）に対応する復号鍵が渡され、人事部あての文書や、課長以上が閲覧できる文書、職員全員が閲覧できる文書が閲覧可能となる。職員 B さんは、属性（総務部、部長）に対応する復号鍵が渡され、総務部あての文書や、部長以上が閲覧できる文書、職員全員が閲覧できる文書が閲覧可能となる。文書の作成者は、閲覧者の属性を意識して、文書を暗号化することになる。ここで紹介した暗号技術が属性ベース暗号となる。より正確には、属性ベース暗号には大きく分けて 2 種類、暗号文に条件を入れる場合と、暗号鍵に条件を入れておく場合があり、ここで紹介した事例は暗号鍵に条件を入れた属性ベース暗号となる。

属性ベース暗号のその他の利用方法として、

- 複数の工場や事業所を所有する企業における、ロケーション／役職に応じた文書管理。
- クラウドサービスに文書を保管し、文書毎にアクセス可能な企業を制御できるファイル交換サービス。

などがあり、属性ベース暗号関連のソフトウェアがライブラリとして製品化されている [116, 117]。これらは、Windows ベースで開発されており、各種のインターフェース、ディレクトリサービスとの連携までも考えられている。

例２．DB に保存されたデータの保護 – 検索可能暗号技術の活用例

組織内、組織外を問わず、電子データをサーバに置いて管理することはよく行われている。電子データはサーバ内のデータベース（DB）で管理され、ユーザからのデータ追加や検索などに応じている。通常、サーバもしくはＤＢの管理者はＤＢ内のデータを閲覧できる。悪意のある管理者であれば、データを悪用することは容易である。管理者の善意に頼るのみではなく、技術的な対策も求められる。さらに、近年はサイバー攻撃が多くなり、この対策も求められている。このため、ＤＢのデータを管理者やサイバー攻撃者が知らない鍵によって暗号化することが対策として考えられてきた。さらに、ユーザからの検索キーワードも暗号化することで、何を検索しているのか等の情報も漏洩しないことが求められている。これらの要求に応える技術として、検索可能暗号（3.3.7 章参照）を利用した社会実装例を以下に紹介する。

検索可能暗号とは、データベース（DB）に暗号化されたデータを復号することなく検索できる暗号技術である。DB 内のデータはすべて暗号化されているため、DB が保管されているサーバが攻撃され、データが漏洩したとしても、安全性が担保できる。さらに、検索のためのキーワードも暗号化されているため、暗号化されていない通信路を経由しての検索であっても安全である。さらに、暗号化されたキーワードや、データをサーバ内で復号することがないため、サーバの管理者に対しても安全性が担保される。

従来、DB のデータは平文で管理されていた。もしくは、DB に平文データが入力された後、DB が暗号化を行い、暗号化されたデータを保存していた。前者のように DB で平文を管理する場合、検索も高速である。すなわち、平文の検索キーワードにより検索が行われるためである。ただし、DB にアクセス可能なユーザであったり、サイバー攻撃によって DB の中身が漏洩する場合を考えると、情報漏洩のリスクは高いものであった。また検索キーワードも平文であり、検索キーワードからも情報が漏洩していた。後者のように、DB が暗号化を行う場合は、ＤＢに保存されているデータは安全に保存される。ただし、検索はやや遅くなる。すなわち、データの検索時には、保存されている暗号文を復号した後に、検索キーワードとのマッチングを行い検索結果を出力するため、暗号文を復号する時間が必要となる。さらに、セキュリティの観点からすると、DB の管理者が悪意を持つ場合、および、サイバー攻撃に DB がさらされた場合、情報漏洩のリスクが大きくなる。すなわち、どちらの場合も、ＤＢが保存している秘密鍵の漏洩が考えられる。秘密鍵の漏洩は、ＤＢが保存しているすべての暗号化データが平文データとなって漏洩することと同じ結果となる。また、メモリ解析もセキュリティリスクの一つであり、攻撃として考えられる。すなわち、検索処理は多くの場合、サーバのメモリ上で行われる。このため、管理者やマルウェアがメモリを監視することで平文情報がわかってしまう可能性がある。

これらのセキュリティリスクを回避する目的で開発された技術が検索可能暗号である。検索可能暗号を利用することで、ＤＢに保存されているデータは暗号化され、さらに、復号鍵をＤＢは所持していない。すなわち、ＤＢの管理者が悪意を持って情報を漏洩させた場合、および、サイバー攻撃にＤＢがさらされたとしても、漏洩するのは暗号化されたデータだけであり、復号鍵は漏洩しない。さらに、検索キーワードも暗号化されており、ＤＢ内で復号されることはないため、検索キーワードからの情報漏洩もない。

検索可能暗号の実装例として、

- クラウドにＤＢ管理を委託した場合に、ＤＢ内のデータを検索するセキュア

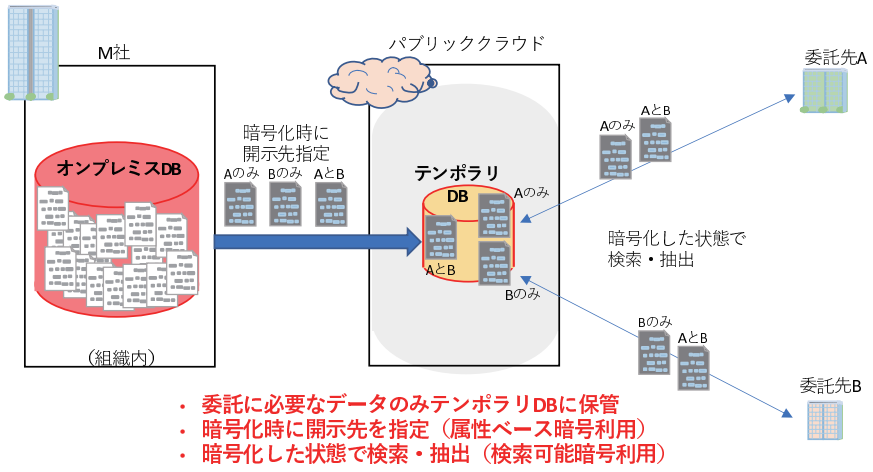


図 2.3　社会実装例：パブリッククラウドを用いたデータ共有

なアクセスを可能とする秘匿検索サービス

などがあり、検索可能暗号関連のソフトウェアがライブラリとして製品化されている [118]。

例 3．パブリッククラウドを用いたデータ共有

多くの企業、団体、組織（以後、組織）では協業の効率化等の目的でパブリッククラウドを利用する機会が多くなっている。例えば、自組織のサービスをパブリッククラウド上で展開したり、逆に、他組織のサービスを利用する際に利用することもある。さらに、パブリッククラウド上で展開されている機能／ライブラリ群なども多くなったこともあり、パブリッククラウドの活用が加速されている。このような状況下で、パブリッククラウドに置かれるデータの安全性を担保する目的で、属性ベース暗号と検索可能暗号の両方を社会実装したケースがあるので紹介する。

複数の組織で電子データを共有する場合にパブリッククラウドを利用することで、サービスの効率化を図れる場合がある。組織が所有する電子データが少ない場合、他組織と共有する電子データの量が少なく、関連する他組織の数も少なくなる。このような場合には、情報を共有するとしても、メールにデータを添付したり、記録媒体で渡すなどの、小規模な手段で情報共有が可能であった。これに対し、近年では、組織が持つ電子情報が膨大になり、他組織と共有する電子データの量が多くなり、関連する他組織の数も多くなり、状況が大きく変わっている。

例えば（図 2.3 参照）、ある組織（M 社）が業務効率化のために、他組織（A 社）と別の他組織（B 社）に作業委託を行うことがあり、M 社と A 社、B 社は機密デー

タや顧客データを共有することもある。そして、A 社と共有する情報と B 社と共有するデータには、共通のデータもあり、異なるデータもある。そして、M 社は、社内にオンプレミス型の巨大な DB（オンプレ DB）が構築されていることはよくある状況である。委託先 A 社、B 社は M 社にとっては他組織であるため、M 社のオンプレ DB に A 社や B 社がアクセスすることは、委託に関係しないデータへのアクセスに対する危惧もあり、セキュリティ上の理由で好まれない。このため、パブリッククラウド（以後、サーバ）を利用して、オンプレ DB の中の共有すべきデータだけをサーバに置き、テンポラルな DB（テンポラリ DB）を作成し、A 社、B 社とデータ共有を行うことが考えられている。

テンポラリ DB を利用し、オンプレ DB 内の必要最小限のデータ共有であったとしても、M 社にとっては機密データであるため、このデータ共有を安全に行うことは重要である。さらに、利用するサーバの管理者は、容易にテンポラリ DB に置かれたデータにアクセスできる。さらには、サーバがパブリックであるが故に、サーバの管理者、M 社、A 社、B 社とは関係のない、不特定多数のユーザがアクセスする可能性もある。

このような状況であっても、テンポラリ DB に置かれた機密データを保護するために、属性ベース暗号と検索可能暗号を利用する。M 社が共有するデータをテンポラリ DB に置く際に、委託に必要なデータを抽出して、“A 社のみが閲覧できるデータ”、“B 社のみが閲覧できるデータ”、“A 社と B 社の両方が閲覧できるデータ” の 3 種類に分類する。そして、属性ベース暗号を利用して、“属性 A”、“属性 B”、“属性 A または属性 B” の属性を持つ復号鍵であれば、復号できるようにそれぞれのデータに対して暗号化を行い、暗号文をテンポラリ DB に置く。そして、A 社には “属性 A” を持つ復号鍵を、B 社には “属性 B” を持つ復号鍵を配布しておく。また、これらのデータを検索できるように、共有するデータからキーワードを抽出し、検索可能暗号を用いて、キーワードを暗号化し、テンポラリ DB に置く。A 社、B 社はテンポラリ DB のデータを閲覧する際、まず、検索キーワード（クエリ）を検索可能暗号を用いて暗号化して、暗号化クエリをテンポラリ DB に送る。テンポラリ DB では、検索可能暗号の検索機能を用いて検索結果を出力し、A 社もしくは B 社に返送する。

検索結果を受領した A 社もしくは B 社では、属性ベース暗号の復号鍵を用いて、受領したデータを復号するが、A 社が持つ復号鍵は “属性 A” を持ち、B 社が持つ復号鍵は “属性 B” を持つ。このため、A 社は、“A 社のみが閲覧できるデータ” と “A 社と B 社の両方が閲覧できるデータ” を復号することができるが、“B 社のみが閲覧できるデータ” を復号することはできない。逆に、B 社は、“B 社の

みが閲覧できるデータ” と “A 社と B 社の両方が閲覧できるデータ” を復号することができるが、“A 社のみが閲覧できるデータ” を復号することはできない。

この一連の属性ベース暗号と検索可能暗号の実装により、データ閲覧に対するアクセス制御と、いわゆる秘匿検索、暗号化したままでの検索が可能となり、テンポラリ DB 内データに対する不要なデータ閲覧を防ぎ、キーワードからの情報漏洩を防いでいる。

参考文献

[1] CRYPTREC: “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”. https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2012r6.pdf.

[2] S. Mitsunari: “MCL: a Portable and Fast Pairing-Based Cryptography Library”. GitHub Repository.

[3] Stanford University: “The Pairing-Based Cryptography Library”. https://crypto.stanford.edu/pbc/.

[4] Voltage: “Voltage SecureMail”. https://www.microfocus.com/en-us/cyberres/data-privacy-protection/secure-mail.

[5] N. Suilvan: “Geo Key Manager: How It Works”. The Cloudflare Blog, https://blog.cloudflare.com/geo-key-manager-how-it-works/.

[6] IETF: “RFC 5091: Identity-Based Cryptography Standard (IBCS)#1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems”.

[7] IETF: “RFC 5408: Identity-Based Encryption Architecture and Supporting Data Structures (January 2009)”.

[8] IETF: “RFC 5409：Using the Boneh-Franklin and Boneh-Boyen Identity-Based Encryption Algorithms with the Cryptographic Message Syntax (CMS)”.

[9] IETF: “RFC 6267：MIKEY-IBAKE: Identity-Based Authenticated Key Exchange (IBAKE) Mode of Key Distribution in Multimedia Internet KEYing (MIKEY)”.

[10] IETF: “RFC 6539：IBAKE: Identity-Based Authenticated Key Exchange”.

[11] IETF: “RFC 6507：Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)”.

[12] IETF: “RFC 6508：Sakai-Kasahara Key Encryption (SAKKE)”.

[13] IEEE: “IEEE Std 1363.3-2013 - IEEE Standard for Identity-Based Cryptographic Techniques using Pairings”.

[14] ISO/IEC: “ISO/IEC 18033-5:2015 - Identity-Based Ciphers”.

[15] D. K. Smetters and G. Durfee: “Domain-Based Administration of Identity-Based Cryptosystems for Secure Email and IPSEC”, USENIX Security Symposium 2003 (2003).

[16] S. Ruoti, N. Kim, B. Burgon, T. W. vander Horst and K. E. Seamons: “Confused Johnny: When Automatic Encryption Leads to Confusion and Mistakes”, Usable Privacy and Security, ACM (2013).

[17] S. Ruoti, J. Andersen, T. Hendershot, D. Zappala and K. E. Seamons: “Private Webmail 2.0: Simple and Easy-to-Use Secure Email”, User Interface Software and Technology, ACM, pp. 461–472 (2016).

[18] S. Meiklejohn, K. Mowery, S. Checkoway and H. Shacham: “The Phantom Tollbooth: Privacy-Preserving Electronic Toll Collection in the Presence of Driver Collusion”, USENIX security symposium, Vol. 201, No. 1 (2011).

[19] L. Veltri, G. Morabito, S. Salsano, N. Blefari-Melazzi and A. Detti: “Supporting Information-Centric Functionality in Software Defined Networks”, IEEE ICC 2012 (2012).

[20] M. Santos, B. D. Oliveira, C. Margi, B. Astuto, T. Turletti, I. S. Antipolis and K. Obraczka: “Software-Defined Networking Based Capacity Sharing in Hybrid Networks”, IEEE ICNP 2013 (2013).

[21] M. Santos, B. Astuto, K. Obraczka, T. Turletti, B. D. Oliveira and C. Margi: “Decentralizing SDN’s Control Plane”, IEEE Conference on Local Computer Networks 2014 (2014).

[22] 三菱電機 IT ソリューションズ：“パッケージプラス (R) トランスポーター”. https://www.mdsol.co.jp/products/general_affairs/transporter.html.

[23] テキサス大学：“Ciphertext-Policy Attribute-Based Encryption：PBC (Pairing-Based Cryptography) ライブラリ（C ライブラリ）を利用したライブラリ”. http://acsc.cs.utexas.edu/cpabe/.

[24] “Java Pairing-Based Cryptography Library (JPBC): [GGH+13] 方式を利用した Java による高機能暗号ライブラリ”. http://gas.dia.unisa.it/projects/jpbc/.

[25] “OpenABE: [27] 方式の変形版を利用した C/C++ オープンソースライブラリ”. https://github.com/zeutro/openabe.

[26] S. Garg, C. Gentry, S. Halevi, A. Sahai and B. Waters: “Attribute-Based

Encryption for Circuits from Multilinear Maps", CRYPTO 2013, Vol. 2, Springer-Verlag, pp. 479–499 (2013).

[27] V. Goyal, O. Pandey, A. Sahai and B. Waters: "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data", ACM CCS 2006, pp. 89–98 (2006).

[28] AACS: "Advanced Access Content System (AACS) Introduction and Common Cryptographic Elements Book". https://aacsla.com/wp-content/uploads/2019/02/AACS_Spec_Common_Final_0953.pdf.

[29] D. Naor, M. Naor and J. Lotspiech: "Revocation and Tracing Schemes for Stateless Receivers", CRYPTO 2001, Springer-Verlag, pp. 41–62 (2001).

[30] https://aacsla.com/.

[31] NICT, 神戸大学, エルテス："プライバシー保護深層学習技術を活用した不正送金検知の実証実験において金融機関 5 行との連携を開始". https://www.nict.go.jp/press/2020/05/19-1.html, https://www.kobe-u.ac.jp/research_at_kobe/NEWS/collaborations/2020_05_19_01.html.

[32] "DeepProtect". https://deepprotect.nict.go.jp/.

[33] L. T. Phong, Y. Aono, T. Hayashi, L. Wang and S. Moriai: "Privacy-Preserving Deep Learning via Additively Homomorphic Encryption", **Vol.13, No.5**, pp. 1333–1345 (2018).

[34] 日本経済新聞："Suica 履歴を「秘密計算」データの中身、読まずに分析". https://www.nikkei.com/article/DGXZQOUC233NT0T20C21A6000000/.

[35] P. Paillier: "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes", EUROCRYPT 1999, Springer-Verlag, pp. 223–238 (1999).

[36] Z. Brakerski, C. Gentry and V. Vikuntantathan: "(Leveled) Fully Homomorphic Encryption without Bootstrapping", ITCS 2012, ACM, pp. 309–325 (2012).

[37] J. Cheon, A. Kim, M. Kim and Y. Song: "Homomorphic Encryption for Arithmetic of Approximate Numbers", ASIACRYPT 2017, Vol. 1, Springer-Verlag, pp. 409–437 (2017).

[38] "HElib (Homomorphic Encryption library)". https://github.com/homenc/HElib.

[39] Z. Brakerski: "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP", CRYPTO 2012, Springer-Verlag, pp. 868–886 (2012).

[40] J. Fan and F. Vercauteren: “Somewhat Practical Fully Homomorphic Encryption”. ePrint 2012/144.
[41] “Simple Encrypted Arithmetic Library”. https://www.microsoft.com/en-us/research/project/microsoft-seal/.
[42] “HEAAN”. https://github.com/snucrypto/HEAAN.
[43] L. Ducas and D. Micciancio: “FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second”, EUROCRYPT 2015, Springer-Verlag, pp. 617–640 (2015).
[44] I. Chillotti, N. Gama, M. Georgieva and M. Izabachène: “Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds”, ASIACRYPT 2016, Springer-Verlag, pp. 3–33 (2016).
[45] “PLISADE”. https://palisade-crypto.org/.
[46] “Lattigo”. https://github.com/ldsec/lattigo.
[47] “CRYPTOLAB”. https://www.cryptolab.co.kr/eng/product/heaan.php.
[48] “Homomorphic Encryption Standardization”. https:homomorphicencryption.org/standard/.
[49] ISO/IEC: “ISO/IEC 18033-6:2019 - Homomorphic Encryption”. https://www.iso.org/obp/ui/#iso:std:iso-iec:18033:-6:ed-1:v1:en.
[50] NuCypher: “Umbral: A Threshold Proxy Re-Encryption Scheme”. https://raw.githubusercontent.com/nucypher/umbral-doc/master/umbral-doc.pdf.
[51] ITmedia: “東芝「デジタル貸金庫」”. https://www.itmedia.co.jp/pcuser/articles/1211/21/news053.html.
[52] 川合, 小関, 柴田, 大松：“属性ベース署名を用いたブロックチェーン”, DICOMO2017, 情報処理学会, pp. 669–671 (2017). https://ipsj.ixsq.nii.ac.jp/ej/?action=repository_uri&item_id=190099&file_id=1&file_no=1.
[53] 堀川, 高谷, S. Z. Fazekas, 山村：“属性ベース署名を用いた匿名シングルサインオンの提案” (2017). https://www.kurims.kyoto-u.ac.jp/~kyodo/kokyuroku/contents/pdf/2051-26.pdf.
[54] S. Hirose and J. Shikata: “Non-Adaptive Group-Testing Aggregate MAC Scheme”, ISPEC 2018, Springer-Verlag, pp. 357–372 (2018).
[55] S. Sato and J. Shikata: “Interactive Aggregate Message Authentication

Scheme with Detecting Functionality", AINA 2019, Springer Nature, pp. 1316–1328 (2019).

[56] ITU-T: "Recommendation ITU-T X.1366: Aggregate message authentication schemes for Internet of things environment". https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=14262&lang=en.

[57] 日本ネットワーク・オペレーターズ・グループ："アグリゲート署名を用いたbgpsecの改良". https://www.janog.gr.jp/meeting/janog43/program/bgpsec/.

[58] ARMORY: "Armory Secure Wallet". https://www.bitcoinarmory.com/.

[59] "ERC: Standard API for multisig wallet smart contracts #763". https://github.com/ethereum/EIPs/issues/763.

[60] ISO/IEC: "ISO/IEC 20008-2:2013 - Information technology - Security techniques — Anonymous digital signatures — Part 2: Mechanisms using a group public key". https://www.iso.org/standard/56916.html, https://www.iso.org/obp/ui/#iso:std:iso-iec:20008:-2:ed-1:v2:en.

[61] ISO/IEC: "ISO/IEC 11889 - ISO/IEC 11889-1:2015 - Information technology —Trusted platform module library — Part 1: Architecture". https://www.iso.org/standard/66510.html.

[62] S. Noether: "Ring Confidential Transactions". ePrint 2015/1098.

[63] P. Wuille, J. Nick and T. Ruffing: "Bitcoin Improvement Proposal 340" (2020). https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki.

[64] K. McCusker and B. Spector: "Qredo Network" (2020). https://www.qredo.com/qredo-yellow-paper.

[65] J. Doerner, Y. Kondi, E. Lee and A. Shelat: "Secure Two-party Threshold ECDSA from ECDSA Assumptions". ePrint 2018/499.

[66] フレセッツ株式会社："世界初、フレセッツがhsmを活用した完全オフライン環境でのセキュアなマルチシグ実装に成功" (2020). https://prtimes.jp/main/html/rd/p/000000017.000028896.html.

[67] Coinbase 株式会社："What is a Dapp Wallet". https://help.coinbase.com/en/coinbase/trading-and-funding/trade-on-dex/what-is-a-dapp-wallet.

[68] DFINITY: "Threshold ECDSA Signatures". https://forum.dfinity.

org/t/threshold-ecdsa-signatures/6152.

[69] NIST: “Threshold Schemes for Cryptographic Primitives: Challenges and Opportunities in Standardization and Validation of Threshold Cryptography”. https://csrc.nist.gov/publications/detail/nistir/8214/final.

[70] NIST: “NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives”. https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8214A.pdf.

[71] NIST: “NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives”. https://csrc.nist.gov/news/2022/nist-requests-comments-on-ir-8214b-initial-public.

[72] IETF: “Two-Round Threshold Signatures with FROST”. https://datatracker.ietf.org/doc/draft-irtf-cfrg-frost/.

[73] 保坂, 多田, 加藤：“秘密分散法とその応用”, 東芝レビュー, **62**, 7 (2007).

[74] “NRI: Windows Azure と秘密分散法を利用したセキュリティ性の高い「世界分散ストレージサービス」”, https://cloud.watch.impress.co.jp/docs/column/cloud/493419.html.

[75] S. Shin, S. Yamada, G. Hanaoka, Y. Ishida, A. Kunii, J. Oketani, S. Kunii and K. Tomomura: “An Extended CTRT for AES-256”, WISA 2019, Springer-Verlag, pp. 79–91 (2019).

[76] “ZENMU-AONT に関する論文が 国際会議で best paper premium award を受賞”, https://www.atpress.ne.jp/news/191661.

[77] “Alchemy - What is MPC wallet?”, https://www.alchemy.com/overviews/mpc-wallet.

[78] “ISO/IEC 19592-2:2017 Information technology - Security techniques - Secret sharing - Part 2: Fundamental mechanisms”.

[79] R. Cleve, D. Gottesman and H.-K. Lo: “How to Share a Quantum Secret”, Physical Review Letters, **83**, 3, pp. 648–651 (1999).

[80] G. R. Blakley and C. Meadows: “Security of ramp schemes”, CRYPTO 1984, Springer-Verlag, pp. 242–268 (1984).

[81] 山本：“(k, L, n) しきい値秘密分散システム”, 電子通信学会論文誌, **J68-A**, 9, pp. 945–952 (1985).

[82] M. Itoh, A. Saito and T. Nishizeki: “Secret Sharing Scheme Realizing General Access Structure”, IEEE Globecom 1987, pp. 99–102 (1987).

[83] R. Cramer, I. Damgård and Y. Ishai: "Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation", TCC 2005, Springer-Verlag, pp. 342–362 (2005).

[84] T. Araki, J. Furukawa, Y. Lindell, A. Nof and K. Ohara: "High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority", ACM CCS 2016, pp. 805–817 (2016).

[85] K. Chida, D. Genkin, K. Hamada, D. Ikarashi, R. Kikuchi, Y. Lindell and A. Nof: "Fast Large-Scale Honest-Majority MPC for Malicious Adversaries", CRYPTO 2018, Springer-Verlag, pp. 34–64 (2018).

[86] R. Kikuchi, K. Chida, D. Ikarashi, W. Ogata, K. Hamada and K. Takahashi: "Secret Sharing with Share-Conversion: Achieving Small Share-Size and Extendibility to Multiparty Computation", IEICE Trans., **E98-A(1)**, pp. 213–222 (2015).

[87] H. Krawczyk: "Secret sharing made short", CRYPTO 1993, Springer-Verlag, pp. 136–146 (1993).

[88] 千葉大学病院, NTT Com："秘密計算ディープラーニング」などの技術を活用した臨床データ分析". https://www.ntt.com/about-us/press-releases/news/article/2021/0208.html.

[89] NEC, 大阪大学："医療データ活用". https://jpn.nec.com/rd/technologies/201805/index.html.

[90] CYBERNETICA: "収入と教育履歴の分析". https://sharemind.cyber.ee/big-data-analytics-protection/.

[91] Cosmian: "エネルギー供給の最適化". https://cosmian.com/use-case/.

[92] MPC Alliance. https://www.mpcalliance.org/.

[93] 秘密計算コンソーシアム. https://secure-computation.jp/.

[94] 秘密計算研究会. https://securecomputing.jp/.

[95] NIST: "NIST Project on Privacy-Enhancing Cryptography (PEC)". https://csrc.nist.gov/projects/pec.

[96] ISO/IEC: "ISO/IEC 4922-1/2 Information security — Secure multiparty computation". https://www.iso.org/standard/80508.html.

[97] Curv. https://medium.com/@william_97682/curv-mpc-is-not-key-sharding-276b020130c4.

[98] INGBank. https://www.ingwb.com/en/insights/distributed-ledger-technology/

ing-launches-major-addition-to-blockchain-technology.

[99] https://z.cash/technology/.

[100] Ernst & Young. https://www.ey.com/en_gl/news/2019/12/ey-releases-third-generation-zero-knowledge-proof-blockchain-\technology-to-the-public-domain, https://assets.ey.com/content/dam/ey-sites/ey-com/ja_jp/topics/library/info-sensor/2020/12/pdf/ey-japan-info-sensor-2020-12-08.pdf.

[101] ZKProof Standards. https://zkproof.org/, https://docs.zkproof.org/pages/reference/reference.pdf.

[102] TechCrunch: "Kalepso Looks to Break into The Crowded Encrypted Database Space" (2018). https://techcrunch.com/2018/11/29/kalepso-database-encryption-battlefield-berlin/.

[103] Coinspeaker: "Decentralized Service Protocol, Automata Network, Launches a Solution for the Miner Extractable Value (MEV) Issue" (2021). https://www.coinspeaker.com/automata-network-miner-extractable-value-conveyor/.

[104] S. Wang, X. Ding, R. H. Deng and F. Bao: "Private Information Retrieval Using Trusted Hardware", ESORICS 2006, Springer-Verlag, pp. 49–64 (2006).

[105] M. Melucci and R. Baeza-Yates: "Advanced Topics in Information Retrieval", Vol. 33, Springer (2011).

[106] D. Augot, F. Levy-dit Vehel and A. Shikfa: "A Storage-Efficient and Robust Private Information Retrieval Scheme Allowing Few Servers", CANS 2022, Springer International Publishing, pp. 222–239 (2022).

[107] "事業者向け暗号資産ウォレット開発のフレセッツ、暗号資産ウォレットユーザに対するプライバシー向上技術の開発に成功". https://prtimes.jp/main/html/rd/p/000000014.000028896.html.

[108] "Microsoft sealpir". https://github.com/microsoft/SealPIR.

[109] MIT: "CryptDB". http://css.csail.mit.edu/cryptdb/.

[110] "Kafedb: End-to-end structually-encrypted relational database system". https://zheguang.github.io/kafedb/.

[111] 日立製作所："秘匿情報管理サービス 匿名バンク". https://www.hitachi.co.jp/products/it/harmonious/cloud/service/tokumei/index.html.

[112] 三菱電機：“部分一致対応秘匿検索基盤ソフトウェア”. https://www.mitsubishielectric.co.jp/corporate/randd/list/info_tel/a29/index.html.

[113] G. Amjad, S. Kamara and T. Moataz: “Forward and Backward Private Searchable Encryption with SGX”, European Workshop on Systems Security 2019, Vol. 4, pp. 1–6 (2019).

[114] NEC, 大阪大学：“複数機関が保有するゲノム情報をプライバシー侵害リスクを抑えて解析できることを実証”. https://jpn.nec.com/press/201907/20190723_03.html.

[115] NEC: “連合学習技術と秘密計算技術を用いた創薬における予測モデル構築に関する実証実験を実施”. https://jpn.nec.com/press/202203/20220311_01.html.

[116] 三菱電機：“三菱情報セキュリティソフトウェア fefileprotection”. https://www.mitsubishielectric.co.jp/business/security/fileprotect/index.html.

[117] 三菱電機：“関数型暗号とクラウドサービスを利用した機密情報ファイル交換サービス「パッケージプラス (r) トランスポーター」の提供開始”. https://www.mdsol.co.jp/newsrelease/doc/information_20160125.pdf.

[118] 三菱電機：“Mistyguard ＜検索可能暗号＞ cizouxlib を開発”. https://www.mdis.co.jp/news/press/2021_1028.html.

第 3 章

主な高機能暗号技術のアルゴリズム・プロトコルとその性能

3.1 守秘を目的とした高機能暗号技術

3.1.1 ＩＤベース暗号

■特徴 ID ベース暗号（Identity-based encryption, IBE）は、ID をはじめとする任意の文字列を公開鍵に利用可能とする公開鍵ベースの暗号化手法である。1984 年にそのコンセプトを提案した Shamir [1] により、公開鍵の情報として ID 情報を利用可能とすることから ID ベース暗号と名付けられたが、本質的には ID 情報の代表的な具体例である電子メールアドレスやその他の識別情報などが利用可能であり、さらには ID 情報のみならず任意の情報を公開鍵の情報として利用可能である。公開鍵である ID 情報から容易にその所有者を特定することができることから、（通常の公開鍵暗号では必要となる）公開鍵暗号基盤 （Public-Key Infrastructure, PKI） が発行する公開鍵証明書が不要となる。一方で、公開鍵暗号とは異なり、IBE では暗号の作成者（送信者）と秘密鍵をもつユーザ以外に、各ユーザの鍵生成を行う鍵生成センタ（Private-Key Generator, PKG） が存在する。このような機関を想定しなければ、誰でも任意の ID に対して秘密鍵を生成可能となり、安全性が担保できない。また、IBE を実現するうえで最も重要な安全性が結託耐性である。結託耐性とは、多数のユーザが結託し自身の ID の秘密鍵を共有したとしても、結託に関与していないユーザの ID 宛の暗号文の解読が難しいことを保証する安全性である。

さらに、IBE は他の暗号技術を構成するための暗号要素技術として利用価値が高い。例えば、IBE を要素技術として用いることで、秘密鍵の漏洩に耐性のある鍵隔離暗号 [2] や前方秘匿暗号 [3]、暗号文が復号可能となる期間を指定可能な時刻特定暗号 [4]、強い安全性を満たす公開鍵暗号 [5] 等、様々な暗号技術を実現できることが知られている。

■方式の歴史・進展 結託耐性を持つ IBE の効率的な実現は難しく、Shamir によって IBE のコンセプトが提唱された 1984 年から 15 年以上が経過して初めて、2000 年に境-大岸-笠原により ID ベースの鍵共有方式 [6] が日本国内で提案された。さらに、2001 年に Boneh-Franklin による IBE 方式 [7] が海外で提案された。これらの方式の特筆すべき点は、Pairing 写像が初めて暗号構成に利用されたことである。Pairing 写像は 2 入力 1 出力の写像であり、暗号用途では楕円曲線上の Pairing 写像を用いる。特に、Pairing 写像のもつその双線形性によって、IBE を始めとする多くの高機能暗号技術の実現が可能となった（双線形性については具体的な構成例で説明する）実際、Pairing 写像の登場を皮切りに多くの（高機能）暗号技術の

構成研究が発展することとなった。

IBE 研究においても、Pairing 写像を用いた構成手法が現実的な IBE の実現方法として主流であり、活用事例や標準化のほとんどに Pairing 写像を用いた構成手法が採用されている。Boneh-Franklin IBE の提案以降の主要な成果として、まず 2004 年の IBE 方式 [8] が挙げられる。この方式は Boneh-Franklin IBE とは異なり（現実には存在し得ない）理想的なハッシュ関数が構成に不要な一方で、保証できる安全性が弱く、現実的な状況（適応的な攻撃に対する安全性）を捉え切れていなかった。その後、2005 年に（理想的なハッシュ関数なしで）適応的な攻撃に対して安全な方式 [9]、2006 年には安全性の根拠となる計算量仮定は強いものの適応的な攻撃に対して安全かつ効率的なパラメータサイズを達成する方式 [10] と、立て続けに主要成果が提案された。Boneh-Franklin IBE に続くブレークスルーとなった研究が、Waters [11] による Dual System Encryption と呼ばれる証明技法の開発である。安全性証明にのみ登場し実際には利用しない “不完全な” 暗号文や秘密鍵を定義することで、標準的な計算量仮定の下で適応的な攻撃に対して安全かつ効率的なパラメータサイズを達成する方式を初めて提案した。このように、Pairing 写像ベース IBE 方式は効率的なパラメータサイズを達成し得るが、一方で Pairing 写像演算それ自体の計算負荷が高いという欠点がある。

また、Pairing 写像以外を用いる IBE の構成手法として有名なものに、格子構造を用いたものがある。代表的な格子ベース IBE 方式 [12, 13] は基本的に Pairing 写像ベース IBE 方式の構成アプローチを基にして作られており、格子構造のもつ耐量子計算機性が最大の魅力である。効率性の観点では、格子ベース IBE 方式はパラメータサイズが大きくなるものの、暗号化や復号が基本的に行列演算で実現可能という特長がある。格子以外の構成手法として、古典的な乗法巡回群を用いた Döttling-Garg による IBE 方式 [14] も重要な IBE 研究の一つである。Pairing 写像の登場まで IBE が実現されなかったことからも、IBE は通常の公開鍵暗号で利用されるような基本的な代数的構造での実現は難しいと考えられていたが、この研究がその認識を覆すこととなり、IBE の実現可能性の意味で大きなブレークスルーとなった。

学術研究としては更に高機能な ID ベース暗号が提案されており、PKG の秘密鍵生成の負担の分散を目的とした階層型（Hierarchical）IBE [15] や、秘密鍵が部分的に漏洩したとしても安全性を保証可能な漏洩耐性（Leakage-resilient）IBE [16]、効率的に秘密鍵の失効を実現可能な鍵失効機能付き（Revocable）IBE [17] 等、想定される機能は多岐にわたる。

■具体的な構成例　具体的な IBE 方式として、最も基本的かつ標準化もされている Boneh-Franklin IBE のアルゴリズムを紹介する。各ユーザは個別のユニーク ID（ID）を持つものとする。

Setup:　公開パラメータ MPK とマスター鍵 MSK を設定する。素数位数 p の乗法群を $\mathbb{G}_1$, $\mathbb{G}_T$ とし、$\mathbb{G}_1$ の生成元を g とする。また、Pairing 写像を $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ とする。e は双線形性を満たす。すなわち、任意の $a, b \in \mathbb{Z}_p$ に対して $e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab}$ を満たす。ランダムな元 $s \in \mathbb{Z}_p^*$ を選択し、$g_{pub} = g^s$ を計算する。また、ハッシュ関数 $H_1 : \{0,1\}^* \to \mathbb{G}_1$ と $H_2 : \mathbb{G}_T \to \{0,1\}^n$ を定めておく。公開パラメータ MPK を $MPK = (p, \mathbb{G}_1, \mathbb{G}_T, e, n, g, g_{pub}, H_1, H_2)$、PKG のマスター秘密鍵 MSK を $MSK = s$ とする。

Extract(MPK, MSK, ID):　ユーザ鍵 SK_{ID} を生成する。$ID \in \{0,1\}^*$ から $Q_{ID} = H_1(ID) \in \mathbb{G}_1$ を計算し、ユーザ鍵 $SK_{ID} = Q_{ID}^s$ とする。

Encrypt(MPK, ID, M):　メッセージ M の暗号化を行い、暗号文 C を生成する。$g_{ID} = e(Q_{ID}, g_{pub}) \in \mathbb{G}_T$ を計算し、ランダムな元 $r \in \mathbb{Z}_p^*$ を選択する。その後、メッセージ $M \in \{0,1\}^n$ から暗号文 $C = (g^r, M \oplus H_2({g_{ID}}^r))$ を作成する。

Decrypt(SK_{ID}, C):　暗号文 C を復号し、メッセージ M を復号する。暗号文 $C = (U, V)$ に対して、$V \oplus H_2\left(e(SK_{ID}, U)\right)$ を出力する。この時、正しい暗号文であれば $(U, V) = (g^r, M \oplus H_2({g_{ID}}^r))$ であり、$e(SK_{ID}, U) = e(Q_{ID}^s, g^r) = e(Q_{ID}, g^s)^r = g_{ID}^r$ であるから、$V \oplus H_2(e(SK_{ID}, U) = M$ が成り立つ。

3.1.2　属性ベース暗号

■特徴　属性ベース暗号（Attribute-Based Encryption, ABE）は、暗号文または秘密鍵の一方に属性情報を、もう一方にアクセス構造を持たせ、復号時に属性情報に含まれる値がアクセス構造の条件を満たすかどうか判別することで、復号者の制御が実現可能な公開鍵暗号方式の一つである。一般に（狭義の）公開鍵暗号や ID ベース暗号（IBE）においては、暗号文とそれを復号可能な秘密鍵は 1 対 1 の関係を持つため、暗号文の生成時に復号できるエンティティは唯一に決定され、もし複数のエンティティに同一のメッセージを送付する場合には、エンティティ数に比例する暗号文の生成が必要である。一方で、属性ベース暗号ではアクセス構造を指定することで、単一の暗号文でメッセージの秘匿ときめ細かなアクセス

制御を同時に達成可能となる。表現可能なアクセス構造は用いる暗号方式により様々であるが、多くの方式では例えば AND, OR で記述された条件式を表現することができる。

属性ベース暗号の方式は、鍵ポリシ型（Key-Policy, KP）と暗号文ポリシ型（Ciphertext-Policy, CP）の 2 種類の型に大別される。鍵ポリシ型ではエンティティが持つ秘密鍵に、暗号文ポリシ型ではメッセージにそれぞれアクセス構造を埋め込み、もう一方に属性情報を埋め込む点が異なる。そして、適応先のユースケースによってより効果的な型を選択する必要があり、例えば、ユーザや機器等のエンティティが固有に有する情報を元に制御したい場合には暗号文ポリシ型、暗号化対象のメッセージが固有に有する情報を元に制御したい場合には鍵ポリシ型の方式が適していることが多い。

一般に属性ベース暗号方式を構成する場合、セットアップおよび鍵発行を実行する主体である鍵生成局と、暗号化および復号を実行する主体であるエンティティが存在する。エンティティは、自身の有する属性情報（またはアクセス構造）を含む秘密鍵を鍵発行局から発行してもらい、自身のみが扱うように安全に保管する。暗号化については、公開情報を知ることができる任意のエンティティが実行することができ、メッセージにアクセス構造（または属性情報）を指定して暗号化する。復号については、エンティティが各自の秘密鍵と暗号文を用いて復号を行うが、属性情報とアクセス構造の関係が満足する場合には元のメッセージに復号できる。

■方式の歴史・進展　属性ベース暗号は、2005 年に ID ベース暗号の拡張である Fuzzy IBE が Sahai らにより提案され、そのコンセプトが提唱された [18]。Fuzzy IBE では、秘密鍵および暗号文にそれぞれ複数の ID が割り当てられており、各 ID 集合の間で定められたしきい値以上の ID が一致した場合にのみ復号可能である。この複数の ID を属性集合、一致する ID とそのしきい値をアクセス構造と見なすことで、現在の属性ベース暗号の一種と捉えられる。翌年 2006 年、Goyal らにより、より広いクラスの条件式を扱うことができるはじめての実用的な属性ベース暗号が提案された [19]。Goyal らの方式は鍵ポリシ型であり、秘密鍵に対して AND, OR, しきい値を用いた論理式で表現された条件式を設定し、また暗号文に属性集合を指定することで、暗号文に含まれる属性値が自身の持つ秘密鍵の条件式を満たすかにより、きめ細やかな復号者の制御が可能となった。さらに 2007 年には、Bethencourt らにより、暗号文に対してアクセス構造を設定することが可能な暗号文ポリシ型の方式がはじめて提案された [20]。

その後、安全性の研究においては、適応的攻撃に対しても安全な方式 [21] や標準モデルと呼ばれる現実的な仮定の下で安全な方式 [22] の研究等が進められ、ま

た量子計算機に対して安全な格子ベースの方式も提案されている [23]。条件式で使えるクラス拡張の研究においては、数値の大小比較 [24]、NOT 条件 [25] を利用可能な方式等、数多くの方式が提案されている。

さらに、前述した基本的な性質以外の機能性を追加した方式も提案されてきた。例えば、鍵発行処理をより柔軟にするものとして、ある秘密鍵で復号可能な暗号文のうち、その中の一部の暗号文のみを復号できる新たな秘密鍵を生成し、権限を移譲できる方式 [19] や鍵発行局を階層化した方式 [26]、複数の鍵発行局の秘密鍵を用いて復号制御ができるマルチオーソリティ方式 [27]、さらに特権的なルートを持たない分散型マルチオーソリティ方式 [28] 等がある。

■具体的な構成例: GPSW06 KP-ABE　本節では、AND, OR, および (k, n)-しきい値を条件式として記述でき、はじめてきめ細かなアクセス制御が可能となった KP-ABE である Goyal らによる GPSW06 の構成を示す [19]。

はじめに本方式のアクセス構造の表現方法であるアスセスツリー $\mathcal{T}$ について説明する。アスセスツリー $\mathcal{T}$ の葉ノード以外の全てのノードはしきい値ゲートを表現できる。ノード x に対して、子ノードの個数を num_x、しきい値を $1 \leq k_x \leq num_x$ である値 k_x とする。$k_x = 1$ の場合には OR ゲートを、$k_x = num_x$ の場合には AND ゲートを、$1 < k_x < num_x$ の場合には (k_x, num_x)-しきい値ゲートを表現できる。各ノード x の子ノードは順序付けされており、それぞれ 1 から num_x までのインデックスが割り振られている。また葉ノードでは属性値が設定されており、$k_x = 1$ とする。

アスセスツリー中のノードの取り扱いを容易にするため、いくつかのサブ関数を準備する。あるノード x に対して、$\mathtt{parent}(x)$ を x の親ノード、$\mathtt{index}(x)$ を親ノード $\mathtt{parent}(x)$ から割り振られたインデックスを指す値とする。$\mathtt{parent}(x)$ の各子ノードはこのインデックスで一意に特定できる。また $\mathtt{att}(x)$ を x の属性値で、x が葉ノードのときにのみ値を取るものとする。

以降では、素数 p に対して、位数 p の乗法群を $\mathbb{G}_1, \mathbb{G}_T$、$\mathbb{G}_1$ の生成元を g とする。また Pairing 写像を $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ とし、双線型性を満たす。また $\mathbb{G}_T$ の生成元を $g_T = e(g, g)$ とする。ラグランジュ補間において、$\mathbb{Z}_p$ 上の元の集合 S での基底多項式を $\Delta_{i,S}(x) = \sum_{j \in S, j \neq i}(x - j)/(i - j)$ とする。

このとき、セットアップ (Setup)、暗号化 (Encrypt)、鍵生成 (Keygen)、復号 (Decrypt) の 4 つのアルゴリズムは以下の通りである。

Setup:　属性ユニバースを $\mathcal{U} = \{1, 2, ..., n\}$ とする。それぞれの属性値 $i \in \mathcal{U}$ に対して、$\mathbb{Z}_p$ 上から一様ランダムに選択した値を t_i とする。また値 $y \in \mathbb{Z}_p$ を一

様ランダムに選択する。このとき、公開パラメータ MPK およびマスタ秘密鍵 MSK は以下の通りとなる。

$$MPK := \left(T_1 := g^{t_1}, ..., T_n := g^{t_n}, Y := g_T^y\right), MSK := (t_1, ..., t_n, y)$$

Encrypt(M, γ, MPK): メッセージ空間を $\mathcal{M} := \mathbb{G}_T$ とする。また入力としてメッセージ $M \in \mathcal{M}$、指定する属性集合 $\gamma \subseteq \mathcal{U}$ とする。暗号文は、ランダムな値 $s \in \mathbb{Z}_p$ を選択し、

$$C_\gamma := (\gamma, C' := MY^s, \{C_i := T_i^s\}_{i \in \gamma})$$

となる。

Keygen$(\mathcal{T}, MSK)$: ある属性集合 γ に対して $\mathcal{T}(\gamma) = 1$ の場合に限り復号できるアクセスツリー $\mathcal{T}$ に紐づいた秘密鍵を出力する。これは $\mathcal{T}$ を表現するノード集合 $\mathcal{N}$ の全てのノードに対して、ルートノードから下位ノードに向かって順に次の計算をすることで得る。

各ノード $x \in \mathcal{N}$ において、しきい値 k_x から次数 $d_x := k_x - 1$ である多項式 q_x を定める。ルートノード r では、多項式 q_r は、$q_r(0) = y$ とし、q_r 上の他の d_r 個の値をランダムに選択する。非ルートノード x では、親ノード $\mathtt{parent}(x)$ に割り振られたインデックス $\mathtt{index}(x)$ から、多項式 q_x の定数を $q_x(0) = q_{\mathtt{parent}(x)}(\mathtt{index}(x))$ とし、また q_x 上の d_x 個の値をランダムに選択する。最後に、葉ノード x では、$\mathtt{att}(x) \in \mathcal{U}$ に対して、

$$SK_x := g^{q_x(0)/t_{\mathtt{att}(x)}}$$

を計算し、これらがアクセスツリー $\mathcal{T}$ に対する秘密鍵、

$$SK_\mathcal{T} := \{SK_x \mid \mathtt{att}(x), x \in \mathcal{N}\}$$

となる。

Decrypt$(C_\gamma, SK_\mathcal{T})$: 暗号文 $C_\gamma := (\gamma, C' := MY^s, \{C_i := T_i^s\}_{i \in \gamma})$、秘密鍵 $SK_\mathcal{T} := \{SK_x \mid \mathtt{att}(x), x \in \mathcal{N}\}$ を入力として、アクセスツリー $\mathcal{T}$ の各ノード x に対して、関数 $\mathtt{DecryptNode}(C_\gamma, SK_\mathcal{T}, x)$ を再帰的に実行することで計算する。関数 DecryptNode は、計算結果として $\mathbb{G}_T$ 上の元または $\perp$ を出力する。

ノード x が葉ノードである場合、$i = \mathtt{att}(x)$ として、以下を得る。

$$F_x := \mathtt{DecryptNode}(C_\gamma, SK_\mathcal{T}, x) = \begin{cases} e(SK_x, C_i) = g_T^{s \cdot q_x(0)} & \text{if } i \in \gamma \\ \perp & \text{if } i \notin \gamma \end{cases}$$

ノード x が非葉ノードである場合、ノード x の子ノード z_i の結果 F_{z_i} から関数 DecryptNode を計算する。まずはじめに、子ノードの内、$F_{z_i} \neq \perp$ を満たす、ある k_x 個のノードを $S_x := \{z_{i_1}, ..., z_{i_{k_x}} \mid i_j \in [1, num_x]\}$ とする。またこれらのノードのインデックスの集合 $S'_x := \{\mathtt{index}(z_i) \mid z_i \in S\}$ とする。このとき、S_x のノード数が k_x 個に満たない場合は、$\perp$ を出力する。それ以外の場合は、

$$\begin{aligned} F_x &:= \mathtt{DecryptNode}(C_\gamma, SK_{\mathcal{T}}, x) \\ &= \prod_{z \in S_x} F_z^{\Delta_{\mathtt{index}(z), S'_x(0)}} \\ &= \prod_{z \in S_x} (g_T^{s \cdot q_x(\mathtt{index}(z))})^{\Delta_{\mathtt{index}(z), S'_x(0)}} \\ &= g_T^{s \cdot q_x(0)} \end{aligned}$$

とし、多項式補間によりノード x における多項式 q_x の定数 $q_x(0)$ を得られる。
結果として、ルートノード r に対する $\mathtt{DecryptNode}(C_\gamma, SK_{\mathcal{T}}, r)$ は、再帰計算により、$F_r = g_T^{sy} = Y^s$ が得られ、最後に、メッセージ $M' := C'/F_r$ が出力される。

3.1.3 放送型暗号

放送型暗号では、送信者は複数の受信者に暗号文を同時送信し、なおかつ、そのうちの任意の受信者集合に復号権限を指定できる。この仕組みは、従来の暗号方式を単純に組み合わせても実現可能であるが、その場合は受信者数に比例した暗号文長が必要となるため、非効率的となる。これに対し、放送型暗号では受信者数よりも漸近的に短い暗号文長で同等の機能を実現することが可能である。

放送型暗号には、誰でも公開情報のみを用いた暗号化を行うことが可能な公開鍵型と暗号化にも秘密情報を必要とする共通鍵型が存在する。公開鍵型については、楕円曲線上の Pairing 写像に基づき、暗号文サイズが受信者数によらず定数となる方式 [29] が設計されて以降、効率化についての理論的な検討が継続されており、2020 年には、Agrawal と山田により、暗号文と鍵サイズが現実的な仮定の下で定数サイズとなる初めての方式が提案されている [30]。一方、共通鍵型に関しては、木構造などの組み合わせ論的技法を用いてパラメータを圧縮する方法が主流である。公開鍵型の構成と異なり、重い計算負荷を要する代数的な構造は必要

としないため、ある程度高速な処理が可能であり、各利用者に求められる鍵情報のサイズは実用的な範囲に抑えられるものの、公開鍵型のように定数サイズにまで圧縮することは困難だと思われる。

■Boneh, Gentry, Waters による放送型暗号　本稿では具体的な方式の例として Boneh-Gentry-Waters 放送型暗号方式（以下、BGW 方式）[29] を記述する。BGW 方式は、結託耐性と受信者人数に劣線形（比例しない）短い定数長の暗号文長を達成した初めての方式であり、歴史的に非常に重要である。また、提案以来効率性や安全性の改善の検討が続いてはいるものの、当該方式は現時点においても最も効率の良い方式の一つである。なお、原論文では Key encapsulation mechanism（KEM）の形で方式が記述されているが、ここでは Encryption の形で方式を記述する。平文空間は G_T となっているが、一般の平文を暗号化するためには $e(g_1, g_n)^t$ を適当な鍵導出関数に入力して得られる文字列を共通鍵暗号の秘密鍵として平文を暗号化すればよい。鍵導出関数としては、例えば SHA のような暗号的ハッシュ関数や、Pair-wise independent ハッシュ関数を用いればよい。

以下、G と G_T を素数位数 p をもつ群とし、$e : G \times G \to G_T$ を付随する（対称）Pairing 写像とする。

Setup$(1^\lambda, n)$:　セットアップアルゴリズムは、システムの人数 n とセキュリティパラメータ λ を入力とし、適切な Pairing 写像 e を選び、公開鍵を

$$PK = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, v = g^\gamma) \in G^{2n+1}$$

と設定する。ここで、g は G のランダムな生成元、$g_i = g^{\alpha^i}$、α と v は Z_p のランダムな元である。また、i 番目のユーザに対する秘密鍵を

$$d_i = g_i^\gamma$$

と設定する。最終的な出力は $(PK, d_1, \ldots, d_n)$ である。

Enc(PK, S, m):　暗号化アルゴリズムは、ユーザ集合 S にむけて平文 $m \in G_T$ を暗号化するために、PK, S, m の入力を受け、ランダムな $t \in Z_p$ を選択し、暗号文を

$$CT = (c_0 = g^t, c_1 = (v \prod_{j \in S} g_{n+1-j})^t, c_T = e(g_1, g_n)^t m) \in G^2 \times G_T$$

により計算し、出力する。

Dec(PK, S, i, d_i, CT):　平文を $m := c_T \cdot e(d_i, \cdot \prod_{j \in S, j \neq i} g_{n+1-j+i}, c_0)/e(g_i, c_1)$ により計算し、出力する。

BGW 方式は、暗号文長、秘密鍵長がシステム内のユーザ数に依存せず、定数長であるという利点があるが、公開鍵長はシステム内のユーザ数に比例して長くなるという欠点がある。[29] では、上記方式のほかに公開鍵長と暗号文長のトレードオフを実現する方式が記載されている。

■安全性について　BGW 方式は選択的攻撃に対して結託耐性を持つことが、Pairing 群上の Bilinear Diffie-Hellman Exponent（BDHE）問題と呼ばれる問題の困難性を仮定して証明可能である。結託耐性とは、システム上の復号権限を持たないユーザ全員が結託しても、暗号化された平文の情報を得ることが識別不可能性の意味でできないことを指しており、選択的攻撃とは攻撃者が公開鍵を得る前に攻撃目標である受信者集合を定める種類の攻撃を指している。BDHE 問題とは、対称 Pairing 群の元 $(g, h, g_1, \ldots, g_\ell, g_{\ell+2}, \ldots, g_{2\ell})$ が与えられた時に、$e(g_{\ell+1}, h)$ とランダムな G_T の元の識別をするという問題である。ここで、g, h はソース群の生成元で、$g_i = g^{(\alpha^i)}$ であり、α は Z_p 上でランダムにサンプルされる。また、システム内のユーザ数 n に応じてパラメータ ℓ は大きくなる。BDHE 問題に対しては、ℓ が特定の条件を満たす際に Cheon 攻撃 [31] の適用が可能であり、特に n が大きい場合には、パラメータ選定における注意が必要である。

■その他の放送型暗号方式について　放送型暗号の概念は Fiat と Naor らにより定式化された [32]。共通鍵型の放送型暗号としては [33] が著名である。BGW 方式は選択的攻撃に対する安全性しか証明できていないことが欠点の一つであったが、より強い適応的攻撃に対する安全性を持つ方式がランダムオラクルモデルで Gentry と Waters によって [34]、標準モデルで Waters によって提案された [11]。Waters による方式は、decisional linear 仮定と呼ばれる BDHE 仮定よりも標準的な数論仮定に基づき安全性証明が与えられている。標準モデルでの方式はその後の研究によってさらに効率化されている [35]。

3.1.4　準同型暗号

準同型暗号は、データを暗号化したまま演算処理（四則演算等）ができる暗号技術であり、一般には公開鍵暗号の機能要件だけで、この性質が保証されるわけではない。準同型暗号は、実現できる演算内容により、単一演算型、Leveled 型、完全型に大別される。単一演算型は、加算または乗算のみ実現可能であり、単純な演算処理のみ実現可能である。例えば、加法準同型性をもつ暗号方式として Okamoto-Uchiyama 方式 [36] や、Paillier 方式 [37] が有名である。Leveled 型は加算と乗算をともに実現可能であるが、演算回路の深さに（特に乗算の深さに）上

限がある。なお、回路の深さとは入力ゲートから出力ゲートまでに最大の長さのことである。完全型は、加算と乗算を共に実現可能であり、かつ乗算の回数に制限がない。実際には、Leveled 型にブートストラッピングと呼ばれる技術を組み合わせることにより完全型が実現されることが多い。また、Leveled 型のうち、扱える計算回路の深さが浅いものは Somewhat 型と呼ばれることも多い。以下に、具体的な準同型暗号技術のアルゴリズムを紹介する。

■Boneh, Goh, Nissim による準同型暗号　具体的な準同型暗号方式の例として Boneh-Goh-Nissim 準同型暗号方式（以下、BGN 方式）[38] を記述する。BGN 方式は 2-DNF formula に対する準同型暗号方式として提案されているが、ここでは整数上の加算と一回の乗算が可能な準同型暗号として記述する。G と G_T を合成数位数 $n = q_1 q_2$ をもつ群とし、$e : G \times G \to G_T$ を付随する（対称）Pairing とする。ここで、q_1 および q_2 は素数である。また、G は位数 q_1 の群 G_{q_1} と位数 q_2 の群 G_{q_2} の直積と同型であり、ここでは $G^{q_i} = \{g^{q_i} : g \in G\}$ を $G_{q_{3-i}}$ $(i \in \{1, 2\})$ と同一視することにする。

Setup(1^λ):　セットアップアルゴリズムでは、適切な Pairing 写像 e を選択、および、G のランダムな生成元 g と u を選択する。さらに、$h = u^{q_2}$ を計算し、公開鍵を

$$PK = (n, g, h)$$

と設定する。また、秘密鍵は $SK = (q_1, q_2)$ である。

Enc(PK, m):　暗号化アルゴリズムは、(整数として符号化された) 平文 $m \in Z$ を暗号化するために、PK, m の入力を受け、ランダムな $r \in \{0, 1, \ldots, n-1\}$ を選択し、暗号文を

$$CT = g^m h^r$$

により計算し、出力する。

Dec(PK, SK, CT):　復号アルゴリズムの入力 CT は、G の元である場合と、G_T の元である場合がある。前者は準同型的乗算を行う前の暗号文で、後者は準同型的乗算を行った後の場合である。どちらの場合も、復号アルゴリズムはまず CT^{q_1} を計算する。次に、$\hat{g}$ を底とした CT^{q_1} の離散対数を計算し、m を得る。ここで、$CT \in G$ の場合は $\hat{g} = g^{q_1}$ であり、$CT \in G$ の場合は $\hat{g} = e(g, g)^{q_1}$ である。復号が可能なためには m はセキュリティパラメータの多項式程度でなければならない。

次に、準同型演算を記述する。

Add(PK, CT_1, CT_2): 二つの暗号文に対して加法を準同型的に行うためには、CT_1 と CT_2 は同じ群 (G か G_T) の元である必要がある。両者ともに G の群である場合には、ランダムな $r \in \{0, 1, \ldots, n-1\}$ を選択し $CT_1 \cdot CT_2 \cdot h^r$ を、両者ともに G_T の群である場合には、ランダムな $r \in \{0, 1, \ldots, n-1\}$ を選択し $CT_1 \cdot CT_2 \cdot e(h, h)^r$ を計算する。

Mult(PK, CT_1, CT_2): 二つの暗号文に対して加法を準同型的に行うためには、CT_1 と CT_2 は G の元である必要がある。その場合にランダムな $r \in \{0, 1, \ldots, n-1\}$ を選択し、$e(CT_1, CT_2)e(g, h)^r$ を計算する。

■安全性について BGN 方式は、部分群判定問題の困難性を仮定すると IND-CPA 安全性をもつことが証明できる。部分群判定問題とは、G のランダムな元 g、G_{q_1} のランダムな元 h が与えられた時、G のランダムな元と G_{q_1} のランダムな元が識別不可能であるという仮定である。

■BGN 方式の後続研究について BGN 方式は、合成数位数 Pairing 群の演算が必要であるが、合成数位数 Pairing 群は素数位数 Pairing 群に比べ大きいパラメータが必要であり、計算効率が悪い。BGN 方式の改良として、素数位数 Pairing 群に基づく準同型暗号方式が Freeman によって提案され [39]、のちにさらなる改良方式が Attrapadung らにより提案された [40]。Attrapadung らは提案方式の実装も行っている。

■完全準同型暗号方式について 初の完全準同型暗号は Gentry によって 2009 年に提案された [41]。Gentry の提案は格子問題の困難性を利用していたものだったが、これが契機となり格子に基づく完全準同型暗号および Leveled 準同型暗号の研究が活発になった。代表的な方式としては、Brakerski らによる BGV 方式 [42]、Gentry らによる GSW 方式 [43]、Chillotti らによる TFHE 方式 [44]、Cheon らによる CKKS 方式 [45] などが挙げられる。CKKS 方式以外は、準同型演算の結果が復号によって正確に得られるタイプの完全準同型暗号方式だが、CKKS 方式は復号結果にエラーが加わるタイプの方式であり、復号結果にエラーが加わることを容認することで効率化を図っている。CKKS を含む復号にエラーが含まれるタイプの完全準同型暗号に関しては、セキュリティ上の懸念が指摘されており、利用シナリオやパラメータ選定等に注意が必要な場合がある [46]。

完全準同型暗号の著名な実装例としては、Homomorphic Encryption library（HElib）[47] や Simple Encrypted Arithmetic Library（SEAL）[48] が挙げられる。HElib は IBM 社がオープンソースで提供している準同型暗号のライブラリであり、BGV 方式と CKKS 方式が利用可能である。また、SEAL は Microsoft 社

がオープンソースで提供している準同型暗号のライブラリであり、Fan らによる FV 方式 [49] と CKKS 方式が利用可能である。

3.1.5 プロキシ再暗号化

プロキシ再暗号化は、あるユーザ宛の暗号文を、復号することなく別のユーザ宛の暗号文へ変換することが可能な暗号技術である。より具体的に、ユーザ A は、もう一方のユーザ B とプロキシ（代理人）と呼ばれる第三者を指定し、プロキシに対して「再暗号化鍵 $\mathsf{rk}_{\mathrm{A}\to\mathrm{B}}$」を預託する。プロキシは、この再暗号化鍵 $\mathsf{rk}_{\mathrm{A}\to\mathrm{B}}$ を使って、ユーザ A 宛の暗号文 ct_{A} の中身を知ることなく、ユーザ B 宛の暗号文 ct_{B} に変換することができる。

プロキシ再暗号化方式は、再暗号化鍵 $\mathsf{rk}_{\mathrm{A}\to\mathrm{B}}$ の性質や暗号文が備える安全性レベルによっていくつかに分類可能である。以下、代表的な分類のいくつかを紹介する。

- 一方向/双方向変換: ユーザ A から B への再暗号化鍵 $\mathsf{rk}_{\mathrm{A}\to\mathrm{B}}$ を用いて、ユーザ B 宛の暗号文をユーザ A 宛の暗号文に変換することができない場合は一方向、できる場合は双方向。
- CPA/CCA 安全性: 選択的平文攻撃に対してのみ安全であれば CPA 安全で、選択的暗号文攻撃に対しも安全であれば CPA 安全。
- シングル/マルチホップ: 再暗号化した暗号文をそれ以上再暗号化できなければシングルホップ、連続して再暗号化できればマルチホップ。

これらの性質については、用途に応じて選定を行う必要があるが、求められる性質によって処理性能に差が生じるため、慎重な判断が必要となる。なお、双方向変換可能で CPA 安全なマルチホップ方式が、最も高速な処理が可能となる。以下、Blaze ら [50] により初めて提案された双方向変換可能で CPA 安全なマルチホップな方式である BBS 方式を紹介する。BBS 方式は、ElGamal 暗号の簡単な変形で導かれ、その後のプロキシ再暗号化の構成の基盤となっている。

■Blaze, Bleumer, Strauss **によるプロキシ再暗号化方式** [50]　以下、$\mathbb{G}$ を素数位数 p を持つ群とし、$g \in \mathbb{G}$ をその生成元とする。

KeyGen:　各ユーザは、鍵生成アルゴリズムを実行し、公開鍵 $\mathsf{pk} = g^x \in \mathbb{G}$ と秘密鍵 $\mathsf{sk} = x \in \mathbb{Z}_p \backslash \{0\}$ を生成する。

ReKeyGen$(\mathsf{sk}_{\mathrm{A}}, \mathsf{sk}_{\mathrm{B}})$:　ユーザ A と B は、各々の秘密鍵 $\mathsf{sk}_{\mathrm{A}} = a$ と $\mathsf{sk}_{\mathrm{B}} = b$ を入力に再暗号化鍵生成アルゴリズムを実行し、再暗号化鍵 $\mathsf{rk}_{\mathrm{A}\to\mathrm{B}} = b/a \in \mathbb{Z}_p$ を

プロキシに渡す。

$\mathsf{Enc}(\mathsf{pk}_\mathrm{A}, \mathsf{msg} \in \mathbb{G})$: 暗号化アルゴリズムを利用し、ユーザ A 宛の暗号文 $\mathsf{ct}_\mathrm{A} = (\mathsf{msg} \cdot g^r, (\mathsf{pk}_\mathrm{A})^r)$ を生成する。なお、r はアルゴリズムがサンプルする乱数である。ここで、通常の ElGamal 暗号は $\mathsf{ct}_\mathrm{A} = (g^r, \mathsf{msg} \cdot (\mathsf{pk}_\mathrm{A})^r)$ であることに注意されたい。

$\mathsf{ReEnc}(\mathsf{rk}_{\mathrm{A}\to\mathrm{B}}, \mathsf{ct}_\mathrm{A})$: プロキシは、再暗号化アルゴリズムを実行し、ユーザ A 宛の暗号文 $\mathsf{ct}_\mathrm{A} = (\mathsf{ct}_1, \mathsf{ct}_2)$ から、ユーザ B 宛の暗号文 $\mathsf{ct}_\mathrm{B} = (\mathsf{ct}_1, (\mathsf{ct}_2)^{\mathsf{rk}_{\mathrm{A}\to\mathrm{B}}})$ を生成する。

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: 各ユーザは、復号アルゴリズムを実行し、まず自身宛の暗号文 $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$ から $\mathsf{msg}' = (\mathsf{ct}_1)^{\mathsf{sk}}/\mathsf{ct}_2$ を計算し、平文 $\mathsf{msg} = (\mathsf{msg}')^{1/\mathsf{sk}}$ を出力する。

暗号文 ct_A がユーザ A 宛であれば、ct_A は $(\mathsf{msg}\cdot g^r, g^{ar})$ であるため、秘密鍵 $\mathsf{sk}_\mathrm{A} = a$ を用いて、msg が正しく復号できることが確認できる。また、$\mathsf{rk}_{\mathrm{A}\to\mathrm{B}} = b/a$ を利用することで、暗号文 ct_A の第 2 項目が、g^{br} に変換できるため、プロキシがユーザ B 宛の暗号文 ct_B に正しく変換できていることも確認できる。$\mathsf{rk}_{\mathrm{B}\to\mathrm{A}} = (\mathsf{rk}_{\mathrm{A}\to\mathrm{B}})^{-1}$ であることに注意すると、$\mathsf{rk}_{\mathrm{A}\to\mathrm{B}} = b/a$ が双方向変換に対応していることが確認できる。最後に、異なるユーザ C に対する再暗号化鍵 $\mathsf{rk}_{\mathrm{B}\to\mathrm{C}}$ を渡されたプロキシは、ct_B をさらにユーザ C 宛の暗号文 ct_C に変換できるため、マルチホップ性も満たす。

■BBS 方式の安全性について BBS 方式は、通常の公開鍵暗号が備える最も基本的な選択平文攻撃に対する安全性（CPA 安全性）に類似した CPA 安全性を備える。通常の CPA 安全性では、攻撃者は公開鍵 pk_A、および、自身が選んだ任意の平文 msg_0 と msg_1 のいずれかに関する暗号文 ct^*_A を入手し、ct^*_A がどちらの平文を暗号化しているか識別できないことを要求する。プロキシ再暗号化方式においては、これに加え、任意の $n \in \mathbb{N}$ に対して、攻撃者は n 人の異なるユーザの公開鍵 $\{\mathsf{pk}_{\mathrm{B}_i}\}_{i\in[n]}$ と再暗号化鍵 $\{\mathsf{rk}_{\mathrm{A}\to\mathrm{B}_i}\}_{i\in[n]}$ を与えられた上での、識別不能性を要求する。具体的に BBS 方式では、$(g, g^a, \{g^{b_i}, b_i/a\}_{i\in[n]}, g^r, \underline{g^{ar}})$ の組みと、$(g, g^a, \{g^{b_i}, b_i/a\}_{i\in[n]}, g^r, \underline{g^{r'}})$ の組みが識別不可能であることを意味する。ここで、後者の下線部に現れる r' は他の変数に依存しないランダムな $\mathbb{Z}_p$ 上の要素である。この組みを識別する問題は、Decisional Diffie-Hellman（DDH）問題の困難性に帰着でき、BBS 方式は DDH 問題の困難性仮定に基づいて CPA 安全であることが示される。

■その他のプロキシ再暗号化方式　BBS 方式は、もし攻撃者が（$\mathsf{ct}_{\mathrm{A}}^{*}$ 以外の）暗号文を復号できる復号オラクルと対話できる場合、平文 msg_0 か msg_1 のどちらが暗号化されているかを自明に識別することができる。従って、BBS 方式は CPA 安全であるが、より強い選択暗号文攻撃に対する安全性（CCA 安全性）を持たない。Canetti と Hohenberger [51] は、CCA 安全性を備える、双方向変換可能でマルチホップな再暗号化方式を提案した。BBS 方式と比べ、構成に双線型写像を要するため、安全性と引き換えに効率性が少し劣る。

双方向変換可能なプロキシ再暗号化方式は、再暗号化鍵 $\mathsf{rk}_{\mathrm{A}\to\mathrm{B}}$ を生成する際にユーザ A と B の秘密鍵 sk_{A} と sk_{B} を必要とする。しかし、一方向変換可能なプロキシ再暗号化方式は、再暗号化鍵 $\mathsf{rk}_{\mathrm{A}\to\mathrm{B}}$ を生成する際に、ユーザ A の秘密鍵 sk_{A} とユーザ B の公開鍵 pk_{B} だけが必要になり、$\mathsf{rk}_{\mathrm{A}\to\mathrm{B}}$ からユーザ B の秘密鍵 sk_{B} の情報が漏洩することはない。従って、ユーザ B の暗号文をユーザ A の暗号文に変更する必要がないアプリケーションにおいては、一方向変換可能なプロキシ再暗号化方式が双方向変換可能な方式と比べ、安全性を担保できるため適切である。そして、Libert と Vergnaud [52] は、CCA 安全で一方向変換可能なプロキシ再暗号化方式を双線型写像から構成し、花岡ら [53] は、特定の性質を満たすしきい値暗号に基づく一般的な構成を示した。いずれの方式もシングルホップであり、CCA 安全で一方向変換可能、かつ、マルチホップな効率的なプロキシ再暗号化方式は現段階では開発されていない。

参考文献

[1] A. Shamir: “Identity-Based Cryptosystems and Signature Schemes”, CRYPTO 1984, Springer-Verlag, pp. 47–53 (1984).

[2] Y. Dodis, J. Katz, S. Xu and M. Yung: “Key-insulated public key cryptosystems”, EUROCRYPT 2002, Springer-Verlag, pp. 65–82 (2002).

[3] R. Canetti, S. Halevi and J. Katz: “A forward-secure public-key encryption scheme”, EUROCRYPT 2003, Springer-Verlag, pp. 255–271 (2003).

[4] K. Paterson and E. Quaglia: “Time-specific encryption”, SCN 2010, Springer-Verlag, pp. 1–16 (2010).

[5] D. Boneh, R. Canetti, S. Halevi and J. Katz: “Chosen ciphertext security from identity based encryption”, SIAM J. Computing, **36**, 5, pp. 1301–1328 (2007).

[6] R. Sakai, K. Ohgishi and M. Kasahara: “Cryptosystems Based on Pairings”,

SCIS 2000 (2000).

[7] D. Boneh and M. Franklin: "Identity-Based Encryption from the Weil Pairing", CRYPTO 2001, Springer-Verlag, pp. 213–229 (2001).

[8] D. Boneh and X. Boyen: "Efficient Selective-ID Secure Identity Based Encryption without Random Oracles", EUROCRYPT 2004, Springer-Verlag, pp. 223–238 (2004).

[9] B. Waters: "Efficient Identity Based Encryption without Random Oracles", EUROCRYPT 2005, Springer-Verlag, pp. 114–127 (2005).

[10] C. Gentry: "Practical identity-based encryption without random oracles", EUROCRYPT 2006, Springer-Verlag, pp. 445–464 (2006).

[11] B. Waters: "Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions", CRYPTO 2009, Springer-Verlag, pp. 619–636 (2009).

[12] C. Gentry, C. Peikert and V. Vaikuntanathan: "Trapdoors for Hard Lattices and New Cryptographic Constructions", STOC 2008, ACM, pp. 197–206 (2008).

[13] S. Agrawal, D. Boneh and X. Boyen: "Efficient lattice (H)IBE in the standard model", EUROCRYPT 2010, Springer-Verlag, pp. 553–572 (2010).

[14] N. Döttling and S. Garg: "Identity-based encryption from the Diffie-Hellman assumption", CRYPTO 2017, Springer-Verlag, pp. 537–569 (2017).

[15] D. Boneh, X. Boyen and E.-J. Goh: "Hierarchical Identity Based Encryption with Constant Size Ciphertext", EUROCRYPTO 2005, Springer-Verlag, pp. 440–456 (2005).

[16] S. S. Chow, Y. Dodis, Y. Rouselakis and B. Waters: "Practical leakage-resilient identity-based encryption from simple assumptions", ACM CCS 2010, pp. 152–161 (2010).

[17] A. Boldyreva, V. Goyal and V. Kumar: "Identity-based encryption with efficient revocation", ACM CCS 2008, pp. 417–426 (2008).

[18] A. Sahai and B. Waters: "Fuzzy Identity-Based Encryption", EUROCRYPT 2005, Springer-Verlag, pp. 457–473 (2005).

[19] V. Goyal, O. Pandey, A. Sahai and B. Waters: "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data", ACM CCS 2006, pp. 89–98 (2006).

[20] J. Bethencourt, A. Sahai and B. Waters: "Ciphertext-Policy Attribute-

Based Encryption", IEEE S&P 2007 (2007).

[21] A. Lewko, T. Okamoto, A. Sahai, K. Takashima and B. Waters: "Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption", EUROCRYPT 2010, Springer-Verlag, pp. 62–91 (2010).

[22] B. Waters: "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization", PKC 2011, Springer-Verlag, pp. 53–70 (2011).

[23] S. Gorbunov, V. Vaikuntanathan and H. Wee: "Attribute-Based Encryption for Circuits", STOC 2013, ACM, pp. 545–554 (2013).

[24] D. Boneh and B. Waters: "Conjunctive, Subset, and Range Queries on Encrypted Data", TCC 2007, Springer-Verlag, pp. 535–554 (2007).

[25] R. Ostrovsky, A. Sahai and B.Waters: "Attribute-Based Encryption with Non-Monotonic Access Structures", ACM CCS 2007, pp. 195–203 (2007).

[26] G. Wang, Q. Liu, J. Wu and M. Guo: "Hierarchical Attribute-Based Encryption and Scalable User Revocation for Sharing Data in Cloud Servers", Computers and Security, Vol. 30, Issue 5, ACM, pp. 320–331 (2011).

[27] M. Chase: "Multi-Authority Attribute Based Encryption", TCC 2007, Springer-Verlag, pp. 515–534 (2007).

[28] A. Lewko and B. Waters: "Decentralizing Attribute-Based Encryption", EUROCRYPT 2011, Springer-Verlag, pp. 568–588 (2011).

[29] D. Boneh, C. Gentry and B. Waters: "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys", CRYPTO 2005, Springer-Verlag, pp. 258–275 (2005).

[30] S. Agrawal and S. Yamada: "Optimal Broadcast Encryption from Pairings and LWE", EUROCRYPT 2020, Springer-Verlag, pp. 13–43 (2020).

[31] J. Cheon: "Security Analysis of the Strong Diffie-Hellman Problem", EUROCRYPT 2006, Springer-Verlag, pp. 1–11 (2006).

[32] A. Fiat and M. Naor: "Broadcast Encryption", CRYPTO 1993, Springer-Verlag, pp. 480–491 (1993).

[33] D. Naor, M. Naor and J. Lotspiech: "Revocation and Tracing Schemes for Stateless Receivers", CRYPTO 2001, Springer-Verlag, pp. 41–62 (2001).

[34] C. Gentry and B. Waters: "Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts)", EUROCRYPT 2009, Springer-Verlag,

pp. 171–188 (2009).

[35] J. Chen, R. Gay and H. Wee: "Improved Dual System ABE in Prime-Order Groups via Predicate Encodings", EUROCRYPT 2015, Springer-Verlag, pp. 595–624 (2015).

[36] T. Okamoto and S. Uchiyama: "A New Public-Key Cryptosystem as Secure as Factoring", EUROCRYPT 1998, Springer-Verlag, pp. 308–318 (1998).

[37] P. Paillier: "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes", EUROCRYPT 1999, Springer-Verlag, pp. 223–238 (1999).

[38] D. Boneh, E. Goh and K. Nissim: "Evaluating 2-DNF Formulas on Ciphertexts", TCC 2005, Springer-Verlag, pp. 325–341 (2005).

[39] D. Freeman: "Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups", EUROCRYPT 2010, Springer-Verlag, pp. 44–61 (2010).

[40] N. Attrapadung, G. Hanaoka, S. Mitsunari, Y. Sakai, K. Shimizu and T. Teruya: "Efficient Two-level Homomorphic Encryption in Prime-order Bilinear Groups and A Fast Implementation in WebAssembly", AsiaCCS 2018, ACM, pp. 685–697 (2018).

[41] C. Gentry: "Fully Homomorphic Encryption Using Ideal Lattices", STOC 2009, ACM, pp. 169–178 (2009).

[42] Z. Brakerski, C. Gentry and V. Vikuntantathan: "(Leveled) Fully Homomorphic Encryption without Bootstrapping", ITCS 2012, ACM, pp. 309–325 (2012).

[43] C. Gentry, A. Sahai and B. Waters: "Homomorphic Encryption from Learning with Errors: Coceptually-Simpler, Asymptotically-Faster, Attribute-Based", CRYPTO 2013, Springer-Verlag, pp. 75–92 (2013).

[44] I. Chillotti, N. Gama, M. Georgieva and M. Izabachène: "Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds", ASIACRYPT 2016, Springer-Verlag, pp. 3–33 (2016).

[45] J. Cheon, A. Kim, M. Kim and Y. Song: "Homomorphic Encryption for Arithmetic of Approximate Numbers", ASIACRYPT 2017, Springer-Verlag, pp. 409–437 (2017).

[46] B. Li and D. Micciancio: "On the Security of Homomorphic Encryption on Approximate Numbers", EUROCRYPT 2021, Springer-Verlag, pp. 648–677 (2021).

[47] "HElib (Homomorphic Encryption library)". https://github.com/homenc/HElib.

[48] "Simple Encrypted Arithmetic Library". https://www.microsoft.com/en-us/research/project/microsoft-seal/.

[49] J. Fan and F. Vercauteren: "Somewhat Practical Fully Homomorphic Encryption". ePrint 2012/144.

[50] M. Blaze, G. Bleumer and M. Strauss: "Divertible Protocols and Atomic Proxy Cryptography", EUROCRYPT 1998, Springer-Verlag, pp. 127–144 (1998).

[51] R. Canetti and S. Hohenberger: "Chosen-Ciphertext Secure Proxy Re-Encryption", ACM CCS 2007, pp. 185–194 (2007).

[52] B. Libert and D. Vergnaud: "Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption", PKC 2008, Springer-Verlag, pp. 360–379 (2008).

[53] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang and Y. Zhao: "Generic Construction of Chosen Ciphertext Secure Proxy Re-Encryption", CT-RSA 2012, Springer-Verlag, pp. 349–364 (2012).

3.2 認証・署名を目的とした高機能暗号技術

3.2.1 属性ベース署名

属性ベース署名は、署名者が持つ署名鍵が、署名を付与する平文に付随する条件を満たしている時に限り署名を生成でき、かつ、署名からは条件が満たされたという情報以外は漏れない署名方式である。より具体的に、属性ベース署名は、平文ポリシー型と鍵ポリシー型に大別されるが、前者は平文に条件、いわゆるアクセス構造が付随し、署名生成鍵に属性が付随する。平文と署名鍵に付随する情報を逆転させると後者になる。この際、属性（例えば、 会社員、30 歳以上、東京都在住）がアクセス構造（例えば、18 歳以上の都民であるか？）を満たす時に限り、署名を生成することができる。例えば平文ポリシー型であれば、アクセス構造を満たす属性を所持する署名者が署名したという情報以外が漏れないため、匿名性を担保しながらアクセス制御が可能になる。どちらのポリシー型が適切かはアプリケーションによるが、一般的に平文ポリシー型の方が利便性が高い属性ベース署名と考えられており、より多くの構成方法が知られている。

Maji ら [1] により初めて属性ベース署名が提案されて以降、扱えるアクセス構造をより豊かにする研究が進み、今では論理回路やチューリング機械によって表現できる非常に幅広いアクセス構造を扱える属性ベース署名が知られている [2, 3, 4]。いずれの方式も構成方法の概略は似ている。以下典型的な、論理回路をアクセス構造に扱える平文ポリシー型の属性ベース署名の実現方法の概略を説明する。記載する方式は、初めて論理回路をアクセス構造に扱える方式を楕円曲線上で達成した坂井ら [2] の方式を、電子署名・コミットメント方式・非対話型ゼロ知識証明と呼ばれる暗号要素技術で一般化した El Kaafarani と勝又 [4] の方式に基づいている。

■論理回路をアクセス構造に扱える平文ポリシー型の属性ベース署名

Setup: 管理者は、Setup アルゴリズムを実行し、システム共有のパラメータを生成する。具体的に、電子署名の公開鍵 vk と秘密鍵 sk、コミットメント方式の公開鍵 $\mathsf{pk_c}$ を生成し、属性ベース署名の管理者公開鍵 $\mathsf{mpk} = (\mathsf{vk}, \mathsf{pk_c})$ を公開し、管理者秘密鍵 $\mathsf{msk} = \mathsf{sk}$ を保管する。

KeyGen(msk, x): 管理者は、KeyGen アルゴリズムを実行し、（L ビットにエンコードされた）属性 $\mathsf{x} \in \{0,1\}^L$ を所持するユーザの署名鍵 K_{x} を生成する。具体的に、管理者は sk を用いて、x に対して署名 σ を生成し、署名鍵 $K_{\mathsf{x}} = \sigma$ を

出力する。

Sign($\mathsf{mpk}, K_{\mathsf{x}}, C, \mathsf{msg}$): 属性 x を所持するユーザは、Sign アルゴリズムを実行し、$C(\mathsf{x}) = 1$ を満たす論理回路 $C : \{0,1\}^L \to \{0,1\}$ で表されるアクセス構造が付随する平文 msg に対して署名を生成する。具体的に、以下の手順で署名を生成する。

1. 属性 $\mathsf{x} \in \{0,1\}$ と署名鍵 $K_{\mathsf{x}} = \sigma$ を管理者公開鍵に含まれる $\mathsf{pk_c}$ でコミットし、それらを各々 $\mathsf{com}_{\mathsf{x}}$ と com_{σ} と置く。
2. 属性 $\mathsf{x} \in \{0,1\}$ と署名鍵 $K_{\mathsf{x}} = \sigma$ を秘匿にしながら、次のような関係を示す非対話型ゼロ知識証明 π を生成し、$\Sigma = (\mathsf{com}_{\mathsf{x}}, \mathsf{com}_{\sigma}, \pi)$ を署名として出力する。本説明では、平文 msg が署名 Σ に紐づいていないが、一般的には、論理回路 C か非対話型ゼロ知識証明の内部に msg を埋め込むことで紐づけられることに注意されたい。
   (a) $C(\mathsf{x}) = 1$ である。つまり、属性がアクセス構造を満たす。
   (b) σ が管理者公開鍵に含まれる vk の下で正当な署名である。
   (c) $\mathsf{com}_{\mathsf{x}}$ と com_{σ} が管理者公開鍵に含まれる $\mathsf{pk_c}$ の下で x と σ に対するコミットメントである。

Verify($\mathsf{mpk}, \Sigma, C, \mathsf{msg}$): 検証者は、Verify アルゴリズムを実行し、論理式 C によって表されるアクセス構造を満たす属性を所持したユーザが msg に対して署名 Σ を生成したかを確認する。具体的に、非対話型ゼロ知識証明の証明 Σ の第 3 項 π が正しいことを検証する。

■安全性について 属性ベース署名は、適応的文書攻撃に対して存在的偽造不可能性(EUF-CMA 安全性)、および、秘匿性を満たす必要がある。前者は、一般的な電子署名における EUF-CMA 安全性に似ており、攻撃者がある論理回路 C^* で表されるアクセス構造が付随した(以下、C^* が付随した)平文 msg に対する署名 Σ を偽造できないことを要求する。この際、攻撃者は、$C^*(\mathsf{x}) = 0$ を満たす任意の属性 x(つまり、アクセス構造を満たさない x)に対する署名鍵 K_{x} を管理者から入手でき、さらに、異なる論理回路 $C \neq C^*$ が付随した平文 msg に対する署名を得られる。後者は、属性ベース署名に特有な安全性要件で、署名から属性情報が漏れないことを要求する。具体的には、異なる属性 x^* と $\mathsf{x}^{*\prime}$ が論理回路 C^* を満たす場合、C^* が付随した平文 msg に対して属性 x と $\mathsf{x}^{*\prime}$ を所持するいずれのユーザが署名を打ったかが識別できないことを要求する。これは、署名からはアクセス構造を満たした誰かが署名を作成したことしか漏れず、具体的な属性が秘匿されることを表す。

上記で紹介した方式の EUF-CMA 安全性は、概略的には電子署名の EUF-CMA 安全性と非対話型ゼロ知識証明の健全性から導かれる。仮に攻撃者に属性ベース署名の偽造ができたとした場合、非対話型ゼロ知識証明の健全性より、管理者公開鍵に含まれる電子署名の公開鍵 vk の下で正当な署名 σ を利用していなければならない。攻撃者は、管理者から $C^*(\mathsf{x}) = 0$ を満たす任意の属性 x に対する署名鍵 $K_{\mathsf{x}} = \sigma'$ を入手できるが、これらは C^* が付随した平文に署名することはできない。従って、攻撃者が偽造に利用した σ は、管理者から入手した σ' とは異なるが、これは電子署名の EUF-CMA 安全性に矛盾する。以上より、属性ベース署名の偽造を生成できる攻撃者がいないことが示される。

また、秘匿性については、コミットメントの秘匿性と非対話型ゼロ知識証明のゼロ知識性から直ちに導かれる。署名 Σ は、$(\mathsf{com}_{\mathsf{x}}, \mathsf{com}_{\sigma}, \pi)$ という形をしているが、コミットメントの秘匿性より $\mathsf{com}_{\mathsf{x}}$ と com_{σ} は、属性 x と署名鍵 K_{x} の情報を隠す。さらに、非対話型ゼロ知識証明のゼロ知識性より、証明 π も正しい証拠を利用して証明が作られた以上の情報を漏らさない。以上より、Σ は、論理回路 C^* を満たす属性で生成された署名である以上の情報を漏らしていない。

■具体的な実現方法について　現在知られている属性ベース署名のほとんどは上記構成に従っており、違いはどのような電子署名、コミットメント、非対話型ゼロ知識証明を用いるかである。例えば、坂井ら [2] は上記一般構成の 2.(a) と 2.(b), (c) に各々特化した非対話型ゼロ知識証明 Groth-Sahai [5] と Groth-Ostrovsky-Sahai [6] を融合させることにより、Pairing 写像に基づく効率的な属性ベース署名を構成した。一方で、El Kaafarani と勝又 [4] は上記の暗号要素技術を耐量子計算機安全な道具で実現することで、初めて耐量子計算機安全な論理回路をアクセス構造に扱える属性ベース署名を構成した。

3.2.2　集約 MAC、マルチ MAC、集約署名、マルチ署名

メッセージ認証コード（Message Authentication Code，MAC）や電子署名は、ある平文を特定のユーザが認証したこと、署名したことを暗号学的に保証できる暗号技術である。MAC は秘密鍵を利用してタグを検証することができ、電子署名は公開情報のみで署名を検証できる。通常の MAC や電子署名では、複数のユーザや平文に対してタグを付与したり、署名を生成する場合、MAC や署名を行う回数だけタグや署名が生成されるため、タグや署名が数多く生成される。アプリケーションにおいてはスケーラビリティの課題となる。そこで、この課題を解決するための技術である集約 MAC、マルチ MAC、集約署名、マルチ署名を紹介する。

- **集約 MAC** [7]: 異なる平文に対して、複数の異なる秘密鍵で生成されたタグを、単一のタグに集約する技術であり、タグサイズの削減が可能となる。タグ検証においても、集約したタグの検証のみで複数の平文の正当性を一括で検証可能となる。ただし、平文や秘密鍵は必ずしも異なるとは限らない。
- **マルチ MAC** [7]: 単一の平文に対して、複数の異なる秘密鍵で生成されたタグを、単一のタグに集約する技術である。タグ生成の際の平文が全ユーザで同一な、集約 MAC の特殊ケースとして捉えることができ、集約 MAC と比べてより効率的な構成が可能となる。
- **集約署名** [8, 9]: 電子署名の枠組みで集約 MAC と同じ機能を実現する方法である。
- **マルチ署名** [8, 9, 10, 11, 12]: 電子署名の枠組みでマルチ MAC と同じ機能を実現する方法である。

集約・マルチ MAC/署名の中には、集約する際に MAC/署名を生成するユーザ同士が対話する必要がある方式や、集約が逐次的にしか行えない方式等様々な方式があるため、用途に応じて選定することが求められる。一般的に、利便性が高い方式ほど効率性が悪くなる傾向があることに注意されたい。例えば、常にネットに繋がっているサーバ同士でタグや署名を集約する場合は、効率を高めるために対話型の集約方式の利用が好ましい。一方で、一般のユーザがクラウドやブロックチェーン上にアップロードしたタグや署名を集約する応用では、全ユーザが同時にオンラインであるとは限らないため、非対話型の集約方式の方が好ましい。

以下、代表的な方式を二つを紹介する。

■Katz と Lindell による集約・マルチ MAC [7]　F を鍵空間と出力空間が $\{0,1\}^\lambda$ で入力空間が $\mathbb{M}$ の擬似ランダム関数とする。以下、平文空間に $\mathbb{M}$ を持つ集約 MAC を F から構成する。

$\mathsf{KeyGen}(1^\lambda)$:　各ユーザは、鍵生成アルゴリズムを実行し、$\mathsf{F}$ の鍵 $\mathsf{K} \leftarrow \{0,1\}^\lambda$ をサンプルし、秘密鍵 $\mathsf{key} = \mathsf{K}$ を出力する。

$\mathsf{MAC}(\mathsf{key}, \mathsf{msg} \in \mathbb{M})$:　ユーザは、タグ生成アルゴリズムを実行し、タグ $\mathsf{tag} \leftarrow \mathsf{F}(\mathsf{K}, \mathsf{msg})$ を生成する。

$\mathsf{Aggregate}((\mathsf{msg}_i, \mathsf{tag}_i)_{i\in[N]})$:　タグ集約者は、各ユーザが生成した平文 msg_i に対するタグ tag_i を入力にタグ集約アルゴリズムを実行し、集約タグ $\mathsf{tag} = \bigoplus_{i\in[N]} \mathsf{tag}_i \in \{0,1\}^\lambda$ を生成する。

$\mathsf{Verify}((\mathsf{key}_i, \mathsf{msg}_i)_{i\in[N]}, \mathsf{tag})$:　タグ検証者は、各ユーザの秘密鍵 $\mathsf{key}_i = \mathsf{K}_i$ と署名したであろう平文 msg_i を入力にタグ検証アルゴリズムを実行し、$\mathsf{tag} =$

$\bigoplus_{i\in[N]} \mathsf{F}(\mathsf{K}_i, \mathsf{msg}_i)$ が成り立てば、集約タグが正当だと判定する。ただし、異なる $i, j \in [N]$ に対して $(\mathsf{key}_i, \mathsf{msg}_i) \neq (\mathsf{key}_j, \mathsf{msg}_j)$ とする。

タグ検証アルゴリズムはタグ生成を再度行い、それが集約タグと同一かを確認する。タグ生成が決定的であることより、方式の正当性は確認できる。参考までに、上記方式は擬似ランダム関数の代わりに、タグ生成が決定的な任意の MAC を利用することができる。詳細は [7] を参考にされたい。また、上記方式はタグ集約アルゴリズムが同じ平文だけを入力に取るように制限することで、マルチ MAC にもなる。

■集約・マルチ MAC の安全性について　通常の MAC と同様に、適応的平文と署名攻撃に対して存在的偽造不可能性（EUF-CMVA 安全性）と同等の安全性を要求する。この安全性は、秘密鍵 $\mathsf{key}_i = \mathsf{K}_i$ を持つユーザが平文 msg_i を署名していない限り、$\mathsf{Verify}((\mathsf{key}_i, \mathsf{msg}_i)_{i\in[N]}, \mathsf{tag})$ が正当と判断するような組み $(\mathsf{key}_i, \mathsf{msg}_i)_{i\neq j, i\in[N]}$ と集約タグ τ を攻撃者が出力できないことを意味する。ただし、攻撃者は任意のタグと平文の組み (σ, msg) が秘密鍵 key_i の元で正しい署名か検証してもらえるとする。EUF-CMVA 安全性を上記方式が満たすことは、F が擬似ランダム関数であることより直ちに導かれる。

■Boneh, Drijvers と Neven による集約・マルチ署名 [9]　$\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T$ を素数位数 p を持つ群として、g_0 を $\mathbb{G}_0$ の生成元、$e : \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_T$ を付随する（非対称）Pairing 写像とする。また、H_0 と H_1 をハッシュ関数とする。以下、平文空間に $\mathbb{M}$ を持つ集約署名を構成する。

KeyGen :　各ユーザは、鍵生成アルゴリズムを実行し、$a \leftarrow \mathbb{Z}_p$ をサンプルし、公開鍵 $\mathsf{pk} = h = g_0^a \in \mathbb{G}_0$ と秘密鍵 $\mathsf{sk} = a$ を生成する。

$\mathsf{Sign}(\mathsf{sk}, \mathsf{msg})$:　ユーザは、署名生成アルゴリズムを実行し、署名 $\sigma = H_0(\mathsf{msg})^{\mathsf{sk}} \in \mathbb{G}_1$ を生成する。

$\mathsf{Aggregate}((\mathsf{pk}_i, \mathsf{msg}_i, \sigma_i)_{i\in[N]})$:　署名集約者は、各ユーザが生成した平文 msg_i に対する署名 σ_i を入力に署名集約アルゴリズムを実行し、まず $(t_1, \cdots, t_N) \leftarrow H_1(\mathsf{pk}_1, \cdots, \mathsf{pk}_N) \in \mathbb{Z}_p^N$ を計算し、集約署名 $\sigma = \prod_{i\in[N]} \sigma_i^{t_i}$ を生成する。

$\mathsf{Verify}((\mathsf{pk}_i, \mathsf{msg}_i)_{i\in[N]}, \sigma)$:　署名検証者は、各ユーザの公開鍵 $\mathsf{pk}_i = g^{a_i}$ と署名したであろう平文 msg_i を入力に署名検証アルゴリズムを実行し、まず $(t_1, \cdots, t_N) \leftarrow H_1(\mathsf{pk}_1, \cdots, \mathsf{pk}_N) \in \mathbb{Z}_p^N$ を計算し、$e(g, \sigma) = \prod_{i\in[N]} e(\mathsf{pk}_i^{t_i}, H_0(\mathsf{msg}_i))$ が成り立てば、集約タグが正当なものだと判定する。

アルゴリズムの正当性は、下記の数式が成立するこから確認できる。

$$e(g_0, \sigma) = e(g_0, \prod_{i\in[N]} \sigma_i^{t_i}) = \prod_{i\in[N]} e(g_0, H_0(\mathsf{msg}_i)^{\mathsf{sk}_i \cdot t_i}) = \prod_{i\in[N]} e(\mathsf{pk}_i^{t_i}, H_0(\mathsf{msg}_i))$$

さらに、各ユーザが同じ平文 msg を署名するマルチ署名の設定においては、予め集約された公開鍵 $\mathsf{apk} = \prod_{i\in[N]} \mathsf{pk}_i \in \mathbb{G}_0$ を計算することで、集約署名の検証は $e(g, \sigma) = e(\mathsf{apk}, H_0(\mathsf{msg}))$ を確かめるだけである。計算コストが比較的に高い Pairing 写像の計算が $(n+1)$ 回から 2 回にでき、マルチ署名の方が集約署名よりも検証コストが少なくなる。集約・マルチ MAC の場合、このような検証コストの差はない。

■集約・マルチ署名の安全性について　集約・マルチ MAC の場合と同様、通常の電子署名における適応的平文攻撃に対して存在的偽造不可能性（EUF-CMA 安全性）と同等の安全性を要求する。ただし、MAC とは違い、電子署名では検証を攻撃者自身が行えるため、選択署名攻撃を考慮に入れなくてよいことに注意されたい。上述の方式は、Boneh ら [13] の電子署名の自然な集約・マルチ署名への拡張になっているが、EUF-CMA 安全性も大まかには同様に導かれる。具体的には、$(g_0^a, g_1^a, g^b) \in \mathbb{G}_0 \times \mathbb{G}_1 \times \mathbb{G}_1$ を与えられても $g_0^{ab} \in \mathbb{G}_0$ を求めることができないという、co-Computational Diffie-Hellman 仮定の困難性より導かれる。

■その他の方式　署名の設定において、現在最も効率的な集約署名は上記で紹介した Boneh ら [9] の方式である。マルチ署名については、署名集約アルゴリズムで各ユーザが対話することを許せば、Pairing 写像を用いないより効率的な方式がある。Bellare と Neven[10] は、対話を 3 ラウンド許すことで離散対数問題の困難性仮定を基にしたマルチ署名を構成し、署名長は Schnorr 署名 [14] と同じである。近年、より強い困難性仮定を基にすることで、[10] の方式のラウンド数を減らしより効率性を高めた方式が提案されている [11, 12]。

3.2.3　グループ署名

■グループ署名の特徴　従来の電子署名では、署名検証の際に署名者が誰であるか明示的に分かってしまう。グループ署名は、従来の電子署名と異なり、署名者の匿名性を保証することができる特徴を有する。グループ署名における匿名性とは、ある組織（グループ）に属するメンバのだれかが署名したことしか検証者には分からない性質である。グループ署名では、署名者と検証者に加えて、発行者と呼ばれる各グループメンバに署名生成用の署名鍵を発行する管理者と、開示者と呼ばれる署名から真の署名者を開示する管理者を含む。ただし、それぞれの管理者は機

能分離されていなくてもよい。

署名者の匿名性を保証する高機能暗号技術としてリング署名 (第 3.2.4 章参照) があるが、グループ署名とリング署名は以下の相違がある。

- アドホック性：リング署名では各署名者は独自に鍵生成を行い検証鍵を公開し、署名生成時に任意の検証鍵集合を選択して含めることでグループをアドホックに構成できるのに対し、グループ署名では発行者と署名鍵を生成したユーザ集合がグループとなる。
- 署名者開示の有無：リング署名では署名鍵が開示されたとしても真の署名者であるかどうか分からないのに対し、グループ署名では開示者により真の署名者を開示できる。

この相違に応じ、リング署名は管理者無しの環境で強い匿名性を要求する状況、グループ署名は管理者がグループを管理し緊急時には真の署名者を開示可能であることを要求する状況で使い分けることが望ましい。

■グループ署名の歴史　グループ署名の概念は、Chaum と Van Heyst [15] によって初めて提案された。同時にいくつかの方式が提案されたが、グループメンバをセットアップ後は追加不可能である静的グループに制限されている方式であるか、もしくは開示者が真の署名者を開示する際にグループメンバとの対話が必要という問題があった。また、数学的な安全性定義が示されておらず、非形式的な安全性評価しかされていなかった。

その後、Bellare らによって、静的グループの場合 [16] と署名生成後でも新たなグループメンバを追加可能な動的グループの場合 [17] における統一的な数学的安全性定義がそれぞれ提案された。加えて、Bellare らは従来の電子署名、公開鍵暗号、非対話ゼロ知識証明から一般的に安全なグループ署名を構成する手法を提案した。Bellare らの構成法では、メンバ追加時は、各メンバは発行者から自分がグループに属していることを示す証明書を受け取り、証明書を含む署名鍵を生成する。署名生成時は、自分の ID に紐づいた証明書の暗号文とともに、証明書を知ることと暗号文が正しいことを証明する非対話ゼロ知識証明を署名として出力する。開示者による署名開示時は、署名に含まれる暗号文を復号し、ID を特定した上で、(必要に応じて) 開示内容が正しいことを示す非対話ゼロ知識証明を出力する。Bellare らの安全性定義に基づき、Pairing 写像を利用した署名長の短い方式 [18, 19] 等が提案されている。

また、グループメンバの証明書を失効する機能を持つ方式も知られており、グループ管理者が失効リストを配布し、署名者がそれに基づいて署名を生成し、検証

者が失効済みかどうか検証するリスト公開型失効 [20] と、署名生成時は失効リストを用いずに検証者のみがリスト検証する検証者ローカル型失効 [21] がある。検証者ローカル型失効では、新たな失効者が出る際に、その都度、グループメンバに通知する必要が無いため、リスト公開型失効よりも効率的である。

近年では、量子計算機実用化の研究の進展に伴い、大規模な量子計算機が実現したとしても安全性を保つ耐量子計算機安全な方式の研究が進んでおり、格子問題に基づく方式 [22, 23, 24] が主に提案されている。

■グループ署名方式の比較　一般に、グループ署名の安全性として以下の 3 つの性質が要求される。

- 匿名性：開示者以外（発行者も含む）は署名から真の署名者を識別できない。
- 追跡可能性：開示者が署名者を追跡不可能であり、かつ、検証に合格する署名を攻撃者は生成することができない。
- 濡れ衣不可能性：管理者であっても、署名を作成していないメンバに対して真の署名者であると証明できない。

Bellare らの構成法 [16, 17] 以降に提案された方式の多くは、彼らの構成法を基としており、上記の性質を満たしている。

また、グループ署名の効率は以下の観点で比較できる。

- グループ公開鍵長
- 署名長
- 署名計算量

Chaum と Van Heyst の方式 [15] の方式は、グループ公開鍵長、署名長、署名計算量のいずれもグループメンバ数に比例しているが、Camenisch と Stadler [25] によってグループ公開鍵長、署名長、署名計算量のいずれも定数長の RSA 仮定に基づく方式が提案され、以後古典計算機に対して安全な方式のほとんどは定数長を達成している。その中でも、Pairing 写像を利用したグループ署名方式 [18, 19] では署名長をグループ署名ではない CRYPTREC 暗号リスト [26] に掲載されている電子署名の数倍程度まで短くすることができる。耐量子計算機安全な方式では、グループ公開鍵長と署名計算量が定数長の方式 [24] やグループ公開鍵長と署名長が定数長の方式 [23] が知られている。

■代表的な方式のアルゴリズム　具体的なグループ署名の実現方法として、公開鍵暗号と電子署名に基づく静的な方式 [16] を紹介する。

Setup$(1^k, 1^n)$: 管理者鍵生成アルゴリズム。k をセキュリティパラメータ、n をグループの最大メンバ数とし、共通参照情報 crs、公開鍵暗号の復号鍵と暗号化鍵 (dk, ek)、電子署名の署名鍵と検証鍵 (msk, mvk) を生成し、発行者の秘密鍵 $ik = msk$、開示者の秘密鍵 $ok = dk$、グループ公開鍵 $gpk = (crs, mvk, ek)$ を出力する。

KeyGen(i, ik, gpk): メンバ鍵生成アルゴリズム。メンバ i は電子署名の署名鍵と検証鍵 (sk_i, vk_i) を生成し、vk_i を発行者に送る。発行者は msk を用いて、(i, vk_i) をメッセージとした署名 $cert_i$ を生成し、証明書としてメンバに送る。メンバは秘密鍵 $gsk_i = (i, sk_i, vk_i, cert_i)$ を出力する。

Sign(gsk_i, m): 署名者 i の署名生成アルゴリズム。メッセージ m に対して、sk_i を用いて署名 s を生成し、乱数 r を選び、ek を用いて $(i, vk_i, cert_i, s)$ を暗号化した暗号文 ct を計算する。crs を用いて、次のような関係を示す非対話ゼロ知識証明 π を生成する。 (mvk の下で $((i, vki), cert_i)$ が正しい署名) かつ (vk_i の下で (m, s) が正しい署名) かつ (ek の下で ct が $(i, vk_i, cert_i, s)$ の正しい暗号文) 署名 $\sigma = (ct, \pi)$ を出力する。

Verify(gpk, m, σ): 署名検証アルゴリズム。π が成り立つかチェックする。

Open(ok, gpk, m, σ): 署名者開示アルゴリズム。π が成り立つかチェックし、dk を用いて ct から $(i, vk_i, cert_i, s)$ を復号し、i を出力する。

3.2.4 リング署名

■リング署名の特徴 リング署名は、匿名署名の一種であり、あるグループの誰かが署名したことはわかる（偽造不可能性）が、そのグループの誰が署名したかはわからない（匿名性)、という性質を持っている。具体的には、リング署名は、リングと呼ばれる n 人の署名参加者の集合に対して、リングのなかの少なくとも 1 人の署名者が署名を生成したことは確認できるが、リングのなかの誰が署名を生成したかはわからない、ことを示す署名である。リング署名は、リングに属する n 人の署名参加者の公開鍵に対して、少なくとも 1 人の署名参加者の秘密鍵を知っていることを示す、非対話 1-out-of-n OR 証明と見ることもできる。

匿名署名の一種であるグループ署名（第 3.2.3 章参照）と比較すると、以下ような相違がある。

- アドホック性：リング署名では、各署名者は独自に鍵生成を行い検証鍵を公開し、署名生成時に署名者が任意の検証鍵集合を選択することでグループをアドホックに構成できる。一方、グループ署名では、発行者（グループ管理

者）と協力して署名鍵を生成したユーザの集合がグループとなる。
- 署名者開示の有無：リング署名では、署名者であっても、真の署名者を開示することはできない。一方、グループ署名では、開示者（グループ管理者）は、真の署名者を開示することができる。

このように、リング署名は、署名生成時にアドホックにグループを選ぶことができ、グループ管理者が必要なく非中央集権的である、という特徴を持っている。

■リング署名の歴史　リング署名は、Rivest, Shamir, Tauman [27] によって初めて提案され、落戸付一方向性置換に基づく方式が示された。その後、離散対数問題に基づくリング署名方式 [28] が提案された。これらの方式で用いられている典型的なリング署名の構成法では、n 人の署名参加者の集合であるリングに対して、リングのなかの署名者以外の署名参加者の公開鍵を用いて $n-1$ 個の値の連鎖を生成し、署名者の秘密鍵を用いて連鎖を環になるように連結する 1 個の値を生成することによって、値の連鎖の環を構成しそれを署名としている。署名者の秘密鍵を用いて連結した部分は、署名者の秘密鍵が分かったとしても特定することができず、強い匿名性を実現している。また、署名サイズは $\mathcal{O}(n)$ となっている。

通常のリング署名は前述の通り、署名者であっても自分が真の署名者であることを証明できない強い匿名性を保証しているが、応用のために匿名性を制限した方式も知られている。リンク可能リング署名 [29] では、同じ秘密鍵とタグを用いて異なる 2 つのメッセージについて生成された署名を、リンクすることができる。トレース可能リング署名 [30] では、同じ秘密鍵とタグを用いて異なる 2 つのメッセージについて生成された署名から、署名者を特定することができる。リング署名の非中央集権的な構成が暗号資産に適しており、上記の性質により暗号資産の二重使用を防ぐことができるため、リンク可能リング署名やトレース可能リング署名は匿名の暗号資産で用いられている。

また、しきい値リング署名 [31, 32] は、1-out-of-n のリング署名を k-out-of-n に一般化したものであり、n 人の署名参加者の集合であるリングに対して、リングのなかの少なくとも k 人の署名者が協力して署名を生成したことを確認することができる署名である。

■リング署名方式の比較　リング署名の安全性として以下の 2 つの性質が要求される。

- 匿名性: n 人のリングのなかの誰が署名者であるか、署名から特定できない。
- 偽造不可能性: n 人のリングのなかの少なくとも 1 人の署名者の秘密鍵がなければ、署名を生成できない。

リング署名方式の多くは、上記の性質を満たしている。匿名性を制限したものとして、前述のリンク可能リング署名とトレース可能リング署名がある。さらに、アカウント可能リング署名 [33] では、開示者を置くことによって署名者の開示を可能としている。ただし、アカウント可能リング署名では、グループ署名とは異なり発行者は必要なく署名生成時にグループを選ぶことができる。また、偽造不可能性を k-out-of-n に一般化したものとして、前述のしきい値リング署名がある。

また、リング署名の効率は以下の観点で評価される。

- 署名サイズ
- 署名生成と検証に必要な計算量

リング署名方式の多くは、リングの署名参加者の数を n とすると、署名サイズが $\mathcal{O}(n)$ である。1-out-of-n の効率的なゼロ知識証明を用いることで、署名サイズを $\mathcal{O}(\log\ n)$ にできる方式が知られている [34, 35, 36, 37]。また、信頼できるセットアップを仮定することで、アキュムレータにより n 個の公開鍵を圧縮し署名サイズを定数にできる方式も提案されている [38]。

また、基づいている計算量仮定の観点では、耐量子計算機安全性を持つリング署名方式が提案されている [39, 35, 36, 37]。

■代表的な方式のアルゴリズム　具体的なリング署名の実現方法として、Abe, Ohkubo, Suzuki による離散対数問題に基づく方式 [28] を紹介する。署名者 i について、q_i を素数、g_i を位数 q_i の巡回群 G_i の生成元、$H_i : \{0,1\}^* \to \mathbb{Z}_{q_i}$ をハッシュ関数とする。

$\mathsf{KeyGen}(q_i, G_i, g_i)$:　鍵生成アルゴリズム。

1. 署名者 i は、秘密鍵 $x_i \in \mathbb{Z}_{q_i}$ をランダムに選び、公開鍵 $(q_i, G_i, g_i, y_i = g_i^{x_i})$ を計算する。

$\mathsf{Sign}(x_a, L = \{y_1, ..., y_n\}, m)$:　署名生成アルゴリズム。

1. 署名者 $a \in \{1, ..., n\}$ は、秘密鍵 x_a と公開鍵のリスト $L = \{y_1, ..., y_n\}$ とメッセージ m を入力として受け取る。
2. 署名者 a は、$r_a \in \mathbb{Z}_{q_a}$ をランダムに選び、$e_a = g_a^{r_a}$ と $c_{a+1} = H_{a+1}(L, m, e_a)$ を計算する。
3. 署名者 a は、$s_i \in \mathbb{Z}_{q_i}$ をランダムに選び、$e_i = g_i^{s_i} y_i^{c_i}$ と $c_{i+1} = H_{i+1}(L, m, e_i)$ を、$i = a+1, ...n, 1, ..., a-1$（ただし $i = n+1 = 1$ とする）について計算する。
4. 署名者 a は、$s_a = r_a - c_a x_a \mod q_a$ を計算する。

5. 署名者 a は、署名 $\sigma = (c_1, s_1, ..., s_n)$ を出力する。

Verify$(L = \{y_1, ..., y_n\}, \sigma, m)$: 署名検証アルゴリズム。

1. 検証者は、公開鍵のリスト $L = \{y_1, ..., y_n\}$ と署名 $\sigma = (c_1, s_1, ..., s_n)$ とメッセージ m を入力として受け取る。
2. 検証者は、$e_i = g_i^{s_i} y_i^{c_i}$ と $c_{i+1} = H_{i+1}(L, m, e_i)$ を、$i = 1, ..., n-1$ について計算する。
3. 検証者は、$e_n = g_n^{s_n} y_n^{c_n}$ を計算する。
4. 検証者は、もし $c_1 = H_1(L, m, e_n)$ なら受理を出力し、そうでなければ非受理を出力する。

なお、この方式は、$n = 1$ のとき Schnorr 署名となっており、Schnorr 署名の自然な拡張になっている。

3.2.5 しきい値署名

■しきい値署名の特徴 (k, n) しきい値署名とは、ある $(k, n : k \leq n)$ である自然数の組に対して、以下のアルゴリズムから成る公開鍵署名方式である。

1. 公開鍵署名の署名生成鍵と検証鍵のペアを生成する。
2. 署名生成鍵を n 個のシェアに分割する（しきい値秘密分散）。シェアとは、元のデータを複数のデータに分割したものである。
3. 1 つのメッセージに対していくつかのシェアで署名を生成する。
4. シェアから生成された署名を k 個以上集め、組み合わせる処理を行う。
5. 組み合わされた署名に対し、検証鍵で検証を行い、検証結果を出力する。

偽造等の不正が行われず正当にこのアルゴリズムが実行された場合、第 4 項で組み合わされた署名は、第 1 項で生成された署名鍵を用いて生成された署名と同等となり、第 5 項の検証処理を通る署名となる。

しきい値署名は一般的な DSA 等の公開鍵署名方式と比較すると、ベースとなるしきい値秘密分散の特性によって署名生成鍵を複数にすることができるため、しきい値未満の漏洩や故障に対して優位性を持つ。類似したアクセス構造を実現する署名方式として「署名生成の際に設定された（しきい値等の）アクセス構造を満たす属性集合に対応する署名生成鍵を持つ署名者は単一の正当な署名を生成できる」属性ベース署名が存在するが、属性ベース署名では署名者は単一であり、単体ではアクセス構造を満たさない属性を複数の署名者が持ち寄っても正当な署名は作れない。一方でしきい値署名は複数の署名者に署名生成権限を分散すること

を想定しており、漏洩・故障等への耐性を持つことが属性ベース署名とは異なる。また、マルチパーティ計算のように複数の参加者が想定される署名プロトコルであるため、鍵生成については、信頼できる鍵生成・配布を行うトラステッドディーラーを仮定する場合と、各参加者が独自に鍵を生成した上で暗黙的に署名生成鍵や署名検証鍵が決定される分散鍵生成を用いる方式が存在する。

しきい値署名におけるの安全性の定義には以下が存在する。

- 偽造不可能性（Unforgeablity）：しきい値未満のシェアで生成された署名を集めただけでは、検証鍵で検証可能である正当な署名は生成不可能。さらに不正者の数によって以下のように定義が分かれる。
  - dishonest majority：不正者が n/2 以上しきい値未満
  - honest majority：不正者が n/2、かつしきい値未満
- 堅牢性（Robustness）：正規のシェアがしきい値以上集まれば、必ず正当な署名が生成可能
- Identifiable abort：正直な署名者は不正なシェアが少なくとも 1 つ以上含まれていることを検知可能
- Proactive security：長期的な期間でのシェアの漏洩に対して安全

また署名アルゴリズムとしての安全性もあるため、例えば後述する BLS 署名をベースとしたしきい値署名ではランダムオラクルモデルのもとで Gap Diffie-Hellman 問題の困難性を安全性の根拠として、選択メッセージ攻撃に対する存在的偽造不可能性が証明されている。

■しきい値署名の歴史　しきい値署名は 1987 年 Desmedt によって最初にコンセプトが提案 [40] され、同じく Desmedt と Frankel によって ElGamal 暗号 [41] と Shamir のしきい値秘密分散 [42] を組み合わせた方式が提案された [43]。Gennaro らは安全性証明がついた ElGamal 方式ベースのしきい値署名を 1996 年に発表し [44]、Shoup は同期通信を必要としない RSA 署名をベースとした方式を発表している [45]。また署名の検証に署名生成者の協力が必要な Undeniable な方式や、検証者を複数にする (k, l)-Shared Verification 方式等、様々な方式が提案されている。

近年では暗号資産に関連した技術として注目されている。例えば、Bitcoin では署名者のプライバシに関する問題やトランザクションのストレージ効率化に向けてしきい値署名を採用しやすい Schnorr 署名をとりいれている [46]。また、効率的なスキームとして FROST[47] 等が考案されており、利用されるケースが増えてきている。ECDSA を取り入れたしきい値署名に関する提案が数多くあり [48] セ

キュリティの強化やプロトコルの効率化を目指して様々な方式があげられている [49, 50, 51, 52, 53, 54, 55, 56]。最近では Groth と Shoup によって効率的で堅牢な方式 [57] が提案されている。

■しきい値署名方式の比較　しきい値署名は様々な方式が提案されているがベースとなる署名方式やしきい値秘密分散・分散鍵生成の方式によって効率性や安全性が異なる。

効率性では、(1) 鍵長・署名長などの各パラメータサイズ、(2) シェア生成・組み合わせ処理などの各オペレーション速度、(3) エンティティ間の同期が必要か否かの違いがあげられるが、(1)(2) はベースとなる署名方式や定義されている演算処理（楕円曲線演算や Pairing 写像）によって異なる。例えば、128bit セキュリティ程度のパラメータを基準とすると ECDSA（secp256r1）および EdDSA（Ed25519）ベースのしきい値署名 [57, 47] の各シェア自身のサイズは署名生成鍵と同等で 256bit、シェアから生成される署名長は 512bit となる。BLS 署名（BLS12-381）ベースのしきい値署名 [58] の場合、各シェアのサイズは 256bit、署名長は 768bit となる [59]。なお、数値については圧縮を使った表現も含まれており、シェアに対するコミットメントの保持や通信回数の効率化や安全性の向上のために事前計算を行った signature helpers を共有することもあるため、参加者が保持するデータ量はこれより増加することがある。(3) については方式が同期式・非同期式なのかによって異なり先の項目でもあげたが事前計算を行うことによって非同期式を実現し高速化を行うことがある。また鍵生成方式においてもベースとなる分散鍵生成方式の違いによって決まってくる。例えば通信ラウンド数の比較では Pedersen-DKG[60] は 3 ラウンド、Gennaro-DKG[61] は 4 ラウンド、FROST-DKG[47] は 1 ラウンドというように方式によって大きな違いがありネットワークレイテンシを考慮すると性能に大きな差が生じる可能性がある。しきい値署名に関する各文献で多くの方式が比較されているが、最近の比較では [56] の Figure1 や [62] の Table1 などがあげられる。

■代表的なアルゴリズム・方式　ここでは Ethereum などで既に用いられている Boldyreva が提案した BLS 署名 [13] ベースのしきい値署名 [58] を紹介する。以下では、(t, n)-しきい値署名とし、n 人の参加者を $P_1, ..., P_n$ とする

Setup(1^κ): BLS 署名のセットアップはセキュリティパラメータ κ を入力にとり、素数 p、p を位数とする巡回群 G_1, G_2, G_T、各巡回群を使って定義される Pairing 写像 $e : G_1 \times G_2 \to G_T$、文字列から G_1 の元へのハッシュ関数 $H : \{0,1\}^* \to G_1$、G_2 の元からランダムに決定した Q を準備し公開パラ

メータ $Params := (p, G_1, G_2, G_T, e, H, Q)$ とする

SharedKeygen$(Params, t, n)$: 分散鍵生成アルゴリズム [61] を利用して BLS 署名鍵ペア (vk, sk)（ただし、$sk \in \mathbb{Z}_p^*, vk = skQ$) に対するシェア $\{vk_i, sk_i : 1 \leq i \leq n\}$ を (t, n) 秘密分散になるよう各参加者が保持する。

SharedSign$(Params, m, sk_i)$: 参加者はメッセージ m に対して $\sigma_i = sk_i H(m)$ を計算し提出する

Verify$((Params, m, \{\sigma_{i_j}\}_{1 \leq j \leq t}, \{vk_{i_j}\}_{1 \leq j \leq t}, vk))$: 1. 参加者から収集した各 j に対して、$e(\sigma_{i_j}, Q) = e(H(m), vk_{i_j})$ が成り立つことを確認する

2. 正規の署名を $\sigma = \sum_{1 \leq j \leq t} L(i_j)\sigma_{i_j}$ により計算する。ただし、$L(i_j)$ は分散鍵生成におけるラグランジュ補間係数である。
3. vk を使って $e(\sigma, Q) = e(H(m), vk)$ により署名を検証する。

参考文献

[1] H. K. Maji, M. Prabhakaran and M. Rosulek: "Attribute-Based Signatures", CT-RSA 2011, Springer-Verlag, pp. 376–392 (2011).

[2] Y. Sakai, N. Attrapadung and G. Hanaoka: "Attribute-Based Signatures for Circuits from Bilinear Map", PKC 2016, Springer-Verlag, pp. 283–300 (2016).

[3] Y. Sakai, S. Katsumata, N. Attrapadung and G. Hanaoka: "Attribute-Based Signatures for Unbounded Languages from Standard Assumptions", ASIACRYPT 2018, Springer-Verlag, pp. 493–522 (2018).

[4] A. El Kaafarani and S. Katsumata: "Attribute-Based Signatures for Unbounded Circuits in the ROM and Efficient Instantiations from Lattices", PKC 2018, Springer-Verlag, pp. 89–119 (2018).

[5] J. Groth and A. Sahai: "Efficient Noninteractive Proof Systems for Bilinear Groups", SIAM J. Computing, **41(5)**, pp. 1193–1232 (2012).

[6] J. Groth, R. Ostrovsky and A. Sahai: "New Techniques for Noninteractive Zero-Knowledge", J. ACM, **59(3)**, pp. 1–35 (2012).

[7] J. Katz and A. Y. Lindell: "Aggregate Message Authentication Codes", CT-RSA 2008, Springer-Verlag, pp. 155–169 (2008).

[8] D. Boneh, C. Gentry, B. Lynn and H. Shacham: "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps", EUROCRYPT 2003, Springer-Verlag, pp. 416–432 (2003).

[9] D. Boneh, M. Drijvers and G. Neven: "Compact Multi-Signatures for Smaller Blockchains", ASIACRYPT 2018, Springer-Verlag, pp. 435–464 (2018).

[10] M. Bellare and G. Neven: "Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma", ACM CCS 2006, pp. 390–399 (2006).

[11] C. Komlo and I. Goldberg: "FROST: Flexible Round-Optimized Schnorr Threshold Signatures", SAC 2020, Springer-Verlag, pp. 34–65 (2020).

[12] J. Nick, T. Ruffing and Y. Seurin: "MuSig2: Simple Two-Round Schnorr Multi-Signatures", CRYPTO 2021, Springer-Verlag, pp. 189–221 (2021).

[13] D. Boneh, B. Lynn and H. Shacham: "Short Signatures from the Weil Pairing", J. of Cryptology, **17(4)**, pp. 297–319 (2004).

[14] C.-P. Schnorr: "Efficient Identification and Signatures for Smart Cards", CRYPTO 1989, Springer-Verlag, pp. 239–252 (1989).

[15] D. Chaum and E. V. Heyst: "Group Signatures", EUROCRYPT 1991, Springer-Verlag, pp. 257–265 (1991).

[16] M. Bellare, D. Micciancio and B. Warinschi: "Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions", EUROCRYPT 2003, Springer-Verlag, pp. 614–629 (2003).

[17] M. Bellare, H. Shi and C. Zhang: "Foundations of Group Signatures: The Case of Dynamic Groups", CT-RSA 2005, Springer-Verlag, pp. 136–153 (2005).

[18] D. Boneh, X. Boyen and H. Shacham: "Short Group Signatures", CRYPTO 2004, Springer-Verlag, pp. 41–55 (2004).

[19] J. Groth: "Fully Anonymous Group Signatures without Random Oracles", ASIACRYPT 2007, Springer-Verlag, pp. 164–180 (2007).

[20] J. Camenisch and A. Lysyanskaya: "Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials", CRYPTO 2002, Springer-Verlag, pp. 61–76 (2002).

[21] B. Boneh and H. Shacham: "Group Signatures with Verifier-Local Revocation", ACM CCS 2004, pp. 168–177 (2004).

[22] S. D. Gordon, J. Katz and V. Vaikuntanathan: "A Group Signature Scheme from Lattice Assumptions", ASIACRYPT 2010, Springer-Verlag, pp. 395–412 (2010).

[23] S. Katsumata and S. Yamada: “Group Signatures Without NIZK: From Lattices in the Standard Model”, EUROCRYPT 2019, Springer-Verlag, pp. 312–344 (2019).

[24] V. Lyubashevsky, N. Nguyen, M. Plancon and G. Seiler: “Shorter Lattice-Based Group Signatures via “Almost Free” Encryption and Other Optimizations”, ASIACRYPT 2021, Springer-Verlag, pp. 218–248 (2021).

[25] J. Camenisch and M. Stadler: “Efficient group signature schemes for large groups”, CRYPTO 1997, Springer-Verlag, pp. 410–424 (1997).

[26] CRYPTREC: “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”. https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2012r6.pdf.

[27] R. Rivest, A. Shamir and Y. Tauman: “How to Leak a Secret”, ASIACRYPT 2001, Springer-Verlag, pp. 552–565 (2001).

[28] M. Abe, M. Ohkubo and K. Suzuki: “1-out-of-n Signatures from a Variety of Keys”, ASIACRYPT 2002, Springer-Verlag, pp. 415–432 (2002).

[29] J. Liu, V. Wei and D. Wong: “Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract)”, ACISP 2004, Springer-Verlag, pp. 325–335 (2004).

[30] E. Fujisaki and K. Suzuki: “Traceable Ring Signature”, PKC 2007, Springer-Verlag, pp. 181–200 (2007).

[31] E. Bresson, J. Stern and M. Szydlo: “Threshold Ring Signatures and Applications to Ad-hoc Groups”, CRYPTO 2002, Springer-Verlag, pp. 465–480 (2002).

[32] M. Abe, M. Ohkubo and K. Suzuki: “Efficient Threshold Signer-Ambiguous Signatures from Variety of Keys”, IEICE Trans., **E87-A(2)**, pp. 471–479 (2004).

[33] S. Xu and M. Yung: “Accountable Ring Signatures: A Smart Card Approach”, CARDIS, vol. 153 of IFIP, pp. 271–286 (2004).

[34] J. Groth and M. Kohlweiss: “One-out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin”, EUROCRYPT 2015, Springer-Verlag, pp. 253–280 (2015).

[35] W. Beullens, S. Katsumata and F. Pintore: “Calamari and Falafl: Logarithmic (Linkable) Ring Signatures from Isogenies and Lattices”, ASIACRYPT 2020, Springer-Verlag, pp. 464–492 (2020).

[36] V. Lyubashevsky, N. Nguyen and G. Seiler: “SMILE: Set Membership from

Ideal Lattices with Applications to Ring Signatures and Confidential Transactions", CRYPTO 2021, Springer-Verlag, pp. 611–640 (2021).

[37] T. Yuen, M. Esgin, J. Liu, M. Au and Z. Ding: "Dualring: Generic Construction of Ring Signatures with Efficient Instantiations", CRYPTO 2021, Springer-Verlag, pp. 251–281 (2021).

[38] Y. Dodis, A. Kiayias, A. Nicolosi and V. Shoup: "Anonymous Identification in Ad Hoc Groups", EUROCRYPT 2004, Springer-Verlag, pp. 609–626 (2004).

[39] M. Esgin, R. Zhao, R. Steinfeld, J. Liu and D. Liu: "MatRiCT: Efficient, Scalable and Post Quantum Blockchain Confidential Transactions Protocol", ACM CCS 2019, pp. 567–584 (2019).

[40] Y. Desmedt: "Society and Group Oriented Cryptography: A New Concept", CRYPTO 1987, Springer-Verlag, pp. 120–127 (1987).

[41] T. ElGamal: "A Public Key Cryptosystem and Signature Scheme Based on Discrete Logarithms", IEEE Trans. Inform. Theory, **31**, pp. 469–472 (1985).

[42] A. Shamir: "How to Share a Secret", Commun. ACM, **22**, 11, pp. 612–613 (1979).

[43] Y. Desmedt and Y. Frankel: "Threshold Cryptosystems", CRYPTO 1989, Springer-Verlag, pp. 307–315 (1989).

[44] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin: "Robust Threshold DSS", EUROCRYPT 1996, Springer-Verlag, pp. 354–371 (1996).

[45] V. Shoup: "Practical Threshold Signatures", EUROCRYPT 2000, Springer-Verlag, pp. 207–220 (2000).

[46] "BBitcoin Improvement Proposal 340". `https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki`.

[47] C. Komlo and I. Goldberg: "FROST: Flexible Round-Optimized Schnorr Threshold Signatures". ePrint 2020/852.

[48] J. P. Aumasson, A. Hamelink and O. Shlomovits: "A Survey of ECDSA Threshold Signing". ePrint 2020/1390.

[49] Y. Lindell, A. Nof and S. Ranellucci: "Fast Secure Multiparty ECDSA with Practical Distributed Key Generation and Applications to Cryptocurrency Custody", ACM CCS 2018, pp. 1837–1854 (2018).

[50] R. Gennaro and S. Goldfeder: "Fast Multiparty Threshold ECDSA with Fast Trustless Setup", ACM CCS 2018, pp. 1179–1194 (2018).

[51] J. Doerner, Y. Kondi, E. Lee and A. Shelat: "Secure Two-Party Threshold ECDSA from ECDSA Assumptions: The multiparty Case", IEEE S&P 2018, pp. 980–997 (2018).

[52] I. Damgard, T. P. Jakobsen, J. B. Nielsen, J. I. Pagter and M. B. Østergaard: "Fast Threshold ECDSA with Honest Majority", SCN 2020, Springer-Verlag, pp. 382–400 (2020).

[53] R. Gennaro and S. Goldfeder: "One Round Threshold ECDSA with Identifiable Abort". ePrint 2020/540.

[54] G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta and I. Tucker: "Bandwidth-Efficient Threshold EC-DSA", PKC 2020, Springer-Verlag, pp. 266–296 (2020).

[55] A. Gagol, J. Kula, D. Straszak and M. Świetek: "Threshold ECDSA for Decentralized Asset Custody". ePrint 2020/498.

[56] R. Canetti, R. Gennaro, S. Goldfeder, N. Makriyannis and U. Peled: "UC Non-Interactive, Proactive, Threshold ECDSA with Identifiable Aborts", ACM CCS 2020, pp. 1769–1787 (2020).

[57] J. Groth and V. Shoup: "Design and Analysis of A Distributed ECDSA Signing Service". ePrint 2022/506,.

[58] A. Boldyreva: "Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman Group Signature Scheme", PKC 2003, Springer-Verlag, pp. 31–46 (2003).

[59] Y. Sakemi, T. Kobayashi, T. Saito and R. S. Wahby: "Draft-irtf-cfrg-pairing-friendly-curves-10" (2021).

[60] T. P. Pedersen: "A Threshold Cryptosystem without a Trusted Party", EUROCRYPT 1991, Springer-Verlag, pp. 522–526 (1991).

[61] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin: "Secure Distributed Key Generation for Discrete-Log Based Cryptosystems", EUROCRYPT 1999, Springer-Verlag, pp. 295–310 (1999).

[62] M. Pettit: "Efficient Threshold-Optimal ECDSA", CANS 2021, Springer-Verlag, pp. 116–135 (2021).

3.3 その他の高機能暗号技術

3.3.1 秘密分散

■秘密分散法の特徴 秘密分散法（Secret Sharing Scheme，SSS）は現在、様々な暗号技術の要素技術として用いられているが、提案された当時は秘密鍵などの安全な管理を目的としていた [1, 2]。通信の秘匿性を担保するために保持しなければならない秘密鍵のような情報を保管する際に、破損や紛失を避けるためにはいくつかのコピーを作る必要がある一方で、コピーを作ることは盗難や漏洩の危険性を増大させるといった相反する問題が存在する。そこで、秘密情報を「シェア」と呼ばれる n 個の分散情報に符号化し、そのうちの任意の $k \leq n$ 個のシェアを集めることで秘密が復元できるが、$k-1$ 個では秘密情報が情報理論的に一切漏洩しない方式を考える。このような方式は (k, n) しきい値法と呼ばれ、$n-k$ 個以下の破損や紛失、$k-1$ 個以下の盗難や漏洩に対して安全となることから、上記の問題に対する一つの解決策となる。(k, n) しきい値法は、秘密情報へのアクセスを許されるシェア集合と、許されないシェア集合を指定可能な秘密分散法に拡張することができる。このような秘密分散法を、一般アクセス構造に対する秘密分散法と呼ぶ [3]。また、可検証秘密分散法（Verifiable SSS、[4]）などをはじめとして様々な機能拡張がなされている。

秘密分散法は、上記のような秘密情報の分散保管を目的として提案されたが、現在ではマルチパーティ計算（Multi-Party Computation，MPC）をはじめとした様々な暗号プロトコルの要素技術として用いられている [5]。また、様々な媒体でも実現手法が知られており、視覚暗号 [6]、量子秘密分散法 [7, 8] などが提案されている。

■秘密分散法の歴史、実装例 秘密分散法は 1979 年に Shamir [9] と Blakley [2] によって独立に提案された。どちらの論文も (k, n) しきい値法の構成手法であるが、Shamir は多項式補間（Lagrange 補間）による方法を、Blakley は射影幾何に基づく方法を提案している。これらの手法は線形変換のランクを元にした構成法に一般化できる [10]。このような線形変換のランク関数と関連付ける研究は一般アクセス構造をもつ秘密分散法の構成法やシェアサイズの下界を求める際にも有用であり、ポリマトロイドの性質などと関連付けられて近年でも研究が進んでいる [11]。また、線形代数的なアプローチとして Monotone span program による構成法が挙げられる [12]。

文献 [10] では線形変換に基づく構成法に加えて、情報理論的な手法でシェアの

サイズの下界が導出されており、秘密情報へのアクセスが許されないシェア集合から 1 ビットも情報が漏れない完全秘匿性という性質を要求する場合、秘密情報 S を確率変数としたときの S のシャノンエントロピー（直観的にはビット長）$H(S)$ は i 番目のシェア V_i のシャノンエントロピー $H(V_i)$ に対して常に $H(V_i) \geq H(S)$ が成り立つ。Shamir の手法は任意の i に対して $H(V_i) = H(S)$ が成り立つことが容易に分かるため、Shamir の構成法がシェアサイズの観点からは最良であることになる。これらの結果を受けて、完全秘匿性を弱める代わりに、シェアのサイズを秘密情報のサイズより小さくするランプ型秘密分散法（ramp SSS）が提案された [13, 14]。ランプ型秘密分散法は Shamir の秘密分散法の一般化として構成できる。

計算量的秘密分散法は、Shamir 法と共通鍵暗号を組み合わせることで実現でき、Shamir 法に比べて大幅にシェアサイズが削減できることが知られている [15]。近年では、計算量的 (n, n) しきい値法として All-Or-Nothing Transform（AONT）が知られている。Shin らによる AONT [16] は ZenmuTech 社によって実装され、実用化されている [17]。

■秘密分散法の代表的アルゴリズム　秘密分散法は極めて多くの方式が提案されているため、最も基本的で利用頻度が多いと思われる Shamir の多項式補間によるアルゴリズム [9]、およびその拡張方式について、ISO/IEC 標準 [18] に関連するいくつかの方式を説明する。以下では全ての計算を有限体 $\mathbb{F}$ の上で行うものとする。また、自然数 n をシェアの総数として、参加者の集合を $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ とし、参加者 p_i がシェア v_i を持つとする。

Shamir 法とランプ型秘密分散法

はじめに最も基本的な Shamir の (k, n) しきい値法 [9]（Shamir 法）を説明する。秘密情報 $ss \in \mathbb{F}$ をもつディーラは、秘密情報と独立で、互いに独立かつ一様な乱数を $k-1$ 個選び、$r_1, r_2, \ldots, r_{k-1} \in \mathbb{F}$ とする。このようにして選んだ値から、$k-1$ 次のランダム多項式 $f(x) := s + r_1 x^1 + r_2 x^2 + \cdots + r_{k-1} x^{k-1}$ を作成し、参加者 P_i にシェアとして $v_i := f(i)$ を秘密通信路を介して配布する。復号にあたっては、任意の k 人の参加者 $P_{i_1}, P_{i_2}, \ldots, P_{i_k}$ が各自のシェア $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ をもって集まり、多項式 $f(x)$ を復元することによって秘密情報 $s = f(0)$ が復元される。多項式は Lagrange 補間によって求められるので、秘密情報は $s = f(0) = \sum_{\alpha=1}^{k} v_{i_\alpha} \prod_{\beta=1, \beta\neq\alpha}^{k} (-i_\beta)/(i_\alpha - i_\beta)$ で求められる。

完全秘匿性を達成する秘密分散法では、シェアのサイズをこれ以上短くできないことが知られている [10]。ただし、秘密情報の一部を漏らすことを許せば、シェアのサイズを小さくすることができる。このように、情報漏洩とシェアのサイズのトレードオフを実現する秘密分散法がランプ型秘密分散法である [13, 14]。ラン

プ型秘密分散法は秘密情報を $s_0, s_1, \ldots, s_{L-1} \in \mathbb{F}$ として、Shamir 法の多項式 $f(\cdot)$ の代わりに $g(x) := \sum_{i=0}^{L-1} s_i x^1 + \sum_{i=L}^{k-1} r_i x^1$ を用いる事で構成できる。シェアが k 個以上ある場合、Shamir 法と同様に Lagrange 補間を用いて復号が可能であり、シェアが $k-L$ 個以下なら s に関する情報が情報理論的に（完全）秘匿される。一方で、シェアを $k-\ell$ $(1 \leq \ell \leq L-1)$ 個集めた場合は秘密情報は条件付エントロピーの意味で $(1-\ell/L)H(S)$ だけ情報が漏洩する。

ただし、完全秘匿性を満たす Shamir 法では秘密情報 S はどのような確率分布に従っても良いが、ランプ型秘密分散法は $s_0, s_1, \ldots, s_{L-1} \in \mathbb{F}$ を独立かつ一様ランダムに選ぶ必要があることに注意する。また、ランプ型秘密分散における情報漏洩は、秘密情報が存在しうる空間のサイズが $\mathbb{F}^L$ ではなく $\mathbb{F}$ となることを表し、秘密情報の探索空間のサイズは同じでも、秘密情報の一部が一意に定まる場合と一意に定まらない場合がある。秘密情報のどのシンボルも一意に定まらないランプ型秘密分散法は強いランプ型秘密分散（strong ramp SSS）と呼ばれるが [14]、上に示した Shamir 法のランプ型への拡張は必ずしも強いランプ型秘密分散法になるとは限らないことが知られている [19]。

一般アクセス構造への拡張と複製型秘密分散法

しきい値型秘密分散法を用いると、一般アクセス構造に対する秘密分散法を構成することもできる。まず、参加者の集合を $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ に対する一般アクセス構造について簡単に説明する。アクセス構造とは参加者の集合 $\mathcal{X} \subseteq \mathcal{P}$ に秘密情報 $s \in \mathbb{F}$ へのアクセス権を与えるかどうかを決定するものであり、秘密情報へのアクセス権を有する有資格集合の族 $\Gamma_{\mathsf{Q}} \subseteq 2^{\mathcal{P}}$ とアクセス権のない禁止集合の族 $\Gamma_{\mathsf{F}} \subseteq 2^{\mathcal{P}}$ からなる。$\Gamma_{\mathsf{Q}} \cap \Gamma_{\mathsf{F}} = \emptyset$ が要請される一方で、$\Gamma_{\mathsf{Q}} \cup \Gamma_{\mathsf{F}} = 2^{\mathcal{P}}$ である必要はない。また、アクセス構造が自然に意味をもつために、任意の $\mathcal{A} \in \Gamma_{\mathsf{Q}}$ および任意の $\mathcal{A}' \supseteq \mathcal{A}$ に対して $\mathcal{A}' \in \Gamma_{\mathsf{Q}}$ であること、および、任意の $\mathcal{A} \in \Gamma_{\mathsf{F}}$ および任意の $\mathcal{A}' \subseteq \mathcal{A}$ に対して $\mathcal{A}' \in \Gamma_{\mathsf{F}}$ であることが必要となる。この性質を単調性と呼ぶ。

単調性をもつ任意のアクセス構造に対する秘密分散法は [3] で初めて提案されたが、ここでは [20] による、加法的秘密分散法を用いた一般アクセス構造に対する秘密分散法を説明する。

加法的秘密分散法とは、有限体における秘密鍵の加算と減算で暗号化と復号をおこなう、いわゆる使い捨て暗号 [21] の拡張であり、秘密情報 $s \in \mathbb{F}$ に対して (m, m) しきい値法を実現する方法である。具体的には、独立で一様な乱数 $r_1, r_2, \ldots, r_{m-1} \in \mathbb{F}$ を用意し、$i = 1, 2, \ldots, m-1$ に対するシェアを $v_i = r_i$ とする。そして、m 番目のシェアは $v_m = s - \sum_{i=1}^{m-1} r_i$ により計算し、m 個のシェアとする方法である。また、秘密情報 s は m 個のシェアを全て集め、$s = \sum_{i=1}^{m} v_i$

により計算できる。この加法的秘密分散法を用いた一般アクセス構造に対する秘密分散法 [20] は、任意の $\mathcal{A} \in \Gamma_{\mathsf{A}}$ に対して、s を秘密情報とした加法的秘密分散法（$(|\mathcal{A}|, |\mathcal{A}|)$ しきい値法）を独立に実現し、それぞれの加法的秘密分散法で p_i に割り当てられるシェアをまとめて v_i と考えることで実現できる。

なお、[3] で提案された方式は単一の (m, m) しきい値法のシェアを複製型秘密分散法を用いて、適切に配布することで一般アクセス構造を実現している。第 3.3.2 章「MPC - 秘密分散ベース」の項で紹介する「参加者が 3 人の複製秘密分散法に基づく方法」で用いる秘密分散法はこれの特殊ケースであり、加法型の $(3, 3)$ しきい値法に対して複製型秘密分散法を適用して構成された $(2, 3)$ しきい値の秘密分散法である。

3.3.2 マルチパーティ計算–秘密分散ベース–

秘密分散ベースのマルチパーティ計算（Multi-Party Computation，MPC）は、第 3.3.3 章に記載する Garbled Circuit ベースのマルチパーティ計算と異なり、乗算回数などに応じた通信回数が必要となる。一方で、Garbled Circuit ベースの MPC において支配的であった、共通鍵暗号による真理値表の暗号文の送受信を必要としない分、通信量の面では秘密ベース分散ベースの MPC の方が小さくなる。この特徴から、秘密分散ベースの MPC は Garbled Circuit に基づく方法と比較して、通信量を抑えたい状況の処理に適していると考えられている。秘密分散ベースの MPC と Garbled Circuit ベースの MPC とを組み合わせて用いることもあり、特に 3 者以上の多人数の MPC において、定数回の通信による方式を実現するときなどに利用される [22, 23]。

■秘密分散ベースのマルチパーティ計算の歴史　秘密分散ベースのマルチパーティ計算の草分けとして、Goldreich らによる GMW 方式 [24] や Ben-Or らによる BGW 方式 [25] が知られている。GMW 方式は、加法的秘密分散法に基づく MPC プロトコルである。秘密情報を「シェア」と呼ばれる分散情報に分割して符号化し、シェアから秘密情報を復元することなく、秘密情報どうしの加算・乗算の計算を実現する。本章で紹介する MPC は秘密分散ベースであるため、このシェアとは、第 3.3.1 章に記載したシェアと同じものであり、元となるデータを分割したものとなる。この方式では、紛失通信と呼ばれる暗号プロトコルを利用して乗算を実現している。BGW 方式は、Shamir の秘密分散法 [1] に基づく MPC プロトコルであり、GMW 方式と同様に秘密情報のシェアを復元することなく、秘密情報どうしの加算・乗算の計算を実現する。この方式では、ローカルでの計算による乱数の偏り、多項式の次数増加を回避するため、計算結果を再分散することによって再

ランダム化・次数の削減を行って乗算を実現している。

GMW 方式と BGW 方式やその後に提案された多くの具体的な MPC プロトコルは、共通する基本的な枠組みの中で構築されている。それは、線形秘密分散法と総称される方式群に含まれる秘密分散法をベースとし、線形秘密分散法の性質を利用したローカル計算のみからなる加算プロトコルを持ち、他の参加者との通信を必要とする乗算プロトコルを備える、というものである。そのため、複雑なデータ処理を MPC で実行する際は、当該処理を論理回路/算術回路として見なした場合の AND/乗算の回数分の通信が発生し、これが処理時間全体の多くの部分を占めることになる。従って、この点に関する改善が処理全体の高速化に寄与すると考えられ、さまざまな改良がなされている。アルゴリズムの改良や実装の洗練化により、近年は実用的な処理性能が得られており、たとえば、NEC などが 7,000,000,000 ゲート/秒を達成 [26] し、また、産総研などは、データベース処理等における重要な構成要素となる秘匿状態でのシャッフルを加算・乗算のレベルまで分解せずに効率的に実現する手法を提案し、当該処理について従来の 100 倍程度の高速化に成功している [27]。

■代表的な方式 1: 参加者が 3 人の複製秘密分散法に基づく方法

効率的な方式の代表例として、参加者が 3 人の場合に向けて設計された、Araki らの方法 [26] を単純化した MPC プロトコル [28] を紹介する。以下では、各値はある整数 q を法とする剰余環の元とする。[28] の方式は、以下のような 3 つのうち任意の 2 つのシェアから復元可能な複製秘密分散法に基づいている。秘密情報 x の秘密分散は、$x = x_1 + x_2 + x_3$ を満たす 3 つの乱数 x_1，x_2，x_3 に加法的秘密分散し、それらのうち 2 つずつの対 (x_2, x_3)、(x_3, x_1)、(x_1, x_2) をそれぞれ参加者 P_1, P_2, P_3 のシェアとすることにより実現される。復元の際は、いずれか 2 つのシェアを集め、$x_1 + x_2 + x_3$ を計算すればよい。

加算のプロトコルは、各参加者が自身の持つ 2 つのシェアを加算することで実現される。2 つの秘密情報 x，y が秘密分散されているとき、$z := x + y$ の秘密分散を計算する加算のプロトコルを以下に示す。各参加者 P_i $(i \in \{1, 2, 3\})$ は $z_{i+1} := x_{i+1} + y_{i+1}$ と $z_{i+2} := x_{i+2} + y_{i+2}$ を計算し、(z_{i+1}, z_{i+2}) を z を秘密分散した P_i のシェアとする。ただし、添字の 4 は 1 と読み替えるものとし、以下も同様とする。正当性は $z_1 + z_2 + z_3 = (x_1 + y_1) + (x_2 + y_2) + (x_3 + y_3) = (x_1 + x_2 + x_3) + (y_1 + y_2 + y_3) = x + y = z$ により確認できる。このプロトコルは一切の通信を必要とせず、各参加者のローカルの計算で完結している。

乗算のプロトコルを以下に示す。入力は加算の場合と同じとし、$w := xy$ の秘密分散を計算するものとする。

1. 参加者間で 0 を加法的秘密分散し、各参加者 P_i $(i \in \{1,2,3\})$ は a_i を得る。ここで、a_1，a_2，a_3 は $a_1 + a_2 + a_3 = 0$ を満たす乱数である。
2. 各参加者 P_i $(i \in \{1,2,3\})$ は $w_{i+2} := x_{i+1}y_{i+1} + x_{i+1}y_{i+2} + x_{i+2}y_{i+1} + a_i$ を計算し、P_{i+1} に送る。
3. (w_{i+2}, w_{i+3}) を w の秘密分散の P_i $(i \in \{1,2,3\})$ のシェアとする。

基本的なアイデアは、xy を分解した $\sum_{i,j\in\{1,2,3\}} x_i y_j$ の 9 個の項のうち、参加者 P_i が自身の持つシェアから計算可能な $x_{i+1}y_{i+1} + x_{i+1}y_{i+2} + x_{i+2}y_{i+1}$ を計算結果 w の加法的秘密分散のシェアとするものである。この値は入力 x, y に関する情報を含んでいるため、ステップ 2 で隣の参加者に送る前に乱数 a_i を加算することによって見かけ上ランダムな値にしている。ステップ 1 の 0 の加法的秘密分散の作成は、各参加者が乱数を作成して次の参加者へ送り、自身の作った乱数から受け取った乱数を引いた値をシェアとすることで実現できる。正当性は、$w_1 + w_2 + w_3 = (x_3y_3 + x_3y_1 + x_1y_3 + a_1) + (x_1y_1 + x_1y_2 + x_2y_1 + a_2) + (x_2y_2 + x_2y_3 + x_3y_2 + a_3) = (x_1 + x_2 + x_3)(y_1 + y_2 + y_3) + (a_1 + a_2 + a_3) = xy = w$ により確認できる。このプロトコルは通信を伴い、1 回の実行につき各参加者は 2 つの値を送信する。さらに、pseudorandom secret sharing [29] と呼ばれる方法を使うと、送信する乱数の一部を隣接する参加者間で共有していたシードから生成した擬似乱数で代用でき、各参加者が送信する値を 1 つにまで減らすことができる。

本方式は、NTT の秘密計算システム算師 [30] などに実装され実用化されている [31]。

■代表的な方式 2: Beaver triple を使う乗算プロトコル

参加者が 3 人の複製秘密分散法に基づく方法で述べた MPC プロトコルとは異なる特性を持つ代表的な方式として、Beaver 乗算と呼ぶ乗算プロトコルを紹介する。この方式は Beaver triple [32] と呼ばれる互いに関連した乱数の組を利用するものであり、任意の線形秘密分散法の上で構築することができる。また、事前計算を行うことにより、入力を受け取った後に必要な通信を少量にすることができるという特徴を持つ。

Beaver 乗算の基本的なアイデアは以下の通りである。乗算の実行に先立って、2 つの乱数とそれらの積をそれぞれ秘密分散したシェアの 3 つ組（これを Beaver triple と呼ぶ）を用意しておく。乗算の実行時には、まず、入力の 2 つの値に Beaver triple の 2 つの乱数をそれぞれ減算し、入力値と乱数の差分値のシェアを各参加者と通信して共有する。そして、各参加者は共有した値から差分値を復元する。差分値は乱数によりマスクされているので、復元しても入力値が参加者に

知られることはなく安全である。そして、これらの復元した差分値どうしの積を計算する。この計算は公開値どうしの計算であるから、ローカルで計算可能である。この公開値どうしの積と元々欲しかった入力の積との差は、Beaver triple の各値の（マスクして復元した値を定数として使った）線型結合として表現でき、線形秘密分散法の性質によりローカルで計算できる。すなわち、Beaver triple を消費することで、入力をマスクした値の復元以外はローカル計算だけで乗算が実現される。

Beaver 乗算のプロトコルを説明する。以下では、ある値 x が秘密分散された状態を $[x]$ と書くこととし、秘密分散された x と y の加減算をそれぞれ $[x]+[y]$ と $[x]-[y]$、秘密分散された x の定数 c による定数倍を $c[x]$ と書くことにする。Beaver triple は、a と b を乱数、$c := ab$ として、$([a],[b],[c])$ とする。乗算プロトコルの入力は $[x]$ と $[y]$ とし、$w := xy$ を満たす $[w]$ を出力とする。プロトコルは以下の通りである。

1. 各参加者は自身の持つシェアから $[\alpha] := [x]-[a]$ と $[\beta] := [y]-[b]$ を計算する。
2. 各参加者は自身の $[\alpha]$ および $[\beta]$ のシェアを他参加者に共有し、α と β を復元する。
3. 各参加者は自身の持つシェアから $[w] := [c]+\alpha[b]+\beta[a]+\alpha\beta$ を計算して出力する。

正当性は、$w = c+\alpha b+\beta a+\alpha\beta = ab+(x-a)b+(y-b)a+(x-a)(y-b) = ab+bx-ab+ay-ab+xy-bx-ay+ab = xy$ により確認できる。

Beaver triple は乗算プロトコルの入力とは無関係であるため、入力を受け取る前に計算して蓄えておくことができる。これにより、実際の入力を受け取った後の処理時間を小さくすることができる。Beaver triple の事前計算の方法はいくつも提案されており、例えば MPC プロトコルにより計算する方法 [33] や、信頼する第三者が作成する方法 [34] がある。

Beaver triple を使う方式は ZenmuTech 社と産総研の秘密計算ソフトウェア QueryAhead [35] などに実装され実用化されている [36]。

3.3.3 マルチパーティ計算–Garbled Circuit ベース–

■Garbled Circuit の概要 Garbled Circuit（GC）は、1986 年に Yao [37] により提案された 2 者間のマルチパーティ計算（Multi-Party Computation，MPC）のためのフレームワークである。この方式では、2 者が協調計算したい関数を論理回路と

して表現した後、その回路の各ゲートに対応する真理値表に対し、乱数を鍵として暗号化することで、Garbled Truth Table（GTT）と呼ばれる「暗号化された真理値表」に相当する暗号文リストを作成する。この GTT に含まれる暗号文のうち、2 者の入力に対応する暗号文のみが正しく復号されるように乱数の交換を行うことで、関数値のみが復元される仕組みを実現する。

GC に基づく MPC は、第 3.3.2 章に記載の秘密分散ベースの MPC とは異なり、乗算回数などに応じた膨大な通信回数を必要とはしないが、上記の通り、処理全体をすべての入力に対する真理値表とみなしたうえで、真理値表全体の暗号文を送受信しなければならないため、通信量が非常に大きくなることが知られている。これらの特徴から、GC に基づく MPC は秘密分散ベースの MPC と比較すると遅延の大きいネットワーク環境における小規模な処理に適していると考えられる。GC と秘密分散ベースの MPC を組み合わせて用いることもあり、特に 3 者以上の多人数の MPC において、定数回の通信による方式を実現するときなどに利用される [22, 23]。

また、GC の概念は 1980 年代から知られていたものの、厳密な安全性の取り扱いについては長期にわたり十分な整備がなされていなかったが、Bellare らなどの成果 [38] により、理論的に厳密な取り扱いが可能となっている。

■代表的な方式　ここでは、GC に基づく 2 者計算の代表的な構成方法をいくつか紹介する。

以下、プロトコルに参加する 2 人の参加者を P_1 と P_2 とし、P_1 および P_2 が持つ入力をそれぞれ $x \in \{0,1\}^n$, $y \in \{0,1\}^n$ として、お互いの入力を秘匿したまま関数 $\mathcal{F}(x,y)$ を計算することを目標とする。関数 $\mathcal{F}$ の論理回路による表現を C と表記する。また、κ をセキュリティパラメータとし、$H : \{0,1\}^* \leftarrow \{0,1\}^\kappa$ を暗号学的ハッシュ関数とする。

方式 1: Yao の Garbled Circuit [37]

代表的な方式の一つとして、Yao [37] による GC の構成法を紹介する。

以下の手順で回路 C に対する GC を構成することができる。

ラベル生成:　回路 C の各ワイヤ w_i および $b \in \{0,1\}$ について、ランダムにラベル $w_i^b = (k_i^b \in \{0,1\}^\kappa, p_i^b \in \{0,1\})$ を生成する。ただし、$p_i^b = 1 - p_i^{1-b}$ とする。w_i^0 および w_i^1 は、このワイヤにおける値 0 と 1 に対応するラベルである。

GTT の生成:　C に含まれる各論理ゲート G_i について、入力ワイヤに近いワイヤ

から順に以下の処理を行い、GTT を作成する。

- G_iを関数 $g_i : w_c = g_i(w_a, w_b)$ のゲートとする。入力 w_a に対応するラベルを $w_a^0 = (k_a^0, p_a^0), w_a^1 = (k_a^1, p_a^1)$、$w_b$ に対応するラベルを $w_b^0 = (k_b^0, p_b^0), w_b^1 = (k_b^1, p_b^1)$、出力 w_c に対応するラベルが $w_c^0 = (k_c^0, p_c^0), w_c^1 = (k_c^1, p_c^1)$ とする。
- G_i への入力 $v_a, v_b \in \{0, 1\}$ の考えられる組み合わせ 4 通り（2×2）それぞれについて、$e_{v_a,v_b} = H(k_a^{v_a}||k_b^{v_b}||i) \oplus w_c^{g_i(v_a,v_b)}$ を計算する。
- 前ステップで計算した $\{e_{v_a,v_b}\}_{v_a,v_b\in\{0,1\}}$ を、$(p_a^{v_a}, p_b^{v_b})$ の順でソートする。

出力復号用テーブルの作成:　各ゲートの出力ワイヤ w_i（ラベル $w_i^0 = (k_i^0, p_i^0)$ および $w_i^1 = (k_i^1, p_i^1)$ に対応）およびワイヤが取りうる値 $v \in \{0, 1\}$ について、$e_v = H(k_i^v||\text{“}out\text{”}||j) \oplus v$ を計算する。ただし、この計算においては H の最下位ビット（LSB）のみを用いる。得られた出力 $\{e_v\}$ を p_i^v の順でソートする。

GC に基づく MPC においては、上記の手順を P_1 が行う。この計算を行う参加者のことを Garbler とも呼ぶ。

上記の手順により生成した出力復号用テーブルを含む GC を、P_1 から P_2 へ送信する。同時に、各入力ワイヤについて、P_1 自身の入力に対応するラベルも P_2 へ送信する。

GC の “復号”には、P_2 の入力に対応するラベルを P_1 から P_2 の入力を知ることなく P_2 へ送信する必要がある。このような機能は、紛失通信 （Oblivious Transfer，OT）[39] によって実現することができる。各 P_2 の入力ワイヤ w_b について、P_1 は 0, 1 それぞれの入力に対するラベル w_b^0, w_b^1 を入力し、P_2 は選択ビット c を入力することで、P_1 から P_2 へ w_b^c のみを送信することができる。

以降は P_2 が受け取った入力を基に GC の評価を行う。この手順を行う参加者を Evaluator とも呼ぶ。各ゲートに対応する GTT $T = (e_{0,0}, e_{0,1}, e_{1,0}, e_{1,1})$ および入力ラベル $w_a = (k_a, p_a), w_b = (k_b, p_b)$ を用いて、P_2 は $w_c = H(k_a||k_b||i) \oplus e_{p_a,p_b}$ を計算する。

すべてのゲートを評価し終えたら、最終的な出力を復号するために、第 2 キーに “out” を用いて、計算の平文結果となる最終出力を得ることができる。P_2 は得られた出力を P_1 に返却することで、両者が結果を得ることができる。

方式 2: Half-Gates [40]

GC においては、GTT のサイズを削減するためにいくつかのテクニックが知られている。Yao の方式では 2 入力 1 出力のゲート 1 つあたりに 4 つの暗号文を必要

としたが、XOR ゲートの暗号文数を 0 に削減する Free-XOR と呼ばれるテクニック [41] や、一般のゲートにおける暗号文数を 3 あるいは 2 へ削減する Garbled Row Reduction（GRR）テクニック [42, 43] などが知られる。

これらのテクニックを組み合わせた方式で効率が良い方式の一つとして、Zahur らによって提案された Half-gates [40] と呼ばれる方式を説明する。この方式では、XOR ゲートの GTT を必要とせず、AND ゲートの GTT を暗号文 2 つで構成することが可能である。AND ゲートと XOR ゲートの組み合わせで、任意の回路を表現可能であることに注意する。

Half-gates のアイデアのポイントは、AND ゲートを 2 つの “half gate” の XOR として表現することである。このそれぞれの half gate は、入力の 1 つが参加者のいずれかに既知の AND ゲートであり、暗号文 2 つで GTT を構成することができる。この暗号文 2 つの GTT を GRR テクニックによって暗号文 1 つに圧縮し、2 つの half gate を接続する XOR を Free-XOR テクニックによって削減することで暗号文 2 つによる AND ゲートの GC を達成している。

2 つの half gate はそれぞれ、“generator half gate” と “evaluator half gate” と呼ばれる。以下でそれぞれについて説明する。

• Generator half gate

入力ワイヤを a と b、出力ワイヤを c とする AND ゲートを考える。Generator half gate は $v_c = v_a \wedge v_b$ のような回路で、v_a が generator (P_1) に対して何らかの形で既知であり、evaluator (P_2) はいずれの入力も未知の回路である。このとき、$v_a = 0$ であれば $v_c = 0$ であり、$v_a = 1$ であれば $v_c = v_b$ であることに注意する。

いま、k_a^0, k_b^0, k_c^0 をワイヤ a, b, c それぞれの入力 0 に対応するラベルとする。このとき、generator half gate の GTT は以下のように表される。

$$H(k_b^0) \oplus k_c^0$$
$$H(k_b^1) \oplus k_c^0 \oplus v_a \cdot R$$

ただし、$k_b^1 = k_b^0 \oplus R$ であり、R はランダムなグローバルオフセットである。Free-XOR を用いる際はこのように、0 と 1 それぞれに対応する 2 つのラベルを $(X, X \oplus R)$ のように差分を固定した形で設定する。

Evaluator (P_2) はこの half gate を評価するために、自身の入力 b に対応するラベル（k_b^0 あるいは k_b^1）を用いてそのハッシュ値を計算する。もしワイヤ b の値が 0 の場合、evaluator は k_b^0 を持ち、$H(k_b^0)$ を計算して k_c^0 を得ることができる。一方、もし b の値が 1 の場合は、$k_b^1 = k_b^0 \oplus R$ によって $H(k_b^1)$ を計算して $k_c^0 \oplus v_a \cdot R$ を得ることができる。ちなみに、$v_a = 0$ のとき k_c^0 となり、$v_a = 1$ のとき $k_c^0 \oplus R$ となる。

• Evaluator half gate

Evaluator half gate は、$v_c = v_a \wedge v_b$ のような形で、v_a が evaluator (P_2) にとって既知であり、generator (P_1) はいずれの入力も未知の回路である。このとき、evaluator は自分の知るワイヤ a の入力に応じて異なる振る舞いをすることができる。Evaluator half gate の GTT は以下のように実現される。

$$H(k_a^0) \oplus k_c^0$$
$$H(k_a^1) \oplus k_c^0 \oplus k_b^0$$

ただし、generator half gate と同様、$k_a^1 = k_a^0 \oplus R$ であり、R はグローバルなオフセットである。ワイヤ a の値が 0 の場合、$H(k_a^0)$ を計算し、k_c^0 を得ることができる。一方、a の値が 1 の場合は、$k_c^0 \oplus k_b^0$ を得ることができる。最終的に、これを b に対応するラベルと XOR を取ることで、b の値が 0 の場合、すなわち evaluator が k_b^0 を持っている場合は k_c^0 を、b の値が 1 の場合、すなわち、evaluator が k_b^1 を持っている場合は $k_c^1 = k_c^0 \oplus R$ を得ることができる。

• Generator half gate/Evaluator half gate の結合

上記の 2 種類の half gate を用いて $v_c = v_a \wedge v_b$ のゲートを GC で評価するために、generator は GC の生成時にランダムなビット r を生成し、これを用いて AND ゲートを以下のように変形する。

$$\begin{aligned} v_c &= v_a \wedge (r + r + b) \\ &= (v_a \wedge r) \oplus (v_a \wedge (r \oplus v_b)) \end{aligned}$$

$(v_a \wedge r)$ は generator half gate を用いて garbling することができ、$(v_a \wedge (r \oplus v_b))$ は point-and-permute [22] を用いると $r \oplus v_b$ を evaluator に伝えることができるため、evaluator half gate を用いて garbling することができる。XOR ゲートについては上記の通り、Free-XOR テクニックを用いると GTT が不要であるため、GRR で圧縮した generator half gate の GTT と evaluator half gate の GTT の組が、所望の AND ゲート $v_a \wedge v_b$ に対する GTT となる。

3.3.4 ゼロ知識証明

■ゼロ知識証明の特徴　ゼロ知識証明とは、ある命題 X に対して、証明者と検証者の間で情報をやりとりし、証明者が検証者に命題 X が正しいことを確信させ、かつ命題が正しいこと以外の情報を検証者に与えないプロトコルである。次に述べる完全性、健全性、ゼロ知識性の性質をもつ。

完全性：　証明者の命題が真ならば、検証者は真であることが必ずわかること。

健全性： 証明者の命題が偽ならば、検証者はかなり高い確率で、偽であることを見抜けること。

ゼロ知識性： あらゆる場合で検証者が証明者から何らかの知識を得ようとしても、証明者の命題が真であること以上の知識は得られないこと。

さらに、ゼロ知識証明の中でも zk-SNARKs 方式（Zero-Knowledge Succinct Non-Interactive Argument of Knowledge）は、命題が正しいことの証拠の保有を証明するもので、名前に由来する以下の特徴を持つ。

簡潔 $Succinct$**：** 知識証明のサイズが命題のサイズに依存せず一定で、十分に短い

非対話 $NonInteractive$**：** 検証者は証明者からの一度の知識証明の送付で検証可能

アーギュメント $Argument$**：** 検証者は計算能力に限りある（多項式時間計算力）証明者からは守られる

知識 $Knowledge$**：** 証明者は証拠なしでは知識証明の生成は不可能

ゼロ知識証明で扱える命題は、その証拠を使って多項式時間アルゴリズムで真偽判定できる「クラス NP 問題」であり、関数の解や充足可能問題なども含み幅広い。

■ゼロ知識証明の歴史 ゼロ知識証明の歴史は古く、1985 年に Goldwasser らにより、証明者と検証者間で対話を繰り返し、知識を証明するプロトコルとして定式化された [44]。この対話型の改良として、信頼できる第三者機関が、共通参照情報（Common Reference String, CRS）を事前に生成し、配布することで、非対話型が達成される [45]。また健全性について、離散対数問題をベースとしたシュノアプロトコル [46] は、証明者の計算能力を限定することで健全性を保証する方式で、計算困難な数学的問題を根拠とする方式の原型といえる。

クラス NP 問題に、一方向性関数の存在を仮定すると、非対話型ゼロ知識証明が適用できることは保証されていたが、実用的な手法は 2013 年の Gennaro らによる zk-SNARK である [47]。これを一般化した zk-SNARKs 方式の中でも Pinocchio [48] は、暗号資産の取引システム上で実用化されている。また Pinocchio の効率化が Groth により定式されている [49]。さらに、zk-STARKs [50] と BulletProofs [51] は、証明サイズや計算量は大きくなるが、事前準備を必要としないことを特徴とする。

■代表的なゼロ知識証明方式　Pinocchio アルゴリズムの概要を説明する。はじめに、命題「多項式 $\mathcal{F}$ の解の存在」をゼロ知識証明する場合を例に、全体の流れを示す。

1. $\mathcal{F}$ を加算と積算演算ゲートを持つ算術回路に変換する。
2. 具体的な手続きに従って、回路を 2 次算術プログラム（QAP）で表現する。ここで QAP は 3 つの多項式の集合 V, W, Y と目的多項式 $t(x)$ で構成される。回路を満たす割り当て $\{c_i\}$ がある時、V, W, Y と $\{c_i\}$ から多項式 $p(x)$ が定まり、後述の QAP 定理 [47] より「$\mathcal{F}$ の解の存在」と「$h(x)t(x) = p(x)$ を満たす多項式 $h(x)$ の存在」が同値になる。
3. 証明者は、証拠から $h(x)$ を計算し、$h(x)$ を暗号化した知識証明 π を作り、検証者に渡す。
4. 検証者は、知識証明 π から、$h(x)t(x) = p(x)$ の成立を、Pairing 写像を用いて検証する。

以上のプロセスを、非対話型にするために CRS と呼ばれる乱数の共有が必要である。第 2 項にある QAP 定理を以下に示す。

QAP **定理** [47]:　$\mathcal{F}$ を n 個の元を入力し n' 個の元を出力する関数とし、$N = n+n'$ とする。この時、3 つの多項式集合 $V = \{v_k(x)\}, W = \{w_k(x)\}, Y = \{y_k(x)\}_{k=1,\cdots,m}$ と目的多項式 $t(x)$ が存在し (これを $\mathcal{F}$ の QAP と呼ぶ)、次が成立する。入出力値 $(c_1, \cdots, c_N)$ が $\mathcal{F}$ から得られる回路の正しい割り当て（値）とすると、係数 $(c_{N+1}, \cdots, c_m)$ が存在し、$t(x)$ が $p(x)$ を割り切る。この時、係数 $(c_{N+1}, \cdots, c_m)$ を証拠と呼ぶ。ここで $p(x)$ と $t(x)$ は以下の式となる。

$$p(x) = (\sum_{k=1}^{m} c_k v_k(x))(\sum_{k=1}^{m} c_k w_k(x)) - \sum_{k=1}^{m} c_k y_k(x),$$
$$t(x) = \prod_{g:\text{積算演算ゲート}} (x - r_g)$$

具体例として、$\mathcal{F}(C_1, C_2, C_3, C_4) = (C_1 \times C_2) \times (C_3 + C_4) = C_6$ を当てはめると、$\mathcal{F}$ の QAP は目的多項式 $t(x) = (x - r_5)(x - r_6)$ と $V = \{v_k(x)\}, W = \{w_k(x)\}, Y = \{y_k(x)\}_{k=1,\cdots,6}$ で構成される。V の初項は $v_1(x) = (x - r_5)(x - r_6 + 1)/(r_6 - r_5)$ であり、$i \geq 2$ の $v_i(x)$ と W,Y は順次計算できるが、残りの項の記載は省略する。そして、$\mathcal{F}$ を満たす解 $x = (c_1, c_2, c_3, c_4, c_6)$ と途中ゲートの入出力制約をみたす c_5 を使用して、$p(x) = \{c_1 v_1(x) + c_2 v_2(x) + c_3 v_3(x)\}\{c_4 w_4(x) + c_5 w_5(x)\} - \{c_5 y_5(x) + c_6 y_6(x)\}$ を計算する。$p(x)$ を展開すると、「$(c_1 \times c_2) \times (c_3 + c_4) = c_6$

の成立」と「$t(x)$ が $p(x)$ を割り切る」が同値であることが分かる。

定式化のため写像を 2 つ用意する。1 つ目は巡回群 G とその生成元 g による、暗号化写像 $E(s) := g^s$ であり、単射かつ $E(s+t) = E(s)E(t)$ を満たす。値 g^s と g からは s が計算困難な離散対数困難性を持つ。2 つ目は G 上の Pairing 写像 e で、双線形性 $e(g^a, g^b) = e(g, g)^{ab}$ を持つ。Pinocchio は（CRSGen,Proof, Verify）の 3 つのアルゴリズムからなる。

CRSGen$(1^\lambda, \mathcal{F})$: 信頼できる第三者により CRS を生成するアルゴリズム。

証明したい命題 $\mathcal{F}$ を、算術回路変換を通して QAP 表現する。QAP がサイズ m、次数 d、$V = \{v_k(x)\}, W = \{w_k(x)\}, Y = \{y_k(x)\}_{k=1,\cdots,m}, t(x)$ で構成される時、$[N] = \{1, \cdots, N\}, I_{mid} = \{N, \cdots, m\}, [d] = \{1, \cdots, d\}$ と置く。$s, \alpha, \beta_v, \beta_w, \beta_y, \gamma$ をランダムに選び、$CRS = (EK, VK)$ を生成する。証明者に EK を、検証者に VK を送付する。

$$
\begin{aligned}
EK = \{&\{E(s^i), E(\alpha s^i)\}_{i\in[d]}, \{E(v_k(s)), E(\alpha v_k(s)), E(\beta_v v_k(s))\}_{k\in I_{mid}},\\
&\{E(w_k(s)), E(y_k(s)), E(\alpha w_k(s)), E(\alpha y_k(s), E(\beta_w w_k(s)),\\
&E(\beta_y y_k(s))\}_{k\in[N]}\}\\
VK = \{&E(1), E(\alpha), E(\gamma), E(\beta_v\gamma), E(\beta_w\gamma), E(\beta_y\gamma), E(t(s)), E(v_k(s)_{k\in[N]}\}
\end{aligned}
$$

Proof(EK, u): 証明者により知識証明 π を生成するアルゴリズム。

証明者は、入力 u から $y = \mathcal{F}(u)$ を計算し、回路が満たす $\{c_k\}_{k\in[m]}$ を得る。これから

$$
\begin{aligned}
p(x) &= (\sum_{k=1}^{m} c_k v_k(x))(\sum_{k=1}^{m} c_k w_k(x)) \ - \sum_{k=1}^{m} c_k y_k(x)\\
h(x) &= p(x)/t(x)
\end{aligned}
$$

と多項式 $h(x)$ を得る。次に、この $h(x)$ の暗号化を行う。$h(x)$ の係数 h_i から

$$E(h(s)) = \prod_{i\in[d]} E(s^i)^{h_i}$$

を計算する。さらに、秘匿したい入出力値と証拠 $\{c_k\}_{k\in I_{mid}}$ も使い、$E(v_{mid}(s)) = \prod_{k\in I_{mid}} E(v_k(s))^{c_k}$ や $E(w(s))$ なども同様の計算を行い、合わせて、知識証明 π を構成する。

$$
\begin{aligned}
\pi = \{&E(h(s)), E(v_{mid}(s)), E(w(s)), E(y(s)), E(\alpha h(s)),\\
&E(\alpha v_{mid}(s)), E(\alpha w(s)), E(\alpha y(s))E(\beta_v v(s))E(\beta_w w(s))E(\beta_y y(s))\}
\end{aligned}
$$

π に、y と u の中で秘匿しない入出力値に対応する $\{c_k\}$ を加えて、検証者に送る。

しかし、上記の定式化では、π のゼロ知識性が満たされていない。そのため、一様乱数の組と区別できないように、$v'_{mid}(x) = v_{mid}(x) + \delta t(x)$ と、暗号化前の各多項式に乱数 δ を加えてゼロ知識性を達成する。これに合わせて $p(x)$ と $h(x)$ も変更される。

Verify(VK, u, y, π): 検証者により証明者の知識 π を検証するアルゴリズム。

次の 2 種類のチェックのすべてが通った時に、検証成立となる。

1. 等式 $p(s) \stackrel{?}{=} h(s)t(s)$ の成立チェック
   VK と回路の秘匿しない入出力値 $\{c_k\}$ を使い、$E(v_{i/o}(s)) = \prod_{k\in[N]} E(v_k(s))^{c_k}$ を計算する。そして、

   $$e(E(v_{i/o}(s))E(v_{mid}(s)), E(w(s))/e(E(y(s), E(1)) \stackrel{?}{=} e(E(h(s)), E(t(s))$$

   の両辺を比較し一致する時、e の双線形性より $(v_{i/o}(s) + v_{mid}(s))w(s) - y(s)) = h(s)t(s)$ が成立する。
2. 線形結合チェック

   $$e(E(\alpha h(s)), E(1)) \stackrel{?}{=} e(E(h(s)), E(\alpha)),$$
   $$e(E(\beta_v v(s)), E(\gamma)) \stackrel{?}{=} e(E(v(s)), E(\beta_v \gamma))$$

   を含む、α に関する 8 ペアと β に関する 3 ペアの Pairing 写像を用いたチェックを行う。

■ゼロ知識証明の安全性など アルゴリズムを支える根拠と安全性を述べる。Proof の知識証明 π は回路のサイズに関わらず巡回群 G の 8 つの元で構成され、サイズは一定となる。さらに、多項式自身でなく、ランダム s での評価値 $h(s)$ が知識証明に使用されている。これは Schwartz-Zipple 補題により、異なる多項式をランダムな点で評価すると高い確率で異なる値となることが示されており、このようなランダム点での評価値を用いることができる。さらに、多項式の秘匿評価手法により CRS の $E(s^i)$ を使うことで s 自身を知らずとも $E(h(s))$ が計算可能である。Verify において、線形結合チェックが加えられている。これは、証明者は $h(x)$ や $v_{mid}(x)$ を、証拠（の係数）を持たずとも、等式が成立する任意の多項式として作ることができる。この不正な攻撃に対応するために、検証者は $h(s)$ が $\{s^i\}_{i\in[d]}$ の線形結合から作られていることの確認が必要である。Knowledge of Coefficient（KC）に基づくと、$e(E(\alpha h(s)), E(1)) = e(E(h(s)), E(\alpha))$ が成立する時に、この

線形結合性が確認できる。そして、検証は一度の知識証明の送付で達成しており、非対話型である。

3.3.5 Oblivious Random Access Machine（ORAM）

■ORAM の特徴　外部サーバにデータを預ける際、暗号化していればデータの内容は漏洩しないが、あるデータへのアクセスの偏りが発生した場合、サーバはそのアクセスの偏りから該当データの情報を推測できてしまう可能性がある。ORAM では、暗号化データに対して読み込み書き込みを行うたびに、格納位置をシャッフルするとともに再暗号化することで各アクセス間の関係を秘匿し、サーバに対してアクセスパターンを秘匿できる。クライアントは格納位置に関する情報を保持することによって、目的の暗号化データの格納位置を知ることができる。

同様にサーバに対するプライバシ保護を目的とする高機能暗号技術である第 3.3.6 章に記載の Private Information Retrieval（PIR）と第 3.3.7 章に記載の検索可能暗号との違いは以下の通りである。

- サーバへのクエリ内容：検索可能暗号では格納したデータがあるキーワードを含んでいるかどうかをクエリするのに対し、ORAM と PIR ではクライアントがアクセスしたいデータそのものについてクエリする。
- 格納したデータの暗号文：検索可能暗号と PIR では基本的にサーバに格納されたデータやその暗号文は（データの更新処理などが行われない限り）クエリによって変わることが無いのに対し、ORAM では毎回のクエリごとにデータの更新と暗号文の再暗号化、格納位置のシャッフルが行われる。
- 秘匿できる情報：検索可能暗号では検索に該当するデータの頻度などの情報は秘匿しないのに対し、ORAM と PIR では頻度を含めたアクセスパターンを秘匿する。
- クライアントとサーバの処理：検索可能暗号と PIR ではクエリに応じてサーバが暗号化データベースに対して検索処理を行うのに対し、ORAM では暗号化データベースの一部をサーバから受け取りクライアントが検索処理（およびにシャッフルと再暗号化）を行う。

このように、検索可能暗号はアクセスパターンの秘匿までは必要のない状況、PIR は通信量とサーバの計算量が増えてもクライアントの計算量を抑えたい状況、ORAM はある程度クライアントは計算量を要するが通信量を抑えたい状況で使い分けることが望ましい。

■ORAM の歴史と方式の比較　ORAM の概念は、Goldreich [52] によって初めて提案された。元々はソフトウェア保護の文脈で CPU によるメモリアクセスパターンの秘匿を目的として考案されたが、その後、CPU をクライアント、メモリをサーバに置き換えることにより、暗号化データベースのアクセスパターン秘匿への適用が主な応用先となった。Goldreich と Ostrovsky による初期の方式 [52, 53, 54] では、データブロック数 N に対してアクセスごとの償却通信計算量は $\mathcal{O}(\log^4 N)$ であったが、最悪通信計算量は $\Omega(\log N)$ であった。

その後、Shi ら [55] や Kushilevitz ら [56] によって、木構造を用いてデータを管理することによって最悪通信計算量がそれぞれ $\mathcal{O}(\log^3 N/\log\log N)$ と $\mathcal{O}(\log^4 N)$ となる方式が提案された。通信計算量の観点で効率の良い ORAM 方式として、Stefanov ら [57] による Path ORAM が知られており、データブロック数 N に対して通信計算量は $\mathcal{O}(\log^3 N)$ でよい。ただし、Path ORAM では、クライアントが position map と呼ばれるテーブルとスタッシュと呼ばれるローカルなメモリに情報を格納するため、クライアントの記憶容量は N に依存したサイズが必要である。

近年はクライアントの記憶容量や通信計算量を漸近的に減らす研究が進んでいる。Patel ら [58] は $\mathcal{O}(\log N)$ ビットのクライアント記憶容量かつ $\mathcal{O}(\log^2 N \log\log N)$ の償却通信計算量の方式を提案した。Asharov ら [59] はこれを改良し、$\mathcal{O}(\log^2 N)$ の償却通信計算量の方式を提案した。さらに、Asharov ら [60] により $\mathcal{O}(\log N)$ ビットのクライアント記憶容量かつ $\mathcal{O}(\log^2 N)$ の最悪通信計算量の方式が提案された。この方式は漸近的には効率的であるものの、オーダー表記に隠れた定数倍が大きいため実用は現実的ではない。

ORAM の拡張として、サーバを複数台に分けてデータを格納することによって、単一サーバの ORAM より効率が良い分散 ORAM [61] が知られている。また、通常の ORAM ではクライアントは単一だが、複数のクライアントを含むことができる並行 ORAM [62] も知られている。

■代表的な方式のアルゴリズム　具体的な ORAM の実現方法として、木構造を用いた Path ORAM [57] を紹介する。

- サーバのセットアップ： 高さ L の二分木を生成する。各ノードにそれぞれ Z 個のブロックを割り当て、暗号化データを格納する。もしあるノードに入るブロックが Z 個未満になる場合は、真のブロックとは識別できないダミーのブロックを格納することで常に Z 個のブロックで満たされるようにする。
- クライアントのセットアップ： それぞれのブロックと葉ノードを紐づけた

position map と呼ばれるテーブルを生成する。position map は、ブロックが根ノードから紐づけられた葉ノードの経路上のいずれかにあることを保証し、あるブロック a が左から x 番目の葉ノードと紐づけられているならば、a は根ノードから x 番目の葉ノードへの経路上のいずれかにあることになる。初期状態においてブロックは全て空の状態で、各ブロックはそれぞれランダムに選ばれた葉ノードと紐付けられて、その葉ノードの経路上に配置される。また、いくつかのブロックを保存できるスタッシュと呼ばれるメモリを用意する。

- アクセス：
  1. 処理 $op \in \{read, write\}$、ブロック a、データ d に対して、クライアントは position map からブロック a に対応した葉ノード x を選択し、x をサーバに送信する。その後 position map の a に対応した葉ノードをランダムな葉ノード x' に上書きする。
  2. サーバは受け取った x から根ノードへの経路上にあるブロックを全てクライアントに送る。
  3. クライアントはサーバから受け取ったブロックをスタッシュに保存し、スタッシュにある暗号化データを全て復号する。a を用い参照したいデータを探し出し、$op = write$ ならば内容を d に更新する。
  4. クライアントは全てのデータを再暗号化し、position map と x の最小共通祖先ノードに格納する。参照した a は、x と x' の最小共通祖先ノードに格納される。この時ノードに空きが出来た場合、ダミーのブロックを格納する。入るべきノードがすべて埋まってしまいスタッシュに残ったブロックは、そのままスタッシュに一時保存する。

3.3.6 Private Information Retrieval（PIR）

■PIR の特徴および他の関連技術との差異　クライアントがサーバ上のデータを検索する際、クライアントは検索のためのクエリをサーバに送信し、そのレスポンスとして、クライアントはサーバから検索結果を受け取る。これら一連の作業において検索履歴や検索結果、サーバに保存されたデータ自体の秘匿性をどのように考えるかによって様々な方式が提案されている。秘匿情報検索（Private Information Retrieval，PIR）は、サーバ上のデータは、暗号化されず平文のまま保存されている状況で、クライアントがどのデータにアクセスしたか、サーバに秘匿することを目的とする。サーバに預けたデータをクエリする際に、そのアクセス情報を秘

匿する暗号技術としては、PIR の他に第 3.3.5 章に記載された Oblivious Random Access Machine（ORAM）や第 3.3.7 章に記載された検索可能暗号がある。これらの暗号技術と PIR の具体的な相違点は以下の通りである。

- サーバが保持する情報：PIR ではデータを暗号化しない状態で保持する一方、ORAM や検索可能暗号ではデータを暗号化した状態で保持する。言い換えれば、PIR では元々サーバが管理するデータを、ORAM 及び検索可能暗号ではクライアントが管理するデータを扱うことを想定している。
- 格納した暗号化データの取り扱い： PIR では基本的にサーバに格納されたデータやその暗号文はクエリによって変わることが無いのに対し、ORAM や検索可能暗号では毎回のクエリごとにデータの更新と暗号文の再暗号化等が行われる。特に ORAM では格納位置のシャッフルも行われる。
- サーバへのクエリ内容：ORAM と PIR ではクライアントがアクセスしたいデータそのものについてクエリするが、検索可能暗号では格納したデータがあるキーワードを含んでいるかどうかをクエリする。
- サーバに対して秘匿できる情報：いずれの技術もクエリ内容を秘匿する。特に、クエリに対してどのデータが返されたかに関する情報を秘匿する。ORAM と PIR では頻度を含めた全ての情報を秘匿するが、検索可能暗号では検索に該当するデータの頻度などの一部の情報が漏洩することを許容する。

PIR の具体的な設定は以下の通りである。クライアント U と t 個のサーバ $\mathsf{DB}_1, \mathsf{DB}_2, \ldots, \mathsf{DB}_t$ がある。各サーバは同じデータ $x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ をもっており、結託はしないものとする。U が x_i を検索するために各サーバにクエリを送信し、サーバがレスポンスを返す。必要に応じてこれを繰り返すことでクライアントは x_i を得る。安全性要件として、クライアントが送信するクエリは、クライアントが検索しようとしているデータのインデックス i をサーバに漏らさないことを要請する。この安全性要件に対して、PIR では、情報理論的安全性、計算量的安全性を満たすプロトコルがそれぞれ提案されている。

このような設定のもとで自明なプロトコルは、$t = 1$ として、クライアントはクエリを生成せず、サーバが全てのデータ $x_1, x_2, \ldots, x_n$ をサーバに送ることである。従って、PIR を考えるとき、この自明な方式よりは効率的な手法であることが求められる。

■PIR の歴史　PIR は Chor らによって最初に提案された [63, 64]。ここで提案された方式は情報理論的安全性を満たす。彼らは PIR を情報理論的に達成するために

は（自明な方式を除いて）サーバ 1 台では不可能であることを示し、結託しない 2 台のサーバが同じデータのコピーをもっていれば PIR が実現できることを示した。この結果から、計算量的安全性のもとで、サーバの数を 1 台まで減らせるかが重要な課題となったが、これは Kushilevitz ら [65] によって肯定的に解決された。この方式でのサーバの計算時間は $\Omega(n)$ であり、平方剰余を用いていた。近年の方式では完全準同形暗号を用いて、さらに前処理を含めた複数ラウンドのプロトコルで劣線形時間で実行できるプロトコルが知られている [66, 67]。また符号に基づく PIR も Holzbaur らによって提案されている [68, 69]。

情報理論的な観点からは、t 個のサーバからダウンロードするデータ 1 ビットに対して、検索可能なビット数（通信レート）の最大値（通信容量）に興味がある。例えば、PIR の特徴および他の関連技術との差異の項で示した自明な方式のレートは $1/n$ である。Sun らはこの最大値が $1/\sum_{u=0}^{k-1} n^{-u}$ であることを示し、実際にこの容量を与えるプロトコルを示した [70]。この研究に端を発して、情報理論的な PIR の研究は符号理論などの情報理論的な技術と関連して活発になってきた。例えば、t 台のサーバに同一のコピーを置かずに、サーバに置くデータサイズの削減を目指した Coded PIR [71] は、Katz らによる Locally decodable code [72] を用いて構成される。また、複数のサーバからの応答のうち、いくつかが破損したり改竄されている場合にも正しく情報が得られるロバスト PIR に関する研究も行われている。ロバスト PIR は Beimel と Stahl によって提案され [73]、Belekamp–Welch アルゴリズムによる効率化が Kurosawa によって提案されている [74]。

■代表的方式　本節では、情報理論的に安全な方式として [64] を、計算量的に安全な方式として単一サーバの PIR として初の方式である [65] を紹介する。

情報理論的に安全な 2 サーバ PIR [64]

[64] では、結託しない 2 台のサーバ $\mathsf{DB}_1, \mathsf{DB}_2$ を準備し、それぞれが n ビットのデータ $x := (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ を保持しているものとする。ここでクライアントが x_i の値を知りたいとするとき、以下のようなプロトコルを実行する。

自然数 n に対して、$[n] := \{1, 2 \ldots, n\}$ と定める。

1) $2^{[n]}$ 上の一様分布から集合 $S \subseteq [n]$ を選ぶ。
2) $S_1 := S$ として、$s \in S$ であるなら、$S_2 := S \backslash \{i\}$、$s \notin S$ であるなら $S_2 := S \cup \{i\}$ として集合 S_1, S_2 をそれぞれ $\mathsf{DB}_1, \mathsf{DB}_2$ に送る。
3) $i = 1, 2$ に対して、集合 S_i を受け取った DB_i は $r_i = \oplus_{j \in S_i} x_j$ をクライアントに返送する。ここで $\oplus$ は XOR を表す。

4) クライアントは $r_1 \oplus r_2$ として x_i を求める。

プロトコルの健全性は方式からほぼ明らかである。また、集合 S が $2^{[n]}$ から一様分布に従って選ばれているため、i が S に属す確率は $1/2$ である。従って、i が S_1 に属す確率も $1/2$ であり、S_2 に属する確率も $1/2$ となる。このことから、i の情報は $\mathsf{DB}_1, \mathsf{DB}_2$ のどちらにも情報理論的な意味で漏洩しない。

計算量的に安全な単一サーバ PIR [65]

情報理論的な設定の下で、サーバの数を 1 つにすることができないが [64]、計算量的安全性のもとではサーバを 1 台にすることができる。以下に、Kushilevitz ら [65] によって初めて提案された単一サーバによる PIR プロトコルを示す。

本方式は一人のクライアントとサーバ DB のみで構成され、サーバには s 行 t 列の二値行列 $x \in \{0,1\}^{s\times t}$ が格納されているものとする。クライアントが行列 M の第 a, b 成分、$M_{a,b}$ $(1 \le a \le s, 1 \le b \le t)$ をの値を知りたいとするとき、以下のようなプロトコルを実行する。

1) 自然数 k に対して、クライアント U は $k/2$ ビット素数をランダムに 2 つ選び、その積を N とする。
2) クライアント U は t 個の自然数 $y_1, y_2, \ldots, y_t \in \{y \in \mathbb{Z}_N^* \mid (\frac{y}{N}) = 1\}$ を、y_b は平方非剰余、それ以外は平方剰余であるように一様ランダムに選ぶ。ここで $\left(\frac{\cdot}{\cdot}\right)$ は Jacobi 記号である。
3) 行列 M の第 i, j 成分 $M_{i,j} \in \{0,1\}$, $1 \le i \le s$, $1 \le j \le t$ に対して、サーバ DB は $M_{i,j} = 0$ のときに $w_{i,j} := y_j^2$、$M_{i,j} = 1$ のときに $w_{i,j} := y_j$ として $W := (w_{i,j})_{(i,j)\in[s]\times[t]}$ を作成する。DB はさらに、W から $z_i := \prod_{j=1}^{t} w_{i,j}$, $i = 1, 2, \ldots, s$ を計算する。
4) サーバは $z_1, z_2, \ldots, z_t$ をクライアントに送る。
5) クライアントは z_a を取り出し、z_a が N を法として平方剰余なら $M_{a,b} = 0$、平方非剰余であるなら $M_{a,b} = 1$ とする。

構成法から、$M_{a,b} = 1$ のときに限って z_a は N を法として平方剰余となる。クライアントは N の 2 つの素因数を用いて、任意の $y \in \mathbb{Z}_N^*$ が N を法とした平方（非）剰余であるかを多項式時間で判定できるため、$M_{a,b}$ の値が計算できる。一方で、サーバは N の 2 つの素因数を知らないため、平方剰余仮定により y_j や z_i が平方剰余であるか否かを判定できない。このことから、直観的にはサーバが a, b を特定出来ないことがわかる。

3.3.7 検索可能暗号

検索可能暗号とは、サーバに保存されている（暗号化）文書に対して、クライアントが検索したいキーワード及び暗号化文書の内容を秘匿しながらも、当該キーワードを含む（暗号化）文書のみを効率的に検索できる高機能暗号技術である。特に関連する高機能暗号技術として、第 3.3.5 章に記載された Oblivious Random Access Machine（ORAM）や第 3.3.6 章に記載された Private Information Retrieval（PIR）がある。以下に各技術の違いをまとめる。

- サーバが保持する情報：PIR ではデータを暗号化しない状態で保持する一方、ORAM や検索可能暗号ではデータを暗号化した状態で保持する。言い換えれば、PIR では元々サーバが管理するデータを、ORAM 及び検索可能暗号ではクライアントが管理するデータを扱うことを想定している。
- 格納した暗号化データの取り扱い：PIR では基本的にサーバに格納されたデータやその暗号文はクエリによって変わることが無いのに対し、ORAM や検索可能暗号では毎回のクエリごとにデータの更新と暗号文の再暗号化等が行われる。特に ORAM では格納位置のシャッフルも行われる。
- サーバへのクエリ内容：検索可能暗号では格納したデータがあるキーワードを含んでいるかどうかをクエリするのに対し、PIR と ORAM ではクライアントがアクセスしたいデータそのものについてクエリする。
- サーバに対して秘匿できる情報：いずれの技術もクエリ内容を秘匿する。特に、クエリに対してどのデータが返されたかに関する情報を秘匿する。しかし、検索可能暗号では検索に該当するデータの頻度などの一部の情報は秘匿しないのに対し、PIR と ORAM では頻度を含めた全ての情報を秘匿する。

ORAM は一般に検索可能暗号よりも強い安全性を保証できる代わりに効率性の理論限界が知られており、検索可能暗号ではサーバ自身もサーバに保存された内容を知り得ないという意味で PIR とは異なる。検索可能暗号では、サーバ上の暗号化された文書に対して、実用上問題ないと考えられる程度の多少の情報の漏洩を許しつつ、ORAM の理論限界を超えた効率的な検索機能を達成可能である。

■検索可能暗号の分類と歴史　検索可能暗号は、公開鍵暗号に基づく方式と共通鍵暗号に基づく方式に分類できる。公開鍵ベース方式 [75, 76]（Public-key Encryption with Keyword Search, PEKS）では公開鍵を用いて暗号化が可能であり、任意のクライアントがサーバに（暗号化）文書を格納可能となる。PEKS は Boneh ら [75] によって提案され、Abdalla ら [76] によって任意の匿名 ID ベース暗号方式を効率

的に PEKS に変換できることが示され、基本的な構成枠組みが完成した。以降、当初は考えられていなかった種々の攻撃に対する安全性 [77, 78] や複数キーワードを扱う方式 [79] 等の研究が進められている。一方で、共通鍵ベース方式 [80, 81]（Searchable Symmetric Encryption, SSE）では秘密鍵を知っている特定のクライアントしか（暗号化）文書を格納できないものの、処理が高速であるという特長がある。特に、暗号文一つ一つと検索キーワードの一致を判定するアルゴリズムを考え、そして全ての暗号文にそのアルゴリズムを実行することを想定する PEKS とは異なり、SSE では暗号化データベース全体に対して検索を行うアルゴリズムを考えており、検索処理全体の効率性を意識して設計されている。そのような効率性の観点から、SSE に関する研究が現在の検索可能暗号研究の主流となっている。SSE はその概念自体が Song ら [82] によって提案された後、Curtmola ら [80, 81] によってそのモデルと安全性が定式化され、具体的な方式も提案された。以降、SSE 方式の効率性改善だけでなく、理論限界の解析 [83, 84, 85] や、正しく結果を返さない不正なサーバを検知することで外部プロトコルと組み合わせた時の安全性を保証する方式 [86]、複数キーワードを扱う方式 [87]、一般的な情報検索を扱う方式 [88] 等の様々な研究が続けられているが、最も盛んに研究が進められているのは、暗号化データベースの動的な更新（任意のタイミングでの文書ファイル・キーワードの追加/削除）を可能とする Dynamic SSE [89] である。実システム運用を考えると、暗号化データベースの更新機能は重要となる。

■(Dynamic) SSE 方式の特徴と比較　上で述べた通り、（Dynamic）SSE ではあらかじめ “漏洩を許す情報” を定め、検索処理や更新処理から “許した漏洩” 以上の情報を漏らさないことを保証する。特に Dynamic SSE では、許す漏洩の程度によってフォワード安全性 [90] やバックワード安全性 [91] が明示的に定義される。前者は “文書がデータベースに追加された時、それ以前の検索処理からは追加された文書に関する情報がサーバに漏れない” ことを保証し、検索可能暗号本来の目的である “暗号化したまま検索を行う” ためには、フォワード安全性を満たすことが実用上重要である。後者のバックワード安全性は “文書がデータベースから削除された後、以降の検索処理から削除された文書の情報がサーバに漏れない” ことを保証し、近年最も盛んに研究されている Dynamic SSE の研究テーマである。しかし、その安全性の強さから効率的な実現が難しく構成が複雑になること、バックワード安全性が必要かどうかは応用先に強く依存すると考えられることから、具体的な構成については本ガイドラインでは割愛する。

表 3.1 にいくつかの代表的な Dynamic SSE 方式を示す。表中の記法は次の通り。n はデータベース中の全文書ファイル数を表し、N はデータベース中の（ファ

表 3.1　方式比較。方式名があるものは方式名を、方式名がない場合は著者名を記載している。計算量はいずれもクライアント側のものであり、◎は漸近的に最適である（原理上これ以上の効率化ができない）ことを表す。FP と BP はそれぞれフォワード安全性とバックワード安全性を表す。

	データベースサイズ	更新計算量	検索計算量	FP	BP	備考
[80, 81] ([92])	◎ : $\mathcal{O}(N)$	◎ : $\mathcal{O}(\mu)$	△ : $\mathcal{O}(n)$	✓	–	シンプルな構成
[90]	◎ : $\mathcal{O}(N)$	◎ : $\mathcal{O}(\mu)$	◎ : $\mathcal{O}(n_w)$	✓	–	公開鍵暗号技術が構成に必要
[93]	◎ : $\mathcal{O}(N)$	◎ : $\mathcal{O}(\mu)$	◎ : $\mathcal{O}(n_w)$	✓	–	検索時に少し多く情報漏洩を許す
[94]	○ : $\mathcal{O}(\widetilde{N})$	○ : $\mathcal{O}(u_w)$	○ : $\mathcal{O}(u_w)$	✓	✓	比較的シンプルな構成

イル識別子, キーワード）のペアの数である（データベースの表現については後述する）。通常、削除処理によってデータベースから削除ファイルに対応するペアが削除されるが、多くのバックワード安全な方式では当該ペアを削除せずに “当該ペアが削除された” という情報を新たにデータベースに追加することで削除したとみなす（検索時にクライアントがその情報を確認し、当該ペアを取り除くことで正しく検索結果を取り出すことができる）。そのため、データベースサイズは増大し、表中では $\widetilde{N}$ $(\geq N)$ で書く。μ はその更新処理によってデータベースに追加/削除されるキーワードの数を表す。n_w はデータベース中の検索キーワード w を含むファイル数を表し、u_w はこれまでに w に対して行われてきた更新処理（追加及び削除）の総数を表す。すなわち、$n_w \leq u_w$ が成り立つ。

なお、ほとんどの Dynamic SSE 方式においてクライアント側で秘密鍵以外の秘密情報を記憶しておく必要がある。そのような情報のサイズも比較対象ではあるものの、ここでは割愛する。

■代表的な（Dynamic）SSE 方式の紹介　具体的な検索可能暗号の実現方法として、代表的かつシンプルな方式を紹介する。なお、構成要素技術として、疑似ランダム関数（Pseudo-Random Function, PRF）π を用いる。PRF π は、秘密鍵 k と任意の値 x を入力に取り、y を出力する。この時、y は同じ長さの乱数と見分けがつかず、疑似乱数として扱うことができる。また、データベースの記法についてもまとめる。識別子 id を持つ文書ファイル f_{id} に対し、 形態素解析等を用いてキー

ワードを抽出した集合を $\mathcal{W}_{\mathsf{id}}$ とする。ここで、識別子 id は文書ファイル f_{id} とは全く関係のない値（たとえば文書管理番号や文書ファイルのハッシュ値）であるとする。(Dynamic) SSE で考えるデータベースは、文書ファイル $(\mathsf{id}, \mathcal{W}_{\mathsf{id}})$ を基に作られた（ファイル識別子, キーワード） のペア (id, w) の集合とする。

以下では、Curtmola らによる SSE 方式である SSE-2 [80, 81] を紹介する。より正確には、オリジナルの方式 [80, 81] には誤り及び冗長な点が含まれるため、その点を修正し、かつ Dynamic SSE に拡張した方式 [92]（表 3.1 の一つ目の方式）を示す。なお、本方式はフォワード安全性を満たす。

- セットアップ及びデータベース生成：クライアントは、PRF 鍵 k をランダムに選び、空の配列 Index を暗号化データベースとして用意しておく。
- 文書ファイルの追加：追加したい文書ファイルを $f_{\mathsf{id}} = (\mathsf{id}, \mathcal{W}_{\mathsf{id}})$ とする。クライアントは、全ての $w \in \mathcal{W}_{\mathsf{id}}$ に対して、$\mathsf{addr} := \pi(k, w \| \mathsf{id})$ を計算する。クライアントは計算した全てのアドレスと識別子 id をサーバに送り、サーバは全てのアドレス addr に対し、$\mathsf{Index}[\mathsf{addr}]$ に id を格納し、暗号化データベースを更新する。クライアントは別途追加した id を記憶しておく。
- 文書ファイルの削除：削除したい文書ファイルの識別子 id に対し、クライアントは id をサーバに送る。サーバは Index の格納値を調べ、id が格納されているアドレスを全て空にし、暗号化データベースを更新する。クライアントは記憶していた id を削除する。
- 検索：検索したいキーワード q に対し、クライアントは記憶している全ての識別子 id に対して、トラップドア $\tau_{\mathsf{id}} := \pi(k, q \| \mathsf{id})$ を計算し、計算したトラップドアを全てサーバに送る。サーバは、受け取った各 τ_{id} に対して、$\mathsf{Index}[\tau_{\mathsf{id}}]$ の格納値を確認する。もし何かしらの値（すなわち id）が格納されていれば、それを集合 $\mathcal{X}_q$ に含める（$\mathcal{X}_q$ は最初に空集合として初期化しておく）。最終的にサーバはクライアントに $\mathcal{X}_q$ を渡し、クライアントは $\mathcal{X}_q$ を検索結果として受理する。

上記の構成において、文書ファイルの追加をセットアップ時のみに行い、それ以降追加も削除も行わなければ、Dynamic ではない通常の SSE 方式となる。

参考文献

[1] A. Shamir: “How to Share a Secret”, Commun. ACM, **22**, 11, pp. 612–613 (1979).

[2] G. R. Blakley: “Safeguarding cryptographic keys”, MARK 1979, pp. 313–

317 (1979).
[3] M. Itoh, A. Saito and T. Nishizeki: “Secret Sharing Scheme Realizing General Access Structure”, IEEE Globecom 1987, pp. 99–102 (1987).
[4] B. Chor, G. Goldwasser, S. Micali and B. Awerbuch: “Verifiable Secret Sharing Simultaneity in the Presence of Faults”, FOCS 1985, ACM, pp. 383–395 (1985).
[5] M. Ben-Or, S. Goldwasser and a. Wigderson: “Completeness Theorems for Non-Cryptographic Fault Tolerant Distributed Computation”, STOC 1988, ACM, pp. 1–10 (1988).
[6] M. Naor and A. Shamir: “Visual Cryptography”, EUROCRYPT 1994, Springer-Verlag, pp. 1–12 (1994).
[7] M. Hillery, V. Buzěk and A. Berthiaume: “Quantum Secret Sharing”, Los Alamos e-print archive, **quant-ph/9806063**, (1998).
[8] R. Cleve, D. Gottesman and H.-K. Lo: “How to Share a Quantum Secret”, Physical Review Letters, **83**, 3, pp. 648–651 (1999).
[9] A. Shamir: “How to share a secret”, Commun. ACM, **22**, 11, pp. 612–613 (1979).
[10] E. Karnin, J. Greene and M. E. Hellman: “On Secret Sharing Systems”, IEEE Trans. Information Theory, **29**, 1, pp. 35–41 (1983).
[11] S. Martín, C. Padró and A. Yang: “Secret Sharing, Rank Inequalities and Information Inequalities”, CRYPTO 2013, Springer-Verlag, pp. 277–288 (2013).
[12] M. Karchmer and A. Wigderson: “On Span Programs”, Annual Structure in Complexity Theory Conference, pp. 102–111 (1993).
[13] G. R. Blakley and C. Meadows: “Security of ramp schemes”, CRYPTO 1984, Springer-Verlag, pp. 242–268 (1984).
[14] 山本：“(k, L, n) しきい値秘密分散システム”, 電子通信学会論文誌, **J68-A**, 9, pp. 945–952 (1985).
[15] H. Krawczyk: “Secret sharing made short”, CRYPTO 1993, Springer-Verlag, pp. 136–146 (1993).
[16] S. Shin, S. Yamada, G. Hanaoka, Y. Ishida, A. Kunii, J. Oketani, S. Kunii and K. Tomomura: “How to Extend CTRT for AES-256 and AES-192”, IEICE Trans., **E105-A(8)**, pp. 1121–1133 (2022).
[17] “ZENMU-AONT に関する論文が 国際会議で best paper premium award を

受賞”, https://www.atpress.ne.jp/news/191661.
[18] “ISO/IEC 19592-2:2017 Information technology - Security techniques - Secret sharing - Part 2: Fundamental mechanisms”.
[19] M. Iwamoto and H. Yamamoto: “Strongly Secure Ramp Secret Sharing Schemes for General Access Structures”, Information Processing Letters, **97**, 2, pp. 52–57 (2006).
[20] J. Benaloh and J. Leichter: “Generalized Secret Sharing and Monotone Functions”, CRYPTO 1988, Springer New York, pp. 27–35 (1988).
[21] G. S. Vernam: “Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communications”, J. of American Institute for Electrical Engineering, **45**, pp. 109–115 (1926).
[22] D. Beaver, S. Micali and P. Rogaway: “The Round Complexity of Secure Protocols (Extended Abstract)”, STOC 1990, ACM, pp. 503–513 (1990).
[23] Y. Lindell, B. Pinkas, N. P. Smart and A. Yanai: “Efficient Constant-Round Multi-party Computation Combining BMR and SPDZ”, J. of Cryptology, **32**, 3, pp. 1026–1069 (2019).
[24] O. Goldreich, S. Micali and A. Wigderson: “How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority”, STOC 1987, ACM, pp. 218–229 (1987).
[25] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali and P. Rogaway: “Everything Provable is Provable in Zero-Knowledge”, CRYPTO 1988, Springer-Verlag, pp. 37–56 (1988).
[26] T. Araki, J. Furukawa, Y. Lindell, A. Nof and K. Ohara: “High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority”, ACM CCS 2016, pp. 805–817 (2016).
[27] N. Attrapadung, G. Hanaoka, T. Matsuda, H. Morita, K. Ohara, J. C. N. Schuldt, T. Teruya and K. Tozawa: “Oblivious Linear Group Actions and Applications”, ACM CCS 2021, pp. 630–650 (2021).
[28] Y. Lindell and A. Nof: “A Framework for Constructing Fast MPC over Arithmetic Circuits with Malicious Adversaries and an Honest-Majority”, ACM CCS 2017, pp. 259–276 (2017).
[29] R. Cramer, I. Damgård and Y. Ishai: “Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation”, TCC 2005, Springer-Verlag, pp. 342–362 (2005).

[30] NTT: “秘密計算基盤（Trust-SC）V3.0「算師（R)」｜ NTT R&D Website”. https://www.rd.ntt/research/SI0005.html.
[31] R. Kikuchi, D. Ikarashi, T. Matsuda, K. Hamada and K. Chida: “Efficient Bit-Decomposition and Modulus-Conversion Protocols with an Honest Majority”, ACISP 2018, Springer-Verlag, pp. 64–82 (2018).
[32] D. Beaver: “Efficient Multiparty Protocols Using Circuit Randomization”, CRYPTO 1991, Springer-Verlag, pp. 420–432 (1991).
[33] I. Damgård, V. Pastro, N. P. Smart and S. Zakarias: “Multiparty Computation from Somewhat Homomorphic Encryption”, CRYPTO 2012, Springer-Verlag, pp. 643–662 (2012).
[34] N. P. Smart and T. Tanguy: “TaaS: Commodity MPC via Triples-as-a-Service”, CCSW@CCS 2019, ACM, pp. 105–116 (2019).
[35] ZenmuTech: “データを秘匿化したま計算｜ ZenmuTech”. https://zenmutech.com/products/secure-computation/.
[36] 石田, 國井, 桶谷, 大畑, 松田, アッタラパドゥン, 花岡：“Query Ahead: 平易な記述が可能な秘匿 DB クエリーシステムの設計と実装”, SCIS 2020 (2020).
[37] A. C. Yao: “How to Generate and Exchange Secrets (Extended Abstract)”, FOCS 1986, IEEE Computer Society, pp. 162–167 (1986).
[38] M. Bellare, V. T. Hoang and P. Rogaway: “Foundations of Garbled Circuits”, ACM CCS 2012, pp. 784–796 (2012).
[39] M. O. Rabin: “How To Exchange Secrets with Oblivious Transfer”. ePrint 187/2005.
[40] S. Zahur, M. Rosulek and D. Evans: “Two Halves Make a Whole - Reducing Data Transfer in Garbled Circuits Using Half Gates”, EUROCRYPT 2015, Springer-Verlag, pp. 220–250 (2015).
[41] V. Kolesnikov and T. Schneider: “Improved Garbled Circuit: Free XOR Gates and Applications”, ICALP 2008, Springer-Verlag, pp. 486–498 (2008).
[42] M. Naor, B. Pinkas and R. Sumner: “Privacy preserving auctions and mechanism design”, ACM EC 1999, pp. 129–139 (1999).
[43] B. Pinkas, T. Schneider, N. P. Smart and S. C. Williams: “Secure Two-Party Computation Is Practical”, ASIACRYPT 2009, Springer-Verlag, pp. 250–267 (2009).
[44] S. Goldwasser, S. Micali and C. Rackoff: “The Knowledge Complexity of Interactive Proof Systems”, SIAM J. Computing, **18**, 1, pp. 186–208 (1989).

[45] B. Manuel, Feldman and P. M. Silvio: "Non-Interactive Zero-Knowledge and Its Applications", STOC 1988, ACM, pp. 103–112 (1988).

[46] C.-P. Schnorr: "Efficient Identification and Signatures for Smart Cards", CRYPTO 1989, Springer-Verlag, pp. 239–252 (1989).

[47] R. Gennaro, C. Gentry, B. Parno and M. Raykova: "Quadratic Span Programs and Succinct NIZKs without PCPs", EUROCRYPT 2013, Springer-Verlag, pp. 626–645 (2013).

[48] B. Parno, J. Howell, C. Gentry and M. Raykova: "Pinocchio: Nearly Practical Verifiable Computation", IEEE S&P 2013 (2013).

[49] J. Groth: "On the Size of Pairing-Based Non-interactive Arguments", EUROCRYPT 2016, Springer-Verlag, pp. 305–326 (2016).

[50] E. Ben-Sasson, I. Bentov, Y. Horesh and M. Riabzev: "Scalable Zero Knowledge with No Trusted Setup", CRYPTO 2019, Springer-Verlag, p. 701–732 (2019).

[51] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille and G. Maxwell: "Bulletproofs: Short Proofs for Confidential Transactions and More ", IEEE S&P 2018, pp. 315–334 (2018).

[52] O. Goldreich: "Towards a Theory of Software Protection and Simulation by Oblivious RAMs", STOC 1987, ACM, pp. 182–194 (1987).

[53] R. Ostrovsky: "Efficient computation on oblivious RAMs", STOC 1990, ACM, pp. 514–523 (1990).

[54] O. Goldreich and R. Ostrovsky: "Software protection and simulation on oblivious RAMs", **43 (3)**, pp. 431–473 (1996).

[55] E. Shi, T.-H. H. Chan, E. Stefanov and M. Li: "Oblivious RAM with $O((\log N)^3)$ worst-case cost", ASIACRYPT 2011, Springer-Verlag, pp. 197–214 (2011).

[56] E. Kushilevitz, S. Lu and R. Ostrovsky: "On the (In)Security of Hash-Based Oblivious RAM and A New Balancing Scheme", SODA 2012, pp. 143–156 (2012).

[57] E. Stefanov, M. van Dijk, E. Shi, C. W. Fletcher, L. Ren, X. Yu and S. Devadas: "Path ORAM: an extremely simple oblivious RAM protocol", ACM CCS 2013, pp. 299–310 (2013).

[58] S. Patel, G. Persiano, M. Raykova and K. Yeo: "PanORAMa: Oblivious RAM with Logarithmic Overhead", FOCS 2018, ACM, pp. 871–882 (2018).

[59] G. Asharov, I. Komargodski, W.-K. Lin, K. Nayak, E. Peserico and E. Shi: "OptORAMa: Optimal Oblivious RAM", EUROCRYPT (2) 2020, Springer-Verlag, pp. 403–432 (2020).

[60] G. Asharov, I. Komargodski, W.-K. Lin and E. Shi: "Oblivious RAM with Worst-Case Logarithmic Overhead", CRYPTO (4) 2021, Springer-Verlag, pp. 610–640 (2021).

[61] S. Lu and R. Ostrovsky: "Distributed Oblivious RAM for Secure Two-Party Computation", TCC 2013, Springer-Verlag, pp. 377–396 (2013).

[62] E. Boyle, K.-M. Chung and R. Pass: "Oblivious parallel ram and applications", TCC 2016, Springer-Verlag, pp. 175–204 (2016).

[63] B. Chor, E. Kushilevitz, O. Goldreich and M. Sudan: "Private Information Retrieval", FOCS, IEEE, pp. 41–50 (1995).

[64] B. Chor, E. Kushilevitz, O. Goldreich and M. Sudan: "Private Information Retrieval", J. ACM, **45**, 6, pp. 965–981 (1998).

[65] E. Kushilevitz and R. Ostrovsky: "Replication Is Not Needed: Single Database, Computationally-Private Information Retrieval", FOCS 1997, IEEE, pp. 364–373 (1997).

[66] H. Corrigan-Gibbs and D. Kogan: "Private Information Retrieval with Sublinear Online Time", EUROCRYPT 2020, Springer-Verlag, pp. 44–75 (2020).

[67] H. Corrigan-Gibbs, A. Henzinger and D. Kogan: "Single-Server Private Information Retrieval with Sublinear Amortized Time", EUROCRYPT 2022, Springer International Publishing, pp. 3–33 (2022).

[68] L. Holzbaur, C. Hollanti and A. Wachter-Zeh: "Computational Code-Based Single-Server Private Information Retrieval", IEEE ISIT 2020, pp. 1065–1070 (2020).

[69] G. N. Alfarano, K. Khathuria and V. Weger: "A Survey on Single Server Private Information Retrieval in A Coding Theory Perspective", Applicable Algebra in Engineering, Communication and Computing (2021).

[70] H. Sun and S. A. Jafar: "The Capacity of Private Information Retrieval", IEEE Trans. Information Theory, **63**, 7, pp. 4075–4088 (2017).

[71] D. Augot, F. Levy-dit Vehel and A. Shikfa: "A Storage-Efficient and Robust Private Information Retrieval Scheme Allowing Few Servers", CANS 2014, Springer International Publishing, pp. 222–239 (2014).

[72] J. Katz and L. Trevisan: "On The Efficiency of +Local Decoding Procedures for Error-Correcting Codes", STOC 2000, ACM, pp. 80–86 (2000).

[73] A. Beimel and Y. Stahl: "Robust Information-Theoretic Private Information Retrieval", J. of Cryptology, **20**, 3, pp. 295–321 (2007).

[74] K. Kurosawa: "How to Correct Errors in Multi-server PIR", ASIACRYPT 2019, Springer-Verlag, pp. 564–574.

[75] D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano: "Public Key Encryption with Keyword Search", EUROCRYPT 2004, Springer-Verlag, pp. 506–522 (2004).

[76] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier and H. Shi: "Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions", J. of Cryptology, **21**, 3, pp. 350–391 (2007).

[77] Q. Tang: "Towards Forward Security Properties for PEKS and IBE", Information Security and Privacy, ACISP 2015, Springer-Verlag, pp. 127–144 (2015).

[78] R. Chen, Y. Mu, G. Yang, F. Guo and X. Wang: "A New General Framework for Secure Public Key Encryption with Keyword Search", Information Security and Privacy, ACISP 2015, Springer-Verlag, pp. 59–76 (2015).

[79] P. Golle, J. Staddon and B. Waters: "Secure Conjunctive Keyword Search over Encrypted Data", ACNS 2004, Springer-Verlag, pp. 31–45 (2004).

[80] R. Curtmola, J. Garay, S. Kamara and R. Ostrovsky: "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions", ACM CCS 2006, pp. 79–88 (2006).

[81] R. Curtmola, J. A. Garay, S. Kamara and R. Ostrovsky: "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions", J. of Computer Security, **19**, 5, pp. 895–934 (2011).

[82] D. X. Song, D. Wagner and A. Perrig: "Practical Techniques for Searches on Encrypted Data", IEEE S&P 2000, pp. 44–55 (2000).

[83] G. Asharov, G. Segev and I. Shahaf: "Tight Tradeoffs in Searchable Symmetric Encryption", J. of Cryptology, **34**, 9, pp. 1–37 (2021).

[84] G. Asharov, M. Naor, G. Segev and I. Shahaf: "Searchable Symmetric Encryption: Optimal Locality in Linear Space via Two-Dimensional Balanced Allocations", SIAM J. Computing, **50**, 5, pp. 1501–1536 (2021).

[85] D. Cash and S. Tessaro: "The Locality of Searchable Symmetric Encryption", EUROCRYPT 2014, Springer-Verlag, pp. 351–368 (2014).

[86] K. Kurosawa and Y. Ohtaki: "UC-Secure Searchable Symmetric Encryption", Financial Cryptography and Data Security, FC 2012, Springer-Verlag, pp. 285–298 (2012).

[87] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu and M. Steiner: "Highly-Scalable Searchable Symmetric Encryption with Support for Boolean Queries", CRYPTO 2013, Springer-Verlag, pp. 353–373 (2013).

[88] M. Chase and S. Kamara: "Structured Encryption and Controlled Disclosure", ASIACRYPT 2010, Springer-Verlag, pp. 577–594 (2010).

[89] S. Kamara, C. Papamanthou and T. Roeder: "Dynamic Searchable Symmetric Encryption", ACM CCS 2012, pp. 965–976 (2012).

[90] R. Bost: "$\Sigma o \varphi o \varsigma$: Forward Secure Searchable Encryption", ACM CCS 2016, pp. 1143–1154 (2016).

[91] R. Bost, B. Minaud and O. Ohrimenko: "Forward and Backward Private Searchable Encryption from Constrained Cryptographic Primitives", ACM CCS 2017, pp. 1465–1482 (2017).

[92] Y. Watanabe, T. Nakai, K. Ohara, T. Nojima, Y. Liu, M. Iwamoto and K. Ohta: "How to Make a Secure Index for Searchable Symmetric Encryption, Revisited", IEICE Trans., **E105-A(12)**, pp. 1559–1577 (2022).

[93] M. Etemad, A. Küpçü, C. Papamanthou and D. Evans: "Efficient Dynamic Searchable Encryption with Forward Privacy", Privacy Enhancing Technologies (PoPETs), Vol. 2018(1), pp. 5–20 (2018).

[94] J. G. Chamani, D. Papadopoulos, C. Papamanthou and R. Jalili: "New Constructions for Forward and Backward Private Symmetric Searchable Encryption", ACM CCS 2018, pp. 1038–1055 (2018).

第 4 章

おわりに

本ガイドラインに記載した内容は、2022 年 9 月以前に公開された情報に基づいている。2022 年 9 月以降も多くの高機能暗号に関連する情報が、各社の広報、学会、標準化団体において公開されてきているが、発表されたばかりであり、安全性の評価が完全には行われていない方式もある。本ガイドラインで記載した高機能暗号は、2022 年 9 月時点で、主要国際学会で発表されており、有力な攻撃法が発見されておらず、かつ、十分な性能を持つと考えられる方式を選んでいる。

高機能暗号の特徴であるが、機能が異なれば、異なる暗号となる。このため、多くの種類の高機能暗号があり、それらすべて本ガイドラインに掲載することはできない。主要と考えられる高機能暗号に絞って掲載することになった。

また、高機能暗号は多種多様であるが故に、導入するためには専門的な知識が要求される。この知識を有し、かみ砕いて説明でき、ケースに応じて応用するスキルを有する人材の確保は課題となっている。さらに、既存システムとの整合性を考慮し、例えば、組織内の端末にソフトウェアとして導入できるか、マネージメントシステムと連携できるか、Web ブラウザ上で動作ができるか、インターフェースはどのような形態がよいか等、これらは実際のシステム開発者や利用者と密に連携をとりつつ開発することが必要となる。

本ガイドラインが、用途に適した高機能暗号を選択するため、および、研究者、システム開発者、利用者にとっての指針となれば幸いである。

CRYPTREC 暗号技術ガイドライン（高機能暗号）
[CRYPTREC GL-2005-2022]

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
  (技術本部　セキュリティセンター　暗号グループ)
  INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN
  2-28-8 HONKOMAGOME, BUNKYO-KU
  TOKYO, 113-6591 JAPAN

資料３－８

軽量暗号に関する技術動向調査報告（外部評価）

1. 背景

2019 年度に設置された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にて、「CRYPTREC において、軽量暗号は CRYPTREC 暗号リストには組み込まず、別途ガイドラインという形で取り扱う」ことが決定された。本決定を受け、2020 年度第 2 回暗号技術検討会において、2016 年度に作成した「CRYPTREC 暗号技術ガイドライン（軽量暗号）」（以下、「2016 年度版ガイドライン」という）について 2023 年度中を目処に更新することが承認された。

2016 年度版ガイドラインの更新方針を決定するにあたり、2021 年度に『2016 年度版ガイドラインに掲載されている暗号方式の大幅な安全性の劣化に繋がる脆弱性の有無』『軽量暗号に関わる ISO/IEC 29192 シリーズに近年採択された、又は採択予定の暗号方式の 2016 年度版ガイドライン掲載の有無及びその安全性』『CAESAR プロジェクトで最終的に選ばれたポートフォリオ 6 方式の 2016 年度版ガイドライン掲載の有無および掲載されていない暗号方式の安全性』に関して調査した。

上記の調査結果に基づき 2016 年度版ガイドラインの更新方針を策定し、2021 年度第 2 回暗号技術評価委員会において本更新方針が承認された。

承認された更新方針に従い、今年度は、NIST 軽量暗号プロジェクト（NIST[1] Lightweight Cryptography Project[2]．以下、「NIST LWC」という）のファイナリスト 10 方式（ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJAMBU, Xoodyak）を対象とした安全性評価及び実装性能評価を外部評価により実施した。なお、安全性評価に関しては、上記 10 方式に加え、ISO/IEC 標準規格として 29192 シリーズで規格化された軽量メッセージ認証コードの 1 つである Tsudik’s keymode も対象としている。また、軽量暗号に関わる NIST 公開文書や ISO などの標準化動向に関わる調査を外部評価により実施した。

本資料にて、外部評価実施内容について報告する。

なお、2023 年 2 月 7 日に、NIST LWC 最終選考結果が発表され ASCON が選ばれた。

---

[1] アメリカ国立標準技術研究所（National Institute of Standards and Technology）

[2] https://csrc.nist.gov/projects/lightweight-cryptography

2. 実施概要

2.1. 安全性評価

NIST LWC ファイナリストに選定された 10 方式と ISO/IEC 標準規格として承認された Tsudik's keymode の安全性に関する調査及び評価を実施した。

【件名】

軽量暗号の安全性に関する調査及び評価（各依頼先に応じた暗号方式を記載）

【依頼内容】

対象方式の安全性評価について、公開されている評価結果の有無を調査し、評価結果が存在する場合にはその影響範囲についてまとめ、評価結果が存在しない場合には安全性評価を実施し、評価報告書を作成する。

【依頼先および依頼対象方式】

岩田 哲 准教授（名古屋大学）

- 認証暗号＋ハッシュ関数：PHOTON-Beetle, SPARKLE
- 軽量 MAC：Tsudik's keymode

内藤 祐介 様（三菱電機株式会社）

- 認証暗号＋ハッシュ関数：Xoodyak
- 認証暗号：GIFT-COFB

藤堂 洋介 様（日本電信電話株式会社）

- 認証暗号＋ハッシュ関数：ASCON
- 認証暗号：Grain-128AEAD, TinyJAMBU

井上 明子 様（日本電気株式会社）

- 認証暗号＋ハッシュ関数：Romulus
- 認証暗号：Elephant, ISAP

2.2. 実装性能評価

NIST LWC ファイナリストに選定された 10 方式の実装性能（ハードウェア及びソフトウェア）に関する調査及び評価を実施した。

【件名】

軽量暗号の実装性能に関する調査及び評価（NIST LWC）

【依頼内容】

NIST LWC でファイナリストに選定された 10 方式に対する実装性能評価について、公開されている評価結果を調査し、評価結果についてまとめ、考察などを行い、評価報告書を作成する。

【依頼先および依頼対象方式】

崎山 一男 教授（電気通信大学）

・認証暗号＋ハッシュ関数：ASCON, PHOTON-Beetle, Romulus, SPARKLE, Xoodyak

・認証暗号：Elephant, GIFT-COFB, Grain-128AEAD, ISAP, TinyJAMBU

2.3. 標準化動向など

軽量暗号を取り巻く標準化動向(CAESAR プロジェクト、ISO/IEC の軽量暗号関連カテゴリ、NIST LWC など)の調査を実施した。

【件名】

軽量暗号の評価指標、標準化動向に関する調査（NIST LWC ファイナリストなど）

【依頼内容】

「NIST LWC 最終選考に選定された方式」に関わる選定指標や評価の観点をまとめるとともに、「軽量な方式として ISO/IEC 標準規格として近年承認されたもしくは承認される予定の方式」に関わる軽量暗号に関わる技術動向について、公開情報を基にまとめ、考察などを行い、報告書を作成する。

【依頼先】

菅野 哲 様（GMO サイバーセキュリティ by イエラエ株式会社）

3. 調査結果概要

3.1. 安全性評価

3.1.1. 概要

本節では安全性評価に関する調査結果の概要を方式ごとにまとめた。

NIST LWC ファイナリストに選定された 10 方式（ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJAMBU, Xoodyak）と ISO 採択方式である Tsudik's keymode の安全性評価について、2022 年 9 月現在における調査結果を次のとおり表にまとめた。

安全性を脅かす攻撃が存在しない方式	特定の場合を除き、 安全性を脅かす方式が存在しない方式
ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, Xoodyak	TinyJAMBU, Tsudik's keymode

TinyJAMBU では、関連鍵設定の場合に現実的な計算量での偽造攻撃が実行可能である。本攻撃が成立するような関連鍵の使用を避けることで TinyJambu の安全性を確保できることを確認した。

Tsudik's keymode では、使用するハッシュ関数が伸長攻撃[3]を許す場合に偽造攻撃が実行可能となるという既知の脆弱性が存在する。伸長攻撃が実行不可能なハッシュ関数を使用することで Tsudik's keymode の安全性を確保できることを確認した。

3.1.2. ASCON

ASCON は独自の暗号学的置換をプリミティブとして使用した Duplex 構造の認証暗号（ASCON-128, ASCON-128a）と Sponge 構造のハッシュ関数（ASCON-Hash, ASCON-Hasha）をサポートしている。認証暗号における暗号化処理とハッシュ関数における処理は図 1 と図 2 のとおり。Primary member[4]は ASCON-128 と ASCON-Hash である。

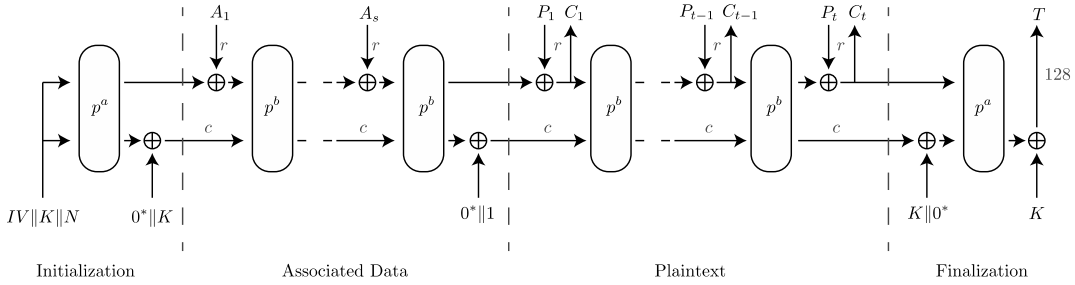


図 1 ：ASCON の認証暗号における暗号化処理

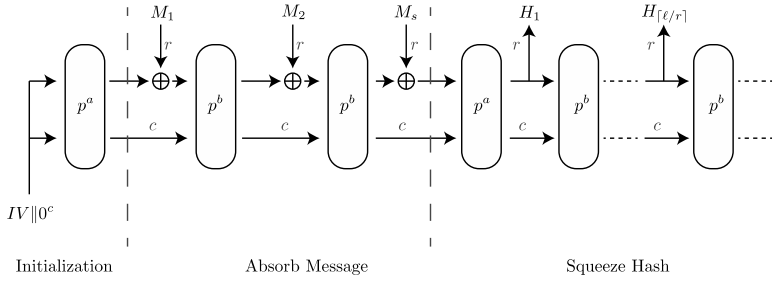


図 2：ASCON のハッシュ関数における処理

---

[3] 伸長攻撃（length-extention attack）は、マークル・ダンガード構成のような反復型ハッシュ関数Hについて、$H(M|M')$が$H(M)$とM'のみから計算できるという性質を利用する攻撃である。

[4] 複数個の提案方式の中から設計者が一番に推奨する暗号方式のことを指す。primary recommendation と記載している仕様書もあるが、本資料では primary member で統一する。なお、提案方式におけるパラメータは NIST LWC の要件である鍵長 128 ビット以上、ナンス長 96 ビット以上、タグ長 64 ビット以上を満たす値が設定されている。

認証暗号に対する安全性はDuplex構造の安全性に依存する[5]。プリミティブを考慮した安全性評価として、初期化フェーズ、データ処理フェーズ、タグ生成フェーズの3つのフェーズを対象とした解析結果が多数報告されている。各フェーズに対する最良の攻撃手法は次のとおり。なお、攻撃可能段数は暗号学的置換におけるラウンド関数の段数を表している。

フェーズ	攻撃手法	攻撃目標	攻撃可能段数
初期化	キューブ攻撃	鍵回復	12段中7段
データ処理	SAT solver	内部状態復元	8段中2段
タグ生成	差分攻撃	偽造	12段中2段

設計者による安全性評価によると、仕様段数である12段の暗号学的置換に対してゼロサム識別攻撃が実行可能である。しかし、本攻撃がASCONの認証暗号としての脆弱性に繋がらないため認証暗号は安全である、と設計者は主張している。

ハッシュ関数に対する安全性はSponge構造の安全性に依存する。プリミティブを考慮した安全性評価として、いくつかの解析結果が報告されている。ハッシュ関数に対する最良の攻撃手法は次のとおり。

攻撃手法	攻撃目標	攻撃可能段数
代数攻撃	原像回復	12段中6段
差分攻撃	衝突	12段中2段

以上より、ASCONの安全性解析に関する文献が多数報告されているものの、認証暗号とハッシュ関数について安全性を脅かす攻撃手法は存在しないことを確認した。

3.1.3. Elephant

Elephantは暗号学的置換をプリミティブとして使用した認証暗号利用モードの名称であり、このモード構成を利用した3つの認証暗号Dumbo, Jumbo, Delirium をまとめた総称でもある。これら3つの暗号方式は使用する暗号学的置換が異なり、それぞれSPONGENT-π[160], SPONGENT-π[176], Keccak-f[200]をプリミティブとして使用する。Primary memberはDumboである。

設計者が主張するElephantの安全性は次のとおり。ここで、オフライン計算量は公開ランダム置換へのアクセス回数、オンライン計算量は1つの鍵で処理可能なデータ量の上限を表す。

[5] プリミティブを理想化した場合において使用する暗号利用モードの安全性に帰着できることを意味する。別途、暗号利用モードで使用されるプリミティブを考慮した安全性評価が必要となる。

方式	オフライン計算量	オンライン計算量
Dumbo	2^{112}	2^{50} バイト
Jumbo	2^{127}	2^{50} バイト
Delirium	2^{127}	2^{74} バイト

Elephant の安全性証明に関する文献、使用するプリミティブの安全性解析に関する文献、サイドチャネル攻撃等の非ブラックボックス安全性の解析に関する文献がいくつか報告されているものの、これら 3 つの暗号方式について、設計者が主張する安全性を脅かす攻撃は存在しないことを確認した。

3.1.4. GIFT-COFB

GIFT-COFB はブロック暗号をプリミティブとして使用した認証暗号であり、ブロック暗号ベースの認証暗号利用モード COFB とブロック暗号 GIFT-128 を組み合わせた方式である。

設計者が主張する GIFT-COFB の安全性は次のとおり。ここで、オフライン計算量は公開ランダム置換へのアクセス回数、オンライン計算量は 1 つの鍵で処理可能なデータ量の上限を表す。

IND-CPA		INT-CTXT	
オフライン計算量	オンライン計算量	オフライン計算量	オンライン計算量
112 ビット	64 ビット	112 ビット	58 ビット

GIFT-128 に対する最良の攻撃手法は差分解読法による鍵回復攻撃であり、20 段の差分識別子を使用して 40 段中 27 段に対して鍵回復攻撃が実行可能となる。つまり、オフライン計算に関して設計者が主張する 112 ビット安全性を有していると言える。

COFB の安全性はブロック暗号が擬似ランダム置換（PRP）であることを仮定している。上述のとおり、GIFT-128 の安全性は担保されているため、GIFT-COFB の安全性は COFB の安全性に依存する。COFB の提案論文において、ブロックサイズが n ビット、ブロック暗号が PRP 安全であると仮定すると、IND-CPA のオンライン計算に関して n/2 ビット、INT-CTXT のオンライン計算に関して $n/2-\log_2 n$ ビットとなることが証明されている。現時点でこの安全性証明を覆す結果が報告されていないことを確認した。

3.1.5. Grain-128AEAD

Grain-128AEAD はストリーム暗号 Grain の系譜（Grain v0, Grain v1, Grain-128, Grain-128a）を継ぐ認証暗号である。NIST LWC 期間中、内部状態が既知という仮定の

下で鍵回復攻撃が実行可能であるとの報告を受け、現在は Grain-128AEADv2 として仕様が更新されている。

Grain-128AEADv1 と Grain-128AEADv2 の安全性評価に関する文献は少ないものの、概ね等価な Grain-128a の安全性評価に関する文献は多数報告されており、これらの解析結果に基づき Grain-128AEADv2 の安全性を考察できる。Grain-128a に対して脅威となった攻撃手法は高速相関攻撃とキューブ攻撃であり、これらの攻撃手法に対する Grain-128AEADv2 の安全性を考察した結果、Grain-128AEADv2 はこれらの攻撃手法に対して安全性マージンを十分に有していることを確認した。

3.1.6. ISAP

ISAP は暗号学的置換をプリミティブとして使用した認証暗号利用モードの名称であり、このモード構成を利用した 4 つの認証暗号 ISAP-A-128A, ISAP-K-128A, ISAP-A-128, ISAP-K-128 をまとめた総称でもある。これら 4 つの暗号方式は使用する暗号学的置換が異なり、ISAP-A-128A と ISAP-A-128 は ASCON-p、ISAP-K-128A と ISAP-K-128 は Keccak-p[400]を使用する。Primary member は ISAP-A-128A である。サイドチャネル攻撃に対して堅牢となるよう Fresh rekeying と呼ばれる概念を導入していることが特徴である。

設計者は全ての ISAP members があらゆる攻撃に対して 128 ビット安全性を有していると主張している。ブラックボックス安全性証明に関する文献、耐漏洩安全性証明に関する文献、使用するプリミティブの安全性解析に関する文献、サイドチャネル攻撃耐性の解析に関する文献がいくつか報告されているものの、設計者が主張する安全性を脅かす攻撃は存在しないことを確認した。

3.1.7. PHOTON-Beetle

PHOTON-Beetle は暗号学的置換をプリミティブとして使用した認証暗号（PHOTON-Beetle-AEAD[32], PHOTON-Beetle-AEAD[128]）とハッシュ関数（PHOTON-Beetle-Hash[32]）をサポートしており、認証暗号の primary member は PHOTON-Beetle-AEAD[128]である。認証暗号に関しては Duplex 構造を改良した認証暗号利用モード Beetle と暗号学的置換 PHOTON-256、ハッシュ関数に関しては Sponge 構造と PHOTON-256 をそれぞれ組み合わせた方式となっている。

PHOTON-256 に対する最良の攻撃手法はゼロサム識別攻撃であり、仕様段数である 12 段に対して識別攻撃が実行可能である。ただし、本攻撃が PHOTON-Beetle の安全性に

直接影響を及ぼすものではないことに注意が必要である。本攻撃を除き、仕様段数に対して実行可能な攻撃手法は報告されていないことを確認した。

設計者が主張する認証暗号の安全性は次のとおり。

方式	IND-CPA		INT-CTXT	
	データ量	計算量	データ量	計算量
PHOTON-Beetle-AEAD[32]	128 ビット	128 ビット	128 ビット	128 ビット
PHOTON-Beetle-AEAD[128]	121 ビット	121 ビット	121 ビット	121 ビット

認証暗号の安全性証明に関する文献がいくつか報告されているものの、一部を除き設計者が主張する安全性と齟齬がないことを確認した。なお、PHOTON-Beetle-AEAD[32]の IND-CPA と INT-CTXT の計算量に関して理論的根拠が無いことに注意が必要である。

また、設計者が主張するハッシュ関数の安全性は次のとおり。

方式	衝突困難性	原像計算困難性
PHOTON-Beetle-Hash[32]	112 ビット (データ量：$2^{111.5}$)	128 ビット

ハッシュ関数の安全性は Sponge 構造の安全性に依存する。ハッシュ関数の安全性証明に関する文献がいくつか報告されているものの、設計者が主張する安全性と齟齬がないことを確認した。

3.1.8. Romulus

Romulus は Tweakable ブロック暗号の Skinny-128-384+をプリミティブとして使用した認証暗号（Romulus-N, Romulus-M, Romulus-T）とハッシュ関数（Romulus-H）をサポートしている。これら 4 方式は全て同じプリミティブを使用するがモード構成が異なる。認証暗号の Primary member は Romulus-N である。

設計者が主張する認証暗号 3 方式の安全性は次のとおり。ここで、NR は Nonce-respecting、NM は Nonce-misuse を表す。

方式	IND-CPA (NR)	INT-CTXT (NR)	IND-CPA (NM)	INT-CTXT (NM)
Romulus-N	128 ビット	128 ビット	-	-
Romulus-M	128 ビット	128 ビット	64～128 ビット	64～128 ビット
Romulus-T	121 ビット	121 ビット	-	121 ビット

これら 3 方式の安全性証明に関する文献がいくつか報告されているものの、全て設計者が主張する安全性と齟齬がないことを確認した。

また、設計者が主張するハッシュ関数の安全性は次のとおり。

方式	衝突困難性	原像計算困難性	第 2 原像計算困難性
Romulus-H	121 ビット	121 ビット	121 ビット

Romulus-H の安全性証明に関する第三者評価の文献は報告されていない。また、40 段中 23 段に簡略化したプリミティブを使用した場合、原像攻撃と Free-start 設定における衝突攻撃が実行できると報告されているが、これらの攻撃は仕様上の安全性を脅かすものではないことを確認した。

3.1.9. SPARKLE

SPARKLE は独自の暗号学的置換をプリミティブとして使用した Duplex 構造の認証暗号（SCHWAEMM）と Sponge 構造のハッシュ関数（ESCH）をサポートしている。暗号学的置換は入出力サイズに応じて SPARKLE256, SPARKLE384, SPARKLE512 を定義しており、これらの暗号学的置換を使用してキャパシティとレートが異なる 4 つの認証暗号と 2 つのハッシュ関数が提案されている。Primary member は SPARKLE384 を使用した SCHWAEMM256-128 と ESCH256 である。

暗号学的置換としての SPARKLE には big instances と slim instances があり、これらはラウンド関数の段数のみ異なる。例えば、SPARKLE384 の big instance は 11 段、slim instance は 7 段である。設計者による SPARKLE の安全性評価によると、3 方式全てにおいて slim instances を脅かす攻撃手法が存在しないと主張されている。第三者による安全性評価に関する文献がいくつか報告されているものの、設計者が主張する安全性を脅かす結果は報告されていないことを確認した。

設計者が主張する SCHWAEMM の安全性は次のとおり。

方式	IND-CPA	INT-CTXT	データ制限
SCHWAEMM256-128	120 ビット	120 ビット	2^{68} バイト
SCHWAEMM192-192	184 ビット	184 ビット	2^{68} バイト
SCHWAEMM128-128	120 ビット	120 ビット	2^{68} バイト
SCHWAEMM256-256	248 ビット	248 ビット	2^{133} バイト

SCHWAEMM は Duplex 構造の認証暗号利用モードである Beetle をベースに構成されており、SCHWAEMM の安全性は Beetle の安全性に依存する。現時点で設計者が主張する安全性を脅かす解析結果は存在しないことを確認した。

また、設計者が主張する ESCH の安全性は次のとおり。

方式	衝突困難性	原像計算困難性	第 2 原像計算困難性	データ制限
ESCH256	128 ビット	128 ビット	128 ビット	2^{132} バイト
ESCH384	192 ビット	192 ビット	192 ビット	2^{196} バイト

ESCH の安全性は Sponge 構造の安全性に依存する。現時点で設計者が主張する安全性を脅かす解析結果は存在しないことを確認した。

3.1.10. TinyJAMBU

TinyJAMBU は独自の鍵付き暗号学的置換をプリミティブとして使用した Duplex 構造の認証暗号である。秘密鍵のサイズは 128, 192, 256 ビットの 3 種類をサポートしており、それぞれ TinyJAMBU-128, TinyJAMBU-192, TinyJAMBU-256 と呼ばれている。

単一鍵設定での最良な攻撃手法は、タグ生成フェーズを対象とした線形解読法による部分鍵回復攻撃である。攻撃可能段数は 640 段中 387 段であるため、安全性マージンを十分に有していることを確認した。なお、TinyJAMBU の現バージョンは v2 であるが、v1 のタグ生成フェーズにおける暗号学的置換は 384 段構成であったため、v1 は単一鍵設定において安全性を有していないことを確認した。

関連鍵設定での最良な攻撃手法は、差分解読法による偽造攻撃である。適切な秘密鍵とナンスペアを利用することで、$2^{32}+2^{14}$ の計算量で関連鍵偽造攻撃が実行可能となる。なお、本攻撃は TinyJAMBU-128 には有効ではない。

その他、プリミティブに対する安全性評価として、スライド攻撃による鍵回復攻撃が報告されている。TinyJAMBU を構成する鍵付き暗号学的置換をブロック暗号とみなした場合、約 2^{64} の計算量で鍵回復攻撃が実行可能となる。なお、本攻撃は認証暗号としての安全性に影響を及ぼすものではない。

以上より、単一鍵設定では TinyJAMBU の安全性を脅かす攻撃は存在しないものの、関連鍵設定では現実的な計算量で偽造攻撃が実行できることに注意が必要である。

3.1.11. Xoodyak

Xoodyak は Xoodoo と呼ばれる暗号学的置換をプリミティブとして使用した Duplex 構造の認証暗号と Sponge 構造のハッシュ関数をサポートしている。

Xoodoo に対する最良の攻撃手法はゼロサム識別攻撃であり、仕様段数である 12 段に対して現実的な計算量での識別攻撃が実行可能である。ただし、本攻撃が Xoodyak の安全性に直接影響を及ぼすものではないことに注意が必要である。本攻撃を除き、仕様段数に対して実行可能な攻撃手法は報告されていないことを確認した。

設計者が主張する認証暗号の安全性は次のとおり。

IND-CPA		INT-CTXT	
オフライン計算量	オンライン計算量	オフライン計算量	オンライン計算量
128 ビット	160 ビット	128 ビット	160 ビット

認証暗号の安全性は Duplex 構造の安全性に依存するが、INT-CTXT に関して設計者の主張する安全性と Duplex 構造の安全性（オフライン計算で 128 ビット、オンライン計算で 64 ビット）に齟齬があることに注意が必要である。なお、現時点で設計者が主張する安全性を破る攻撃は存在しない。プリミティブとして 12 段中 6 段に簡略化した Xoodoo を考慮した場合、条件付きキューブ攻撃による鍵回復攻撃が実行可能である。現時点で本攻撃が最良の攻撃手法であるため、認証暗号が安全性マージンを十分に有していることを確認した。

また、設計者が主張するハッシュ関数の安全性は次のとおり。

衝突困難性	原像計算困難性	第 2 原像計算困難性
128 ビット	128 ビット	128 ビット

ハッシュ関数の安全性は Sponge 構造の安全性に依存する。設計者の主張する安全性と Sponge 構造の安全性に齟齬はなく、現時点でハッシュ関数に対する攻撃は報告されていない。

3.1.12. Tsudik's keymode

Tsudik's keymode はハッシュ関数をプリミティブとして使用した軽量メッセージ認証コードであり、ISO/IEC 29192-6 にて国際標準化されている。秘密鍵とメッセージの組み合わせをハッシュ関数への入力とし、ハッシュ関数の出力値における最下位 t ビットをタグとして出力する方式である。

Tsudik's keymode の安全性は使用するハッシュ関数の安全性に依存する。ハッシュ関数が伸長攻撃を許す場合には偽造攻撃が実行可能となるため、明らかな脆弱性を有している。マークル・ダンガード構成のハッシュ関数（例：SHA-256）では伸長攻撃が可能であり、これらのハッシュ関数を Tsudik's keymode で使用すべきではない。実際に、ISO/IEC 29192-6 では、ハッシュ関数が衝突困難性を有することに加え、伸長攻撃が不可能であることを要件として挙げている。

一方、ランダムオラクルからの強識別不可能性が証明できるようなハッシュ関数を使用する場合、Tsudik's keymode は証明可能安全性を有する方式となる。ISO/IEC 29192-6 には具体例として、ISO/IEC 29192-5 において軽量ハッシュ関数として国際標

準化されている PHOTON、SPONGENT、Lesamnta-LW の使用を推奨している。Lesamnta-LW はマークル・ダンガード構成のハッシュ関数であるが、提案者が Tsudik's keymode で利用した場合の擬似ランダム性を証明している。つまり、これら 3 方式の軽量ハッシュ関数はいずれも Tsudik's keymode での使用に適していることを確認した。

3.2. 実装性能評価

調査報告書概要は以下の通り。

ハードウェア実装性能の評価について、最も多く使用されたプラットフォームの一つに Xilinx 社 の Artix-7 がある。このプラットフォーム上で面積コストとスループット性能を図 3 にプロットした。

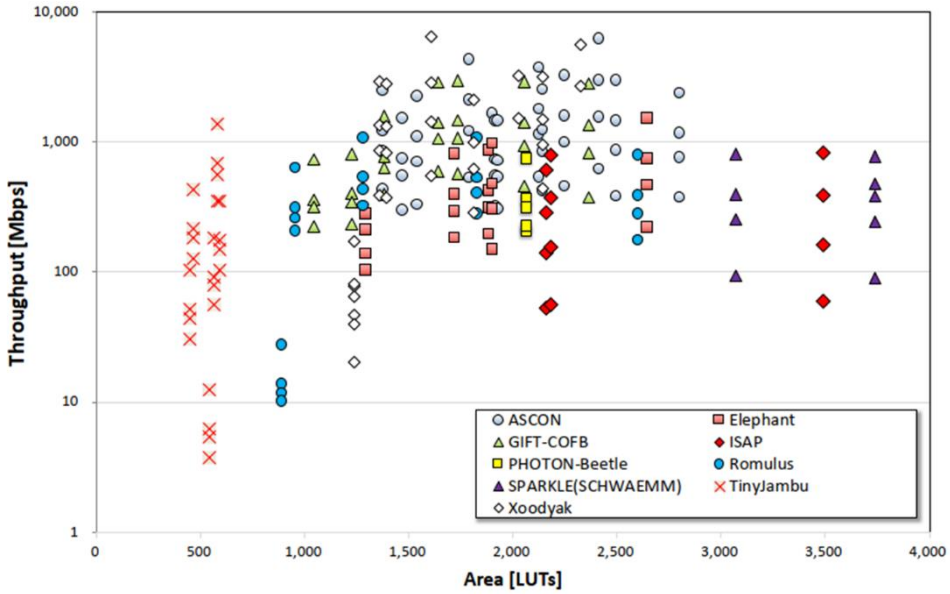


図 3 NIST LWC ファイナリストの Xilinx Artix-7 上の実装性能.
Grain128-AEAD は，今回の調査で結果が見つからなかったためプロットはない.

詳細の設定や測定条件の違いにより同じアルゴリズム同じ面積コストでも異なるスループット性能も存在している[6]。読み取れることとして、TinyJAMBU の回路面積が小さいこと、SPARKLE は比較的面積コストが大きくなることなどが挙げられる。また、ASCON、Elephant、GIFT-COFB、Romulus のプロットは広く分散していることから軽量実装から高速実装まで、FPGA 実装における設計の選択肢が多いアルゴリズムであるこ

[6] 例えば、同じプラットフォームであってもインタフェースの仕様や最適化の方針が異なることにより、面積コストや処理性能は大きな影響を受ける。また、暗号化/復号や AD の有無などによっても異なる。そのため、同じプラットフォーム上であっても、一つの方式について数値にばらつきが出る。

とが分かった。ただし、今後の研究の進展およびデータの公開により他のアルゴリズムについても同様の柔軟性がある可能性はある。

ソフトウェア実装性能の評価について、低リソースプラットフォーム上での実装性能評価は有用である。低リソースプラットフォームの一例として[1]では、Arm Cortex-M0 上のソフトウェア実装の結果が公開されている。これを表 1 に示す。

表 1 NIST LWC ファイナリストの Arm Cortex-M0 上の実装性能

Candidate	Latency [msec]	Code Size [kByte]
ASCON (ascon128v12)	0.633	29.4
Elephant (elephant160v1)	1069	14.4
GIFT-COFB (giftcofb128v1)	6.25	15.1
Grain-128-AEAD (grain128aead)	82.54	15.8
ISAP (isapk128av20)	161.9	14.5
Photon-Beetle (photonbeetleaead128rate128v1)	42.39	15.4
Romulus (romulusn1v12)	17.16	17.0
Sparkle (schwaemm128128v1)	0.76	14.9
TinyJambu (tinyjambu128)	0.394	13.7
Xoodyak (xoodyakv1)	0.599	14.3

表 1 ではレイテンシとコードサイズをまとめている。なお、括弧内の名称は、各アルゴリズムの Primary member のリファレンスソフトウェアである。RAM の使用量は、コンパイル時の静的なメモリサイズのレポートから、どのアルゴリズムも約 1kByte 程度であることが分かった。

レイテンシは、平文と AD(Associated Data)を 0 バイトから 32 バイトまで変化させた共通のテストベクトルを用いた際に、暗号化にかかった時間を平均した値で比較をしたところ、Elephant が最もレイテンシが高く、TinyJAMBU、Xoodyak、ASCON、SPARKLE が低レイテンシであることが分かった。コートサイズは、 ASCON が最も大きく、他の候補暗号方式には大きな差は見られなかった。この観測結果はある一面であり、プラットフォームに適した最適化実装を施すことにより、より良い結果が得られることは考慮の余地がある[7]。

他、報告書では、それぞれの暗号方式毎に、実装性能に関わる調査結果をまとめている。

[1] Ryota Hira, Tomoaki Kitahara, Daiki Miyahara, Yuko Hara-Azumi, Yang Li, Kazuo Sakiyama:Software Evaluation for Second Round Candidates in NIST Lightweight Cryptography. IACR Cryptol. ePrint Arch. 2022: 591 (2022)

[7] 一例として、ASCON について別のプラットフォーム上でコードサイズを最適化した実装などの結果も示されている。プラットフォームに即した最適化や最適化方針の違いによりコードサイズは異なる。

3.3. 標準化動向

調査報告書概要は以下の通り。

2016 年度版ガイドライン(2017 年 3 月発行)、CAESAR プロジェクト(2013 年～2019 年)、ISO/IEC の軽量暗号関連カテゴリ(2012 年～)、および NIST LWC ファイナリスト(2021 年 3 月)について、いずれにも掲載/規格化/選出されている方式はあまりないということが明らかとなった。このような現象が起きた要因としては、それぞれの選考時期のずれや対象カテゴリが完全一致していないなどが考えられる。例えば、ISO/IEC に採録された暗号方式や CAESAR プロジェクトに応募されていた暗号方式を基に改良された方式が NIST LWC に応募されているといった方式などがある。

2016 年度版ガイドラインに掲載されている SIMON と SPECK については、ISO/IEC の軽量暗号のカテゴリで議論されていたが、結果として軽量暗号としては ISO/IEC 標準規格として承認されず、自動認識・データキャプチャ技術に関する仕様で利用可能な軽量暗号方式として規格化されていること[8]が確認された。

評価指標に関して、安全性評価については、提案方式の設計根拠が十分に提示されない場合に第 3 者による評価が十分に行えないと判断され、評価対象から外されるなどの事例があった。実装性能評価については、従来論文ごとに異なる環境や測定シナリオで示されることが多くあったが、近年は AES-GCM や SHA-256 など広く世界で利用されているアルゴリズムとの比較などにより統一的な測定フレームワークを用いて実施することが一般化されてきていることが分かった。

4. 外部評価報告書に対する暗号技術評価委員会の見解

外部有識者による評価報告書は、今年度に目的としていた調査対象の暗号方式に対して、安全性・実装性能・標準化動向の調査として十分な内容を含んでいると考えられることから、本報告書を CRYPTREC の技術調査報告書とすることを承諾した。

---

[8] [SIMON] ISO/IEC (JTC 1/SC31) 29167-21:2018 Information technology - Automatic identification and data capture techniques - Part 21: Crypto suite SIMON security services for air interface communications .(2018 年 10 月出版)
[SPECK] ISO/IEC (JTC 1/SC31) 29167-21:2018 Information technology - Automatic identification and data capture techniques - Part 22: Crypto suite SPECK security services for air interface communications. (2018 年 11 月出版) (暗号技術評価委員会事務局調べ)

【参考】表 外部評価依頼先および依頼対象方式

NIST ファイナリスト			
認証暗号+ハッシュ関数	**安全性**	**実装性能**	**標準化動向など**
Photon-Beetle	岩田様	崎山様	菅野様
Sparkle	岩田様		
Ascon	藤堂様		
Xoodyak	内藤様		
認証暗号	**安全性**	**実装性能**	**標準化動向など**
Grain128-AEAD	藤堂様	崎山様	菅野様
TinyJambu	藤堂様		
Elephant	井上様		
ISAP	井上様		
Romulus	井上様		
GIFT-COFB	内藤様		
ISO 採択方式			
方式	**安全性**	**実装性能**	**標準化動向など**
Tsudik's keymode	岩田様		菅野様
Tsudik's keymode以外	2021年度調査済		

資料３－９

CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）及び CRYPTREC 暗号技術ガイドライン（高機能暗号）について

1. 背景

2020 年度第 2 回暗号技術検討会にて、暗号技術評価委員会の活動として、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）を設置すること、並びに、高機能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(高機能暗号）を設置することが承認された。

2021 年度及び 2022 年度に、各ワーキンググループで検討及び執筆を進め、『CRYPTREC 暗号技術ガイドライン （耐量子計算機暗号)』及び『CRYPTREC 暗号技術ガイドライン (高機能暗号)』の案を作成した。（資料３－４、資料３－５、資料３－６、資料３－７参照）

2. CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）及び CRYPTREC 暗号技術ガイドライン（高機能暗号）【承認事項】

各ワーキンググループで作成したガイドラインの案(資料３－５、資料３－７）を、『CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）』及び 『CRYPTREC 暗号技術ガイドライン（高機能暗号）』として NICT 及び IPA が策定・公開するにあたって、検討会として内容に問題がないことをご承認頂きたい。

資料４－１

2022 度　暗号技術活用委員会活動報告

1.　2022 年度の活動概要

1.1　活動目的

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必要な活動を行っている。

2022 年度は、CRYPTREC 暗号リスト改定に向け、利用実績に関する評価を行う。また、2021 年度に引き続き、暗号鍵管理ガイダンスを作成する。

1.2　活動概要

今年度の活動概要は以下の通りである。

(1) 利用実績に関する評価

2022 年度は CRYPTREC 暗号リストの改定が予定されており、その際、推奨候補暗号リストから電子政府推奨暗号リストへの昇格にあたって利用実績に基づいた選定が行われることが決まっている。

そこで、IPA が実施する「暗号アルゴリズムの利用実績に関する調査」による調査結果に基づき、2021 年度に承認された利用実績による選定基準の下で利用実績に関する評価を行う。

(2) 暗号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を目的として、2021 年度に取りまとめた作業の進め方に基づき、暗号鍵管理ガイダンス WG にて暗号鍵管理ガイダンスを作成する。

(3) 暗号利活用のために作成すべきガイダンス候補の検討

暗号利活用のために作成すべきガイダンス候補を検討し、今後の執筆に向けた準備を行う。

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、2022 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	横浜国立大学　教授
委員	上原　哲太郎	立命館大学　教授
委員	垣内　由梨香	マイクロソフト株式会社　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　教授
委員	佐藤　直之	SCSK 株式会社　シニアプロフェッショナルコンサルタント
委員	須賀　祐治	株式会社インターネットイニシアティブ　シニアエンジニア
委員	田村　裕子	日本銀行　企画役補佐
委員	手塚　悟	慶應義塾大学　教授
委員	寺村　亮一	株式会社イエラエセキュリティ　執行役員　兼　高度解析部　部長
委員	松本　泰	セコム株式会社　IS 研究所　顧問
委員	三澤　学	三菱電機株式会社　主席研究員
委員	満塩　尚史	デジタル庁　セキュリティアーキテクト
委員	山口　利恵	東京大学　特任准教授
委員	渡邊　創	国立研究開発法人産業技術総合研究所　副研究センター長

（2023 年 3 月 31 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	2022 年 7 月 7 日 ～ 7 月 15 日	● 2022 年度暗号鍵管理ガイダンス WG 活動計画について
第一回	2022 年 8 月 4 日	● 2022 年度暗号技術活用委員会活動計画について ● 暗号アルゴリズム利用実績調査の中間報告について ● 2022 年度暗号鍵管理ガイダンス WG 活動計画について ● 暗号鍵管理ガイダンス WG 進捗報告について ● 運用ガイドライン／ガイダンス候補について
第二回	2022 年 12 月 20 日	● 暗号アルゴリズム利用実績調査の最終報告について ● 電子政府推奨暗号リスト掲載への推薦候補案について ● 暗号鍵管理ガイダンス WG 進捗報告について ● 運用ガイドライン／ガイダンス候補について
第三回	2023 年 3 月 14 日	● 暗号鍵管理ガイダンス WG 活動報告 ● 運用ガイドライン／ガイダンス候補について ● 2022 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 2022 暗号技術活用委員会報告[1]を参照されたい。

2.1 利用実績に関する評価

IPA が実施した暗号アルゴリズム利用実績調査の結果、及び 2021 年度に承認された利用実績に基づく選定基準（選定ルール）に基づき、現在の推奨候補暗号リストに掲載のアルゴリズムのうち、電子政府推奨暗号リスト掲載への推薦候補案について検討・選定し、暗号技術検討会に推薦した。

【検討方針】

IPA が実施した暗号アルゴリズム利用実績調査では、現在の推奨候補暗号リストに掲載のアルゴリズムのうち、EdDSA のみアンケートによる利用実績調査の対象外であった。

このため、「EdDSA」以外の「推奨候補暗号リスト」に掲載の暗号アルゴリズムについては「利用実績調査（考慮項目①～⑥)」結果に基づいて判定し、「EdDSA」については「利用実態確認（考慮項目②～⑤)」結果に基づいて判定することとした。

考慮項目		選定目安
採用実績	以下のいずれかを満たす場合、昇格の検討対象に含める。なお、採用実績は、 ● 5 年ごとに実施予定の大規模アンケート調査による「**利用実績調査**」 ● 必要に応じて、事務局が（大規模アンケート調査によらずに）情報収集する「**利用実態確認**」 により確認するものとする。	
	① **利用実績調査**の結果、電子政府推奨暗号リストに掲載されている（同一カテゴリの）暗号技術の採用実績と遜色がないことが確認された場合	**電子政府推奨暗号リスト掲載の（同一カテゴリの）暗号技術の採用実績を同等以上の採用実績がある推奨候補暗号リスト掲載の暗号技術を昇格検討対象とする。**
	② **利用実績調査**又は**利用実態確認**の結果、**電子政府システムや重要インフラ等、日本の基幹システム**においてすでに利用されていることが確認された場合	**必要に応じて、利用実績調査に代わって、各府省庁等への照会を実施し、照会結果（クローズドな利用を含め）を基に昇格検討対象を選定する。**

[1] CRYPTREC Report 2022 暗号技術活用委員会報告, https://www.cryptrec.go.jp/promo_cmte.html

	利用実績調査又は**利用実態確認**の結果、③～⑤のいずれかが確認された場合： ③ **利用者が多い主要な汎用製品群の複数**に搭載されるなど、明らかに採用が進展していると判断された場合 ④ **利用者が多い主要なオープンソースソフトウェアの複数**に搭載されるなど、明らかに採用が進展していると判断された場合 ⑤ **利用者が多い主要なサービスやプロトコルの複数**で利用されるなど、明らかに採用が進展していると判断された場合	**「複数」「利用者が多い（主要な）」というキーワードの両方を十分に満たし、明らかな採用促進が確認された場合には、必要に応じて、昇格検討対象とする。** **※「複数」の意味は、必要条件として「2 個以上が必要」ということであって、「2 個以上あればよい」という十分条件としての意味ではないことに留意**
標準化実績	以下を満たす場合、昇格の検討対象に含める。	
	⑥ **利用実績調査**の結果、電子政府推奨暗号リストに掲載されている（同一カテゴリの）暗号技術の採用実績と遜色がないことが確認された場合	**電子政府推奨暗号リスト掲載の（同一カテゴリの）暗号技術の採用実績を同等以上の採用実績がある推奨候補暗号リスト掲載の暗号技術は昇格検討対象とする。**

【IPA が実施した暗号アルゴリズム利用実績調査[2]概要】

本調査では、電子政府推奨暗号リストに掲載する暗号アルゴリズムを選定するために、製品やシステム等で利用している暗号アルゴリズム名称を特定し、当該暗号アルゴリズムがどの程度利用されているか、またどの程度の標準化や規格化に採用されているかといった「暗号アルゴリズムの製品化、利用実績」を収集することを意図しております。また、政府機関に対する調査、国際標準規格・民間規格に対する調査、オープンソースソフトウェアでの利用実績調査も併せて実施しております。

本調査は株式会社野村総合研究所に委託して実施しております。

調査対象	調査実績
(A) 応募暗号アルゴリズムの応募者に対するアンケート調査	● 応募暗号アルゴリズムの応募会社全 8 社（16 アルゴリズム）から回答受領 ➢ アルゴリズム実績提供：11 アルゴリズム → 調査(C)(D)(E)の情報として活用 ➢ 製品実績提供：4 アルゴリズム → 調査(B)の情報として活用

[2] IPA, 暗号アルゴリズムの利用実績に関する調査へのご協力のお願い,
https://www.ipa.go.jp/security/ipg/crypt_usageperformance

(B) 暗号アルゴリズムを搭載している市販製品の販売会社への調査	● アンケート配布数： ➢ 14 業界団体 2,535 社 ➢ 個別コンタクト 102 社 ➢ 調査パネル 23,747 名 ⇓ ● アンケート回収数：合計 211 社 301 製品 ➢ 業界団体：2,535 社 → 65 社 114 製品 ➢ 個別：108 社 → 28 社 37 製品 ➢ (A)調査：4 社 4 製品 ➢ 調査パネル：23,747 名 → 146 製品 ⇓ ● アンケート有効回答数： ➢ 合計 82 社 128 製品（利用アルゴリズム不明 19 製品を含む） ⇓ ● <u>集計対象：合計 101 社 209 製品</u> ➢ アンケート有効回答：109 製品 ➢ 公開情報調査（補充調査）：41 社 100 製品
(C) 日本の政府機関等に対する調査	● アンケート集計数： 政府機関で利用されている <u>98 システム</u> ● 規格調査数：全 30 件 うち、暗号アルゴリズム記載ありは <u>17 件</u>
(D) 国際標準規格・民間規格等に対する調査	● 規格調査数：IPA が指定した 25 件 ⇓ ● <u>集計対象：合計 171 件</u> ➢ 別途追加分：146 件 ➢ EdDSA だけ独自調査実施
(E) オープンソースソフトウェアでの利用実績調査	● OSS 調査数：IPA が指定した 30 件 ⇓ ● <u>集計対象：合計 30 件</u> ➢ EdDSA だけ独自調査実施

【電子政府推奨暗号リスト掲載への推薦候補案】

● **エンティティ認証、ハッシュ関数、署名を除いた技術分類について：**

技術分類		推薦候補	推薦しない候補	理由
公開鍵暗号	鍵共有	該当なし	PSEC-KEM	● 他の鍵共有と比較して優位な利用実績があるとは認められない

共通鍵暗号	64 ビットブロック暗号	該当なし	CIPHERUNICORN-E Hierocrypt-L1 MISTY1	● 他の 64 ビットブロック暗号と比較して優位な利用実績があるとは認められない
	128 ビットブロック暗号	該当なし	CIPHERUNICORN-A CLEFIA Hierocrypt-3 SC2000	● 他の 128 ビットブロック暗号と比較して優位な利用実績があるとは認められない
	ストリーム暗号	該当なし	Enocoro-128v2 MUGI MULTI-S01	● 他のストリーム暗号と比較して優位な利用実績があるとは認められない
認証暗号		ChaCha20-Poly1305	該当なし	● 考慮項目①②④について、利用実績があると認められる
暗号利用モード	秘匿モード	XTS	該当なし	● 考慮項目①②④について、他の秘匿モードと比較して利用実績があると認められる
メッセージ認証コード		該当なし	PC-MAC-AES	● 他のメッセージ認証コードと比較して優位な利用実績があるとは認められない

- **エンティティ認証について：**

技術分類	推薦候補	推薦しない候補	理由
エンティティ認証	ISO/IEC 9798-4	該当なし	● 考慮項目①②において、他のエンティティ認証と比較して利用実績があると認められる

- **ハッシュ関数について：**

技術分類	推薦候補	推薦しない候補	理由
ハッシュ関数	SHA-512/256 SHA3-256 SHA3-384 SHA3-512 SHAKE128 SHAKE256	該当なし	● 考慮項目④において、他のハッシュ関数と比較して利用実績があると認められる

- **EdDSA について：**

技術分類		推薦候補	推薦しない候補	理由
公開鍵暗号	署名	EdDSA	該当なし	● 考慮項目④において、他の署名と比較して利用実績があると認められる

2.2 SC2000 の CRYPTREC 暗号リストからの取下げ申請への対応

富士通株式会社から取下げ申請があった SC2000 について、審議の結果、暗号技術検討会で CRYPTREC 暗号リストからの取下げルールを整備することを条件に、申請を了承することとした。

2.3　暗号鍵管理ガイダンスの作成

暗号鍵管理検討の初めとして、暗号鍵管理が必要なシステムの設計者向けに、暗号鍵管理の設計で明記する事項や考慮する点などを解説することを目的としたガイダンスである。本ガイダンスの位置づけと想定読者は以下の通りとする。

位置づけ

- 暗号鍵管理機能を持つシステム設計者のガイダンスを作成する。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）」を詳しく解説することを中心に作成する
- 暗号鍵管理で必要となる項目について、シンプルなモデルを例示し説明する
- シンプルなモデルを用いた説明においては、鍵管理における要求や思想が理解できるような記載を行う
- 暗号鍵管理における特に注意すべきリスクを説明する

想定読者

- 暗号鍵管理機能を持つシステム設計者

本ガイダンスは「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として書かれたものであり、中でも、下図において、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、各検討項目についての解説・考慮点を具体的に説明している。また、これらの理解を助けるため、簡単なシステム（トイモデル）を具体的に取り上げ、そのシステムで設定された構成や運用条件などを踏まえた場合の各々の検討項目における記載例を提供している。

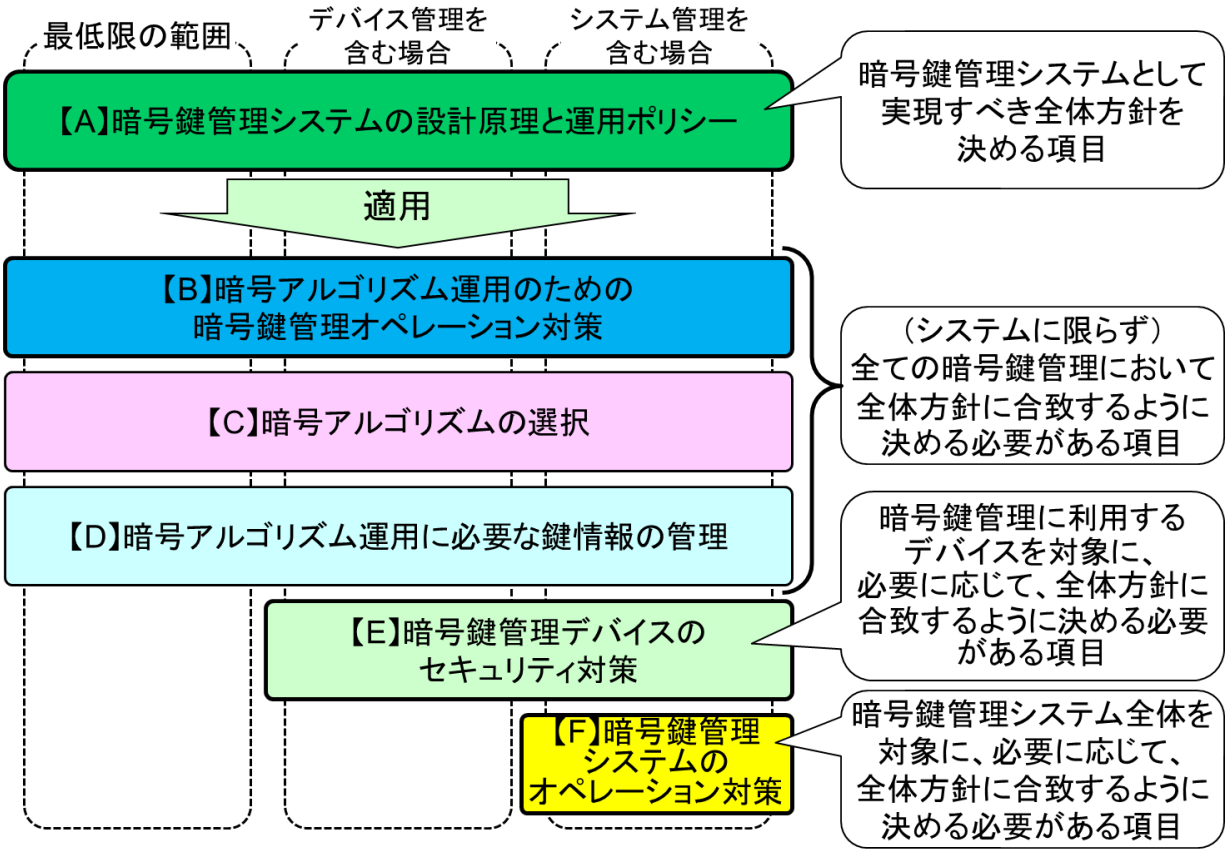


図　暗号鍵管理における目的別分類関係（「暗号鍵管理システム設計指針」より）

2022 年度版暗号鍵ガイダンスの章構成は以下のとおりである。

1. はじめに
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
3. 暗号アルゴリズムの選択
4. 暗号アルゴリズム運用に必要な鍵情報の管理

1 章では、イントロダクションとして、暗号鍵管理の重要性、及び本ガイダンスの位置づけについて説明している。

2 章は、暗号鍵管理システム設計指針（基本編）での「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」における検討項目についての解説・考慮点を記載している。具体的には、CKMS においてどのように暗号鍵が管理されるかを対象にしており、暗号鍵の生成から廃棄までのライフサイクル全期間にわたって暗号鍵を管理するのに必要となる機能や運用方法を取り決める検討項目を取り扱っている。例えば、CKMS をどのような設計方針の下でどのように構築されるのかの高レベルの概要を整理し、それ以降に決めなければならない各項目ではここで決めた内容に矛盾するような内容で定めてはならないことの重要性を指

摘している。このほか、使用している暗号鍵の状態や遷移条件、実施する処理に関する管理機能や、鍵情報の保管、鍵確立、鍵情報の喪失、破損、危殆化などが発生したときの BCP 対策に求められる項目での解説・考慮点を説明している。また、簡単なモデル（トイモデル）として S/MIME をモデルに取り上げ、記載例を示した。

3 章は、暗号鍵管理システム設計指針（基本編）での「暗号アルゴリズムの選択」における検討項目についての解説・考慮点を記載している。具体的には、暗号アルゴリズムや鍵長を選択に関する重要なポイントの解説、特に CRYPTREC 暗号リスト（電子政府推奨暗号リスト）、暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準や暗号鍵設定ガイダンスを参考に選択することを推奨している。また、Web ブラウザをクライアントとするクライアントーサーバシステムをモデルにした記載例を示した。

4 章は、暗号鍵管理システム設計指針（基本編）での「暗号アルゴリズム運用に必要な鍵情報の管理」における検討項目についての解説・考慮点を記載している。ここでは、3 章で決定した暗号アルゴリズムを運用するときに必要な鍵情報の管理の説明、具体的には管理すべき全ての鍵を明確にし、その鍵とメタデータの保護方法についての解説・考慮点を記載している。また、関係者がアクセス可能な Web サーバシステムをモデルにして記載例を示した。

2.4　暗号利活用のために作成すべきガイダンス候補の検討

2023 年度以降に作成すべきガイドライン／ガイダンスの候補について、以下の視点を踏まえ、検討を行った。今回の議論を踏まえ、どのガイドライン／ガイダンスを作成するかを 2023 年度活動計画に反映することを決定した。

- **ガイドライン／ガイダンスの整理学**
  - A. 以前から求められていた利用方法のガイドライン・ガイダンス：1)、2)、3)、8)、9)
  - B. 設定・設計に関するガイドライン・ガイダンス：4)、12)
  - C. 啓発ガイダンス：5)、6)、7)、15)
  - D. 新しい利用方法のガイドライン・ガイダンス：10)、11)
  - E. 少し早い（2023 年度着手ではなくてもいい）かもしれないガイドライン・ガイダンス：13)、14)
  - F. 他団体との連携で作るといいかもしれないガイドライン・ガイダンス：16)、17)、18)

- **検討にあたっての視点**
  - どのようなガイドライン／ガイダンスが求められているか
  - CRYPTREC がメインで作るのがよいか、それとも他の組織（IPA、業界団体など）がメインで／共同で作るのがよいか
  - 暗号技術の切り口メインで有用なガイドライン／ガイダンスになるか（暗号以外の部分がメインになったりしないか）

- **候補に挙がったガイドライン／ガイダンスのテーマ**

1	認証についてのガイダンス（特に二要素認証）
2	身元（本人）確認のためのガイダンス（例えば eKYC）
3	電子メールに関するガイドライン／ガイダンス
4	クラウドにおける鍵管理ガイダンス
5	組込機器の開発における、暗号プロトコル（例：認証プロトコル）のパラメータ選定基準
6	経営層も含めた人達を対象にした、暗号技術の啓発ドキュメント
7	暗号の使い方に関するガイドライン（ガイダンス）
8	PKI ガイドライン（ガイダンス）
9	暗号化消去
10	DNS の暗号に関わるガイドライン（ガイダンス）
11	暗号資産
12	e シール
13	API に関するガイドライン／ガイダンス
14	高機能暗号の標準化
15	耐量子計算機暗号のガイダンス
16	耐量子計算機暗号への移行に関するガイダンス
17	FIDO などの普及促進を促すガイダンス
18	リモート署名などの普及促進を促すガイダンス
19	暗号化消去などの普及促進を促すガイダンス
20	TLS 暗号設定ガイドラインのアップデート
21	運用ガイドラインやガイダンスに求められるニーズ／課題の整理

3. 今後に向けて

「暗号鍵管理システム設計指針（基本編）」の【A】【E】【F】に対する検討項目についても、引き続き、暗号鍵管理ガイダンス WG にて解説・考慮点を検討し、2022 年度版暗号鍵管理ガイダンスに追加していく予定である。

また、暗号利活用のために作成すべきガイダンス候補での議論を踏まえ、新たなガイドライン／ガイダンス作成に着手する予定である。

資料４－２

2022 年度　暗号鍵管理ガイダンス WG 活動報告

1.　2022 年度の活動内容と成果概要

1.1　活動内容

暗号鍵管理ガイダンス WG は、2021 年度に暗号鍵管理ガイダンスの作成するために活動を開始した。暗号鍵管理ガイダンスは、暗号鍵管理が必要なシステムの設計者向けに、暗号鍵管理の設計で明記する事項や考慮する点などを解説することを目的としたガイダンスである。

CRYPTREC では、情報システム設計者とシステム調達者が、暗号鍵管理システムを適切に扱うための支援を目的に活動をおこなっているが、本 WG では暗号鍵管理検討の初めとして、上記目的のため暗号鍵管理ガイダンスを作成している。

暗号鍵管理ガイダンスの位置づけと想定読者は以下の通りとする。

位置づけ

- 暗号鍵管理機能を持つシステム設計者のガイダンスを作成する。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）」を詳しく解説することを中心に作成する
- 暗号鍵管理で必要となる項目について、シンプルなモデルを例示し説明する
- シンプルなモデルを用いた説明においては、鍵管理における要求や思想が理解できるような記載を行う
- 暗号鍵管理における特に注意すべきリスクを説明する

想定読者

- 暗号鍵管理機能を持つシステム設計者

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、2022 年度に開催された暗号鍵管理ガイダンス WG での審議概要は表 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	漆嶌　賢二	GMO グローバルサイン株式会社　プロダクトマネジメント部 部長
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 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セキュリティ PF 開発課 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	西原　敏夫	シスコシステムズ合同会社　カスタマーエクスペリエンス シニアセキュリティアーキテクト
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	満塩　尚史	デジタル庁 戦略・組織グループ　セキュリティ危機管理チーム セキュリティアーキテクト

（2023 年 2 月 14 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	議案
第一回	2022 年 7 月 21 日	■ 2022 年度 WG 活動計画の確認 ■ 「暗号鍵管理システム（CKMS）の設計原理と運用ポリシー」の内容に関する審議 ■ 「暗号アルゴリズム運用に必要な鍵情報の管理」の内容に関する審議 ■ 「暗号鍵管理デバイスへのセキュリティ対策」の内容に関する審議
第二回	2022 年 11 月 29 日	■ 暗号鍵管理ガイダンス作成方針の変更 ■ 「暗号アルゴリズムの選択」に関する審議 ■ 「暗号アルゴリズム運用に必要な鍵情報の管理」に関する審議 ■ 「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」の内容に関する審議
第三回	2023 年 2 月 14 日	■ 「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」の記載内容について ■ 「暗号アルゴリズムの選択」の記載内容について ■ 「暗号アルゴリズム運用に必要な鍵情報の管理」の記載内容について ■ 「はじめに」の記載内容について

2.　成果概要

2.1　活動概要

本年度は表 1-2 のように WG を 3 回開催した。節ごとの記載内容について審議し、暗号鍵管理ガイダンスを作成した。

2.2　暗号鍵ガイダンスの構成の見直し

2021 年度 WG にて決定した執筆方針に基づき、想定読者を「暗号鍵管理機能を持つシステム設計者」「暗号鍵管理の参照プロファイル作成担当者」「暗号鍵管理プロファイルの利用者／暗号鍵管理機能を持つシステム調達者」とし、「暗号鍵管理プロファイルを作成するためのガイダンス」とすることを目的として、作業を実施していた。しかしながら、これらの想定読者の皆に対して理解してもらおうと多くの恩恵や効用を記載した結果、各節ごとの記載内容に大きなブレが生じ、逆に本来伝えるべき人に伝えるべき内容が伝わらない状態になったと判断した。

このため、想定読者の範囲を見直して、「暗号鍵管理システム設計指針（基本編）」と同様、「暗号鍵管理機能を持つシステム設計者」に絞り込んだうえで、システム設計者に伝えるべき内容に整理し、記載内容を修正することとした。この変更作業に伴い、暗号鍵ガイダンスの構成を見直し、2021 年度 WG にて決定した「暗号鍵管理システム設計指針（基本編）」に記載された章立てに沿った形での取りまとめではなく、関連性が強い部分をまとめた形に分割して取りまとめを行うこととした。具体的には、以下のように見直した。

なお、2022 年度に取りまとめを行わなかった部分については、2023 年度以降にも執筆を継続し、2022 年度版に補充していくものとする。

ガイダンス章構成（旧章構成）	見直し後
1. はじめに	2022 年度発行
2. 概要説明	（1 章に集約）
3. 暗号鍵管理システム（CKMS）の設計原理と運用ポリシー	2023 年度以降
4. 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	2022 年度発行
5. 暗号アルゴリズムの選択	2022 年度発行
6. 暗号アルゴリズム運用に必要な鍵情報の管理	2022 年度発行
7. 暗号鍵管理デバイスへのセキュリティ対策	2023 年度以降
8. 暗号鍵管理システム（CKMS）のオペレーション対策	2023 年度以降

2.3 2022 年度版暗号鍵ガイダンスの概要

本ガイダンスは「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として書かれたものであり、中でも、図 1-1 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、各検討項目についての解説・考慮点を具体的に説明している。また、これらの理解を助けるため、簡単なシステム（トイモデル）を具体的に取り上

げ、そのシステムで設定された構成や運用条件などを踏まえた場合の各々の検討項目における記載例を提供している。

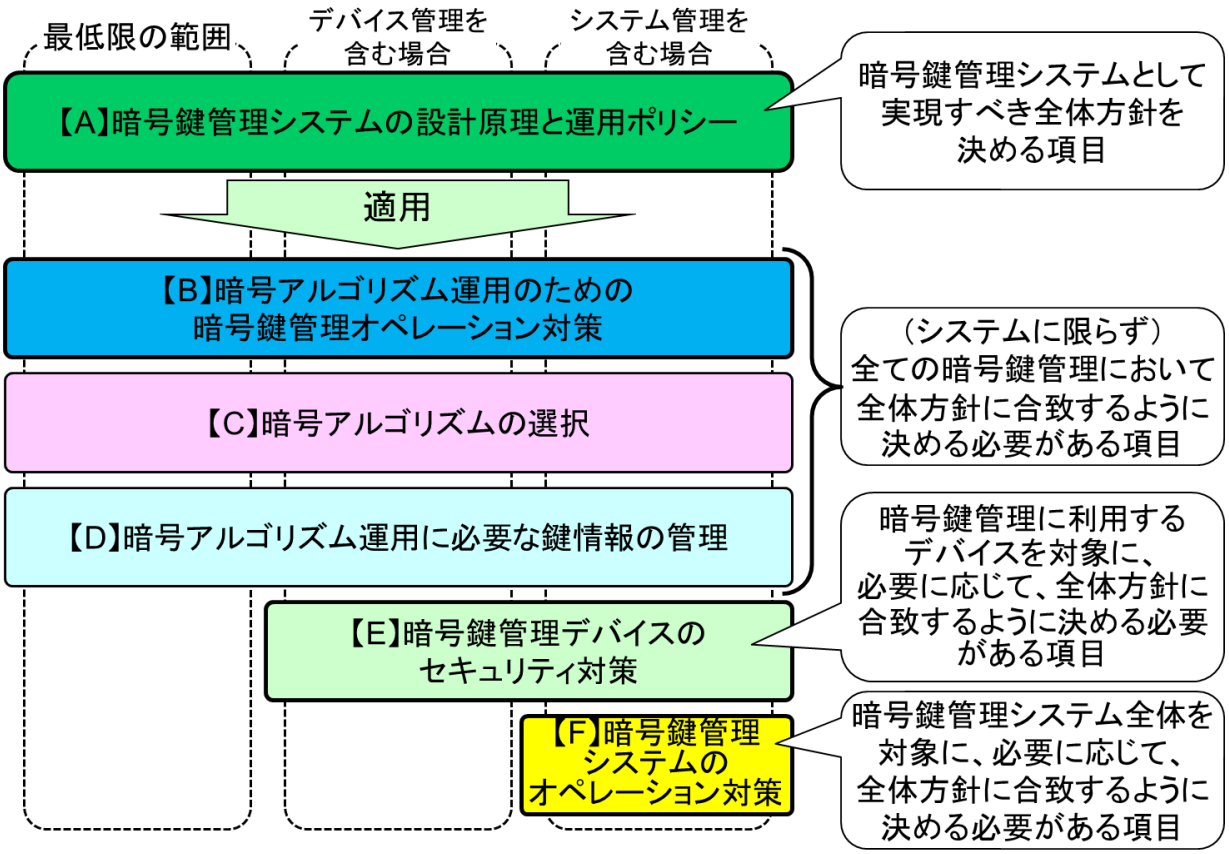


図 1-1　暗号鍵管理における目的別分類関係（「暗号鍵管理システム設計指針」より）

2022 年度版暗号鍵ガイダンスの章構成は以下のとおりである。

1. はじめに
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
3. 暗号アルゴリズムの選択
4. 暗号アルゴリズム運用に必要な鍵情報の管理

1 章では、イントロダクションとして、暗号鍵管理の重要性、及び本ガイダンスの位置づけについて説明している。

2 章は、暗号鍵管理システム設計指針（基本編）での「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」における検討項目についての解説・考慮点を記載している。具体的には、CKMS においてどのように暗号鍵が管理されるかを対象にしており、暗号鍵の生成から廃棄までのライフサイクル全期間にわたって暗号鍵を管理するのに必要となる

機能や運用方法を取り決める検討項目を取り扱っている。例えば、CKMS をどのような設計方針の下でどのように構築されるのかの高レベルの概要を整理し、それ以降に決めなければならない各項目ではここで決めた内容に矛盾するような内容で定めてはならないことの重要性を指摘している。このほか、使用している暗号鍵の状態や遷移条件、実施する処理に関する管理機能や、鍵情報の保管、鍵確立、鍵情報の喪失、破損、危殆化などが発生したときの BCP 対策に求められる項目での解説・考慮点を説明している。また、簡単なモデル（トイモデル）として S/MIME をモデルに取り上げ、記載例を示した。

3 章は、暗号鍵管理システム設計指針（基本編）での「暗号アルゴリズムの選択」における検討項目についての解説・考慮点を記載している。具体的には、暗号アルゴリズムや鍵長を選択に関する重要なポイントの解説、特に CRYPTREC 暗号リスト（電子政府推奨暗号リスト)、暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準や暗号鍵設定ガイダンスを参考に選択することを推奨している。また、Web ブラウザをクライアントとするクライアントーサーバシステムをモデルにした記載例を示した。

4 章は、暗号鍵管理システム設計指針（基本編）での「暗号アルゴリズム運用に必要な鍵情報の管理」における検討項目についての解説・考慮点を記載している。ここでは、3 章で決定した暗号アルゴリズムを運用するときに必要な鍵情報の管理の説明、具体的には管理すべき全ての鍵を明確にし、その鍵とメタデータの保護方法についての解説・考慮点を記載している。また、関係者がアクセス可能な Web サーバシステムをモデルにして記載例を示した。

3. 今後に向けて

2023 年度以降も暗号鍵管理ガイダンス WG にて、2022 年度に取りまとめを行わなかった残りの部分について検討・執筆を継続し、暗号鍵管理ガイダンスの内容を拡充していく。

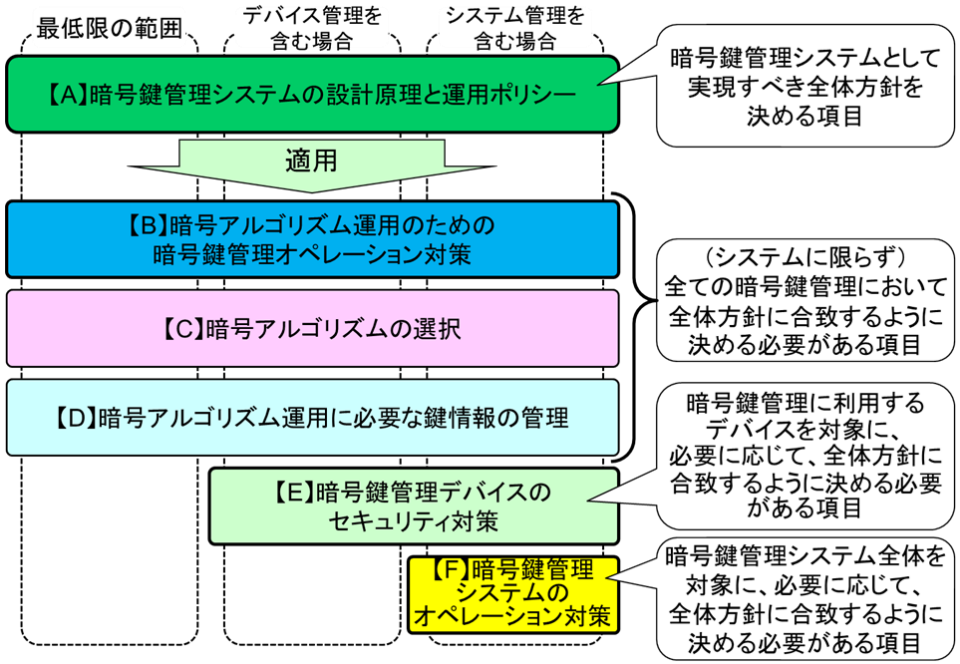
資料４－３

暗号鍵管理ガイダンス概要

1章 はじめに

1.1 位置づけ

- 「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
- CKMSの利用環境に関わらず検討する必要な項目のうちの【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



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）を具体的に取り上げ**、そのシステムで設定された構成や運用条件などを踏まえた場合の各々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**

2章 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

CKMS設計における、暗号鍵の生成から廃棄までのライフサイクル全期間にわたって暗号鍵を管理するのに必要となる機能や運用方法を取り決める

- 2.1 CKMS設計
- 2.2 暗号鍵のライフサイクル

} CKMS設計の全体的な考え方の整理

↓

- 2.3 暗号鍵のライフサイクル管理機能
- 2.4 鍵情報の保管方法
- 2.5 鍵情報の鍵確立方法

} 上記の考え方を実現するために必要となる機能要件や運用方法の明確化

- 2.6 鍵情報の喪失・破損時のBCP対策
- 2.7 鍵情報の危殆化時のBCP対策

} BCP対策（例外運用方法）の明確化

2章 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（2.1 KMCS設計／2.2 暗号鍵のライフサイクル）

2.1 CKMS設計
① 暗号鍵を提供するためにCKMSをどのように構築するかの概要

- CKMSを**どのような設計方針の下でどのように構築されるのかの高レベルの概要**整理
- 次節以降に決める必要がある事項を検討する際に**本概要で定めたことと矛盾していないことが確認できる程度に具体化した情報**のことであり、**次節以降では、本概要に記載したことに矛盾するような内容を定めてはならないことに注意**
- CKMS設計としてどこまでの範囲を対象とするのかを決め、それに応じて境界を定める

2.2 暗号鍵のライフサイクル
① 暗号鍵のライフサイクル全体にわたって取り得る鍵状態及び遷移条件の決定

- CKMS設計の観点からは、**どのような状態が存在し、どのような条件によって状態遷移が起きるのか**を明らかにする
- **本節で記載したことと次節での内容とが整合的であるようにしなければならないことに注意**

2章 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（2.3 暗号鍵のライフサイクル管理機能）

2.3 暗号鍵のライフサイクル管理機能
① 鍵情報に対する管理のために実行される機能の全体像

- 暗号鍵のライフサイクル管理機能として②以降で対象となる機能を全て明記することによって、**詳細を定めなければいけない項目に抜けが生じないようにする**
- ライフサイクル管理機能が暗号鍵のライフサイクルを実現するための手段であるので、**2.2節で記載したことと整合的であるようにしなければならない**
- 鍵情報が誤った使い方をされていないことを確認

② 鍵活性化機能への要求事項

- 活性化前状態から活性化状態への遷移を実現するための手順や遷移条件を具体化
- 2.2節で記載した活性化状態への遷移条件と整合的であるようにしなければならない

③ 暗号機能の実行場所の特定

- 暗号機能が実行されるところでは必ず暗号鍵が平文の形で使われることになるため、暗号鍵が保管される場所と並んで、もっとも暗号鍵の危殆化が発生しやすい場所
- したがって、暗号機能がある場所や実行場所を把握しておくことで、暗号鍵が狙われるリスクを低減させるために重点的に対策・保護すべき場所の絞り込みに活用できる

2章 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（2.3 暗号鍵のライフサイクル管理機能）

④ 鍵非活性化機能への要求事項

- どのような条件を満たしたときにどのような方法で非活性化状態に遷移するのかを決める
- CKMSセキュリティポリシーやCKMS設計などで最大許容暗号鍵有効期間が決まっている場合には、その最大期間と矛盾しないようにしなければならない
- システム上の要請などにより例外的に遷移条件の変更を容認する必要がある場合にはそのことを明確にしておくこと、例外がなし崩しにならないような明確な条件として定めておくことが重要

⑤ 鍵失効機能への要求事項

- 暗号鍵の漏えいなどによる予期せぬ原因により、当該暗号鍵の安全性が担保できないと判断された場合に実行
- どのようなことが発生したら危殆化状態と見なすのかといった遷移条件と誰にどのように危殆化状態に遷移した事実を知らせるのかといった通知方法などの要求事項を明確化
- 2.2節の危殆化状態への遷移条件と整合していなければならない

⑥ 暗号鍵の一時停止機能及び再活性化機能への要求事項

- どのようなことが発生したら一時停止させるのかといった一時停止状態への遷移条件と、どのようなことが満たされたら解除するのかといった活性化状態への遷移条件を必ずセットで準備
- 2.2節の一時停止状態への遷移条件及び再活性化の遷移条件と整合していなければならない

2章 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（2.3 暗号鍵のライフサイクル管理機能）

⑦ 鍵情報の破壊機能への要求事項

- 暗号鍵の漏えいリスクを完全に排除するためには、当該暗号鍵が完全に存在しなくなった状況にするしかない
- バックアップやアーカイブされたものも含めて完全に存在しない状況にする必要がある
- バックアップやアーカイブしている場合で鍵の破壊を行うケースでは、鍵の破壊に関してB.20の記載内容とB.54やB.56の記載内容が整合していなければならない
- 誤って暗号鍵の破壊をしてしまうことがないようにしておくことも重要

⑧ 鍵生成機能への要求事項

- 利用する暗号鍵が何らかの方法で予測できるのであれば、どんなに強力な暗号アルゴリズムを使っていたとしても安全性は担保されない
- 予測できない暗号鍵を生成する手段を利用することが極めて重要

⑨ 鍵導出機能／鍵更新機能への要求事項

- 暗号鍵の予測可能性の観点でいえば、鍵生成機能を利用して生成される暗号鍵よりも予測がしやすくなっている可能性がある
- 信頼できる鍵導出方法や鍵更新方法を使うことで暗号鍵の予測可能性のリスクを低減していることの確認を行う

2章 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（2.3 暗号鍵のライフサイクル管理機能）

⑩ 対称鍵の検証機能への要求事項
⑪ 公開鍵の検証機能への要求事項

- 生成されたり、共有されたりした暗号鍵が正当なものであることは、暗号アルゴリズムを利用する上での前提条件
- 確認を行うタイミングは、利用する暗号鍵の種類や利用方法、利用環境などによってさまざま

⑫ トラストアンカー管理機能への要求事項

- 信頼できる第三者であるトラストアンカーを用意し、そのトラストアンカーが「信頼できるかわからない公開鍵」にお墨付きを与えることで「信頼できる公開鍵」にする仕組みが必要
- トラストアンカーが公開鍵の信頼性の起点となることを意味し、その信頼性のうえで公開鍵暗号・署名の安全性が確保されている
- トラストアンカーの完全性確保は死活的に重要であり、そのための要求事項を明確化

⑬ 公開鍵の有効期間延長機能への要求事項

- 有効期間が経過すると非活性化状態に自動的に遷移して当該公開鍵は失効し、また新しい公開鍵が鍵生成機能を使って生成されるのが一般的
- CKMSセキュリティポリシー等で定める公開鍵の最大許容暗号鍵有効期間は、セキュリティ上の要件として定められたものであるので、これに違反するような延長は認められないことに注意

2章 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（2.3 暗号鍵のライフサイクル管理機能）

⑭ 所有者登録機能への要求事項

- 登録時点で誤った所有者登録が行われてしまうと、誤ったエンティティと暗号鍵が結び付くことになり、その後の暗号鍵のライフサイクルが正しく実行されたとしても全く安全性が担保されない

⑮ プライベート鍵所持の検証機能への要求事項
⑯ プライベート鍵の検証機能への要求事項

- トラストアンカーが「信頼できるかわからない公開鍵」を「信頼できる公開鍵」として利用できるお墨付きを与えるためには、その公開鍵に対応する「正しい」プライベート鍵（及びそれに付随するメタデータ）を「正当な」エンティティが所有していることを確認する必要がある
- プライベート鍵は基本的に唯一のエンティティのみが秘密に所持することが求められるため、当該公開鍵に対応する「正しい」プライベート鍵（及び付随メタデータ）を「正当な」エンティティが所有していることを、トラストアンカーにその中身を直接示して証明するわけにはいかない
- プライベート鍵（及び付随メタデータ）（と称するデータ）を所持していることを「正しい」エンティティが所有していることを確認するための検証方法の明確化

2章 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（2.3 暗号鍵のライフサイクル管理機能）

⑰ 暗号鍵とメタデータの関連付け機能への要求事項

- 鍵タイプによってはメタデータとの組み合わせによりその暗号鍵の完全性を確保する場合もあり、そのようなタイプの暗号鍵ではメタデータの管理も安全な鍵情報の管理を実施するために重要
- 関連付けを提供する保護メカニズムには、暗号学的プロセスを使用する場合と信頼プロセスを使用する場合とがある
- 4章に記載された暗号鍵とメタデータの関連付けを行う上で必要となる機能が用意されているか、整合的であるかの観点で確認することが重要

⑱ メタデータの変更機能への要求事項
⑲ メタデータの削除機能への要求事項
⑳ 暗号鍵のメタデータリスト化機能への要求事項

- メタデータの完全性に影響するような処理を行えるエンティティを制限し、かつ認可されたエンティティであっても許可された範囲内、例えば自分の管理下にあるメタデータに対してのみ変更や削除、参照が行えるようにしておくことが望ましい

2章 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（2.4 鍵情報の保管方法）

2.4 鍵情報の保管方法
① 保管中の鍵情報のセキュリティを確保するための手段の決定

- **「格納権限がある信頼できるエンティティ（利用者）」によって「正しい暗号鍵」が「正しく保管」される**ことが重要。「格納権限がある信頼できるエンティティ（利用者）」であることを確認するための認証認可機能、「正しい暗号鍵」であることを確認するための完全性検証機能、対称鍵やプライベート鍵では秘密裏に「正しく保管」するための機密性保護機能が求められる
- **利用権限を有する正しいエンティティ（利用者）のみが保管された暗号鍵にアクセスできることを保証**するためのアクセス制御機能・認証認可機能が求められる
- ストレージなどでの保管中も、誤操作や故障、改ざん攻撃などによって、暗号鍵の完全性が損なわれないように保護することが必要
- **クラウドサービスを利用する場合の鍵情報のセキュリティ確保については、クラウド事業者とクラウド利用者との責任分界点を認識し、それに応じて検討すべき範囲が変わる**ことを記載

2章 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（2.4 鍵情報の保管方法）

② 運用中の鍵情報の保管場所及び保護方法の決定

- 運用中の暗号鍵は、ストレージなどに保管された状態から読み込まれ、メモリ上などに保存された状態で暗号処理に使われるのが一般的。暗号処理のスループットを上げるため、暗号鍵が平文の形で置かれることが多く、暗号鍵の漏えいリスクが高い場所の一つになっている
- 暗号鍵の漏えいリスクの低減策として、運用中の暗号鍵がどこに存在し、どのように保護されているのかを把握しておくことが重要

③ 鍵情報のバックアップ方法の決定

- バックアップを行うことによるメリットとデメリットを天秤にかけて、暗号鍵のバックアップを行うかどうかを決定することが重要
- できる限り、暗号鍵の複製を作ることによる漏えいリスクを低減する対策を検討する必要があり、そのためには具体的なバックアップの条件や実施方法などを取りまとめておくことが重要
- バックアップした暗号鍵が使用されることがなくなったとき、復元できないように破壊されるべき

2章 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（2.4 鍵情報の保管方法）

④ 鍵情報のアーカイブ方法の決定

- バックアップもアーカイブも、暗号鍵の複製を作ることで漏えいリスクを高めることにつながるという点では同じであるが、バックアップが比較的頻繁に復元を行うことが想定されているのに対して、**アーカイブは長期保管用ストレージ設備などに保管され、限定的な条件下でのみ復元されることが想定**されている点が異なる
- アーカイブは、適用される法律や規則等も考慮して最小限の範囲で実施し、暗号鍵の複製を作ることによる漏えいリスクを低減する対策を検討する必要
- バックアップと比較して、復元頻度が少なく、長期保管が想定されることから、**可用性よりも機密性保護を優先**して対策を考えるべき
- アーカイブする必要がなくなれば、復元できないようにそのアーカイブは破壊されるべき
- アーカイブ鍵のほうが先に有効期間切れになった時に、機密性保護の継続性を確保するための要求事項を明確化

⑤ 鍵情報の復元方法の決定

- 厳格な復元ルールを規定し、そのルールが全て満たされていることを検証された後に、認可されたエンティティ（利用者）によって復元できるようにすべき
- 実現されるセキュリティ水準は、バックアップやアーカイブで実現されるセキュリティ水準と整合的であることが重要

2章 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（2.5 鍵情報の鍵確立方法）

2.5 鍵情報の鍵確立方法
① 鍵確立機能の利用局面の特定

- 鍵確立が行われるタイミングは、**「盗聴」という手段ーすなわち、鍵確立に係る正当なエンティティに検知されない手段ーで第三者が暗号鍵を窃取できる**唯一のタイミング
- 鍵確立の方法として、鍵配送と鍵合意がある。
- **鍵確立を行うとき、(i)確立される暗号鍵が誤りなく、正しく共有されること（暗号鍵の完全性）が求められるだけではなく、(ii)関係するエンティティが全員正当であることの確認と(iii)セキュアな通信路での通信による暗号鍵の機密性保護が極めて重要**

② 鍵配送における鍵情報のセキュリティを確保するための要求事項

- 全ての暗号鍵は完全性保護を、加えて対称鍵及びプライベート鍵は機密性保護も必要
- 機密性保護の観点からは、物理的保護が行われる手段か、対称鍵ラッピング鍵又はひとつ以上の非対称配送鍵ペアが関わる鍵配送手段を使用
- 完全性保護の観点からは、暗号鍵の送信者が信頼できることと、暗号鍵に改ざんやエラーがないことが求められる

2章 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（2.5 鍵情報の鍵確立方法）

③ 鍵合意における鍵情報のセキュリティを確保するための要求事項

- 不正なエンティティに合意鍵を窃取されないためには認可されていないエンティティが鍵合意プロセスに不正に入り込むことを防止することが絶対条件

④ 鍵確認機能を利用するための要求事項

- 実際の暗号処理で利用する前に共有された暗号鍵を鍵確認することにより、何らかの理由で誤った暗号鍵が生成されたり、暗号鍵がうまく共有できなかったりした場合であっても、実害が発生する前に当該暗号鍵の利用を止めることが可能

⑤ 利用する鍵確立プロトコルの決定

- セキュアなプロトコルであることが確認されたものだけを使うべき
- 「必ずしもセキュアとは言えないが相互接続性を実現するために必要」とされる手順や暗号アルゴリズムもデフォルトで選択できるようになっていることがあるので、意図せずに誤って利用することがないよう、CKMS設計の段階でデフォルトでは使えないように設定しておき、真に必要な場合には「例外として意図的に設定変更する」ようにすることが重要
- 安全性と相互接続性のバランスを踏まえた推奨の設定ガイダンスが存在する場合にはその設定に従うことで、セキュアな鍵確立プロトコルを確保できる

2章 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（2.6 鍵情報の喪失・破損時のBCP対策）

2.6 鍵情報の喪失・破損時のBCP対策
① 鍵情報の喪失・破損に対するBCP対策の決定

- 鍵情報（暗号鍵やメタデータ）が喪失又は破損した場合で、バックアップもアーカイブもされていなかった場合、当該暗号鍵で保護されているデータの喪失につながる可能性がある
- 重大な災害は、多数の運用中の鍵情報の喪失又は破損を一気に引き起こす可能性が高い
- 鍵情報の喪失や破損時のBCPを実現するためにどのような対策が必要かを検討し、その結果、**鍵情報のバックアップやアーカイブを行うこととした場合には、バックアップやアーカイブの方針をまず定める必要**がある
- ここで定めたことは**B.52～B.61（バックアップ方法／アーカイブ方法／復元方法）の上位規定として機能することから、B.52～B.61の内容はここでの内容に沿って設定されなければならず、また矛盾していないことを確認することが重要**

2章 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（2.7 鍵情報の危殆化時のBCP対策）

2.7 鍵情報の危殆化時のBCP対策
① 暗号鍵の危殆化に対するBCP対策の決定
② メタデータの危殆化に対するBCP対策の決定

- 暗号鍵が危殆化した場合には、利用を停止し、新しい暗号鍵に置き換えるとともに、すでに暗号処理（暗号化や署名生成）が行われた情報に対しては個別にその正当性の判断を行う
- 危殆化した暗号鍵の使われ方によっては、当該鍵で保護されたデータに対してだけでなく、**他の多くの暗号鍵についても危殆化を連鎖的に引き起こす可能性があることに留意**
- 暗号鍵が危殆化した場合にどの程度の他の暗号鍵に影響を与える可能性があるかを、暗号鍵ごとに把握しておき、さらにBCP対策として影響を受ける可能性がある暗号鍵の更新方法までを決めておく
- 危殆化の影響を小さくするために、使用するそれぞれの暗号鍵に対して適切な暗号鍵有効期間の設定や利用範囲の制限をすることで、暗号鍵の危殆化のリスクを低減することも重要

③ 役員・従業員によるセキュリティ危殆化に対するBCP対策の決定

- **権限必要最小限ルールの徹底が重要**であり、必要な人に必要な権限しか与えない、権限を悪用していないかを監査する、操作ログを隠蔽できないようにする、といった事前対策が重要
- 役員・従業員によるセキュリティ危殆化が発生した場合には、あらかじめ決められた情報セキュリティポリシー及びCKMS機能に基づいた回復手続きで対応・復旧することが重要

3章 暗号アルゴリズムの選択

CKMS設計では、要求される保護レベル（セキュリティ強度）を満たすように暗号アルゴリズムと鍵長を決定しなければならない

セキュリティ強度 → 暗号アルゴリズムと鍵長

3.1 暗号アルゴリズムのセキュリティ
① 要求される保護レベル（セキュリティ強度）に対応した暗号アルゴリズムの決定

- セキュリティ強度の決定では、扱う情報の資産価値、情報の機密性や完全性などのほか、該当システムの利用期間の終了年も重要な要因の一つ
- 具体的に必要なセキュリティ強度の決定にあたっては、「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」又は「暗号鍵設定ガイダンス」を参考
- 資産価値が高い情報と評価されると、その情報を扱うシステムではより強いセキュリティが求められることがある

4章 暗号アルゴリズム運用に必要な鍵情報の管理

CKMS設計において管理すべき暗号鍵を漏れなく洗い出し、(2章で用意される機能や運用方法等を適切に組み合わせて)それらの暗号鍵を安全に管理していることを明確にする
※ 鍵情報＝暗号鍵＋メタデータ

- 4.1 鍵情報の種類 ・・・ 暗号鍵とメタデータの説明
- 4.2 鍵情報の選択 ・・・ 暗号鍵の洗い出しと個々の暗号鍵に対する管理方法を明確化
- 4.3 鍵情報の保護方針 ・・・ 主にメタデータを対象とした保護方針が対象

4.1 鍵情報の種類

- 「暗号鍵管理システム設計指針(基本編)」に記載されている21種類の暗号鍵の鍵タイプを記載
- メタデータは、暗号鍵を適切に管理するために、その暗号鍵に関連付けられている情報。メタデータの典型的な要素として23種類を記載

4章 暗号アルゴリズム運用に必要な鍵情報の管理（4.2 鍵情報の選択）

4.2 鍵情報の選択

① CKMSが取り扱う全ての鍵タイプの利用用途及び生成手段、メタデータ、信頼関係、保護方針などの決定

- CKMS設計で**明示的に選択や管理する必要がある全ての鍵（タイプ）が対象**
- CKMS設計で明示的に選択や管理しておらず、プロトコルや製品仕様により**内部処理として自動的に生成・使用される暗号鍵は基本的には含まない**
- 製品やアプリケーション、システムが自動的に生成・使用される暗号鍵を**「ブラックボックスとして使っている」という認識を持つ**ことが重要
- 信頼性に確信が持てない暗号モジュールを使用している場合などは、可能であれば、内部の処理を調査し、暗号鍵の信頼性を確認することが望ましい
- 暗号鍵とメタデータが正しく関連付けられていることを保証するための方法を明確化し、利用用途や想定される脅威等を踏まえて必要なセキュリティ強度を提供する機能や方法を選択することが重要

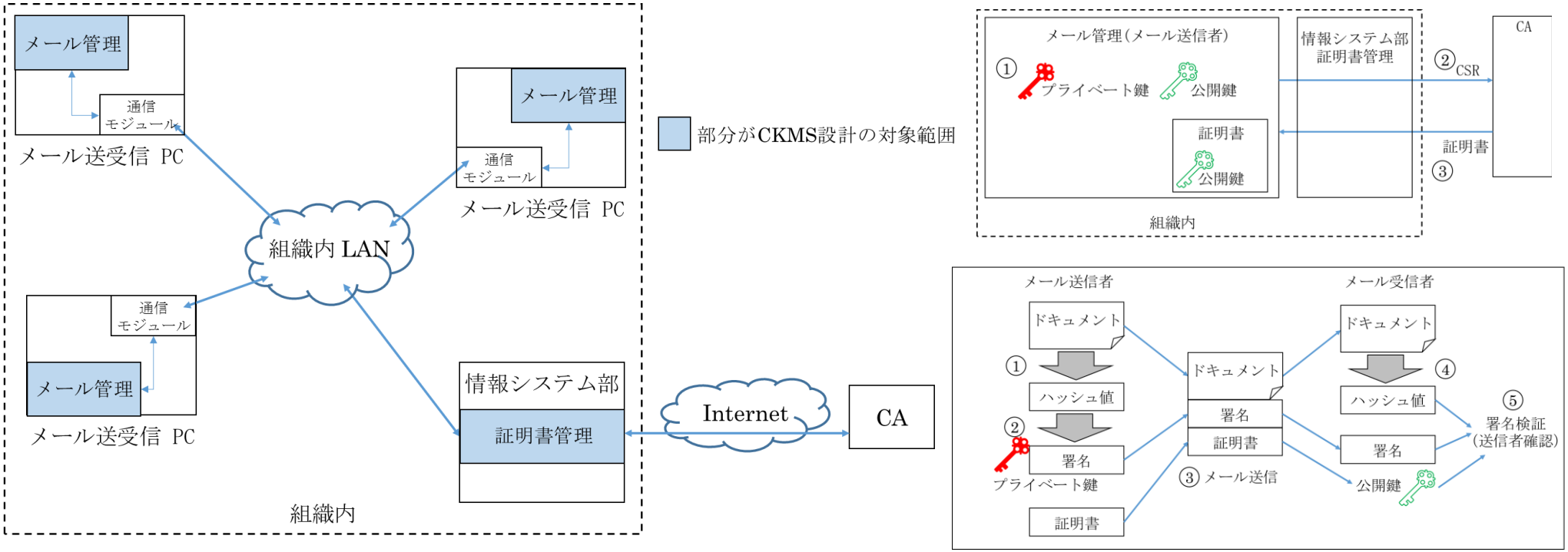
4章 暗号アルゴリズム運用に必要な鍵情報の管理（4.3 鍵情報の保護方針）

4.3 鍵情報の保護方針
① メタデータ要素内に含まれている情報の保護方法の決定

- 鍵タイプによってはメタデータとの組み合わせによりその暗号鍵の完全性を確保する場合もあり、そのようなタイプの暗号鍵ではメタデータの管理も安全な鍵情報の管理を実施するために重要
- 暗号鍵やメタデータ、信頼関係の保護方針を明確にすることで、鍵情報が安全に管理されていることを確認
- 保護手段として暗号メカニズムを使用している場合は、そこで利用している具体的な暗号アルゴリズムや鍵パラメタなどを明確化。信頼プロセスを使用している場合は、そこで利用している具体的なプロセスについての項目を明確化
- ここでの内容は、D.02、D.03での記載内容と整合していなければならない

2章でのトイモデル

■ メールの送信元認証をS/MIMEの署名付きメールで実現するシステム



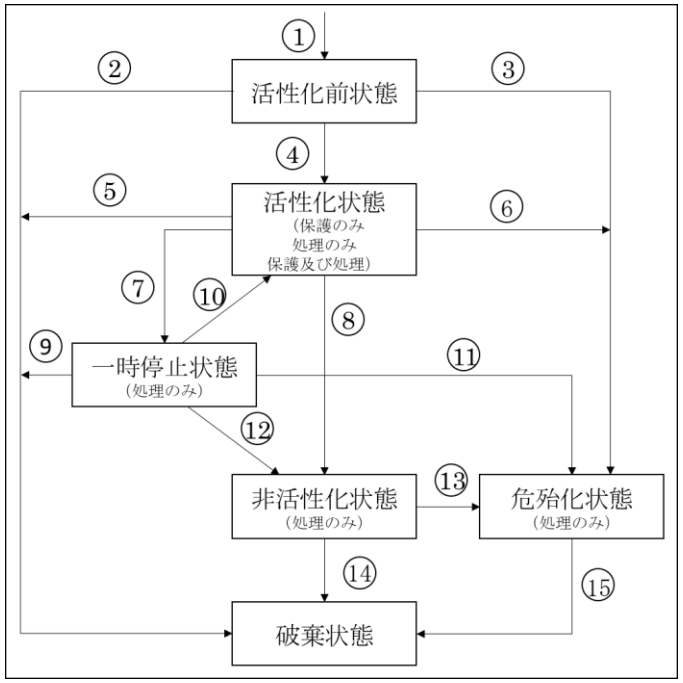
CKMS設計範囲

- メールの署名生成と署名検証の処理
- 署名生成や署名検証に使う暗号鍵の管理
- 公開鍵証明書の発行処理

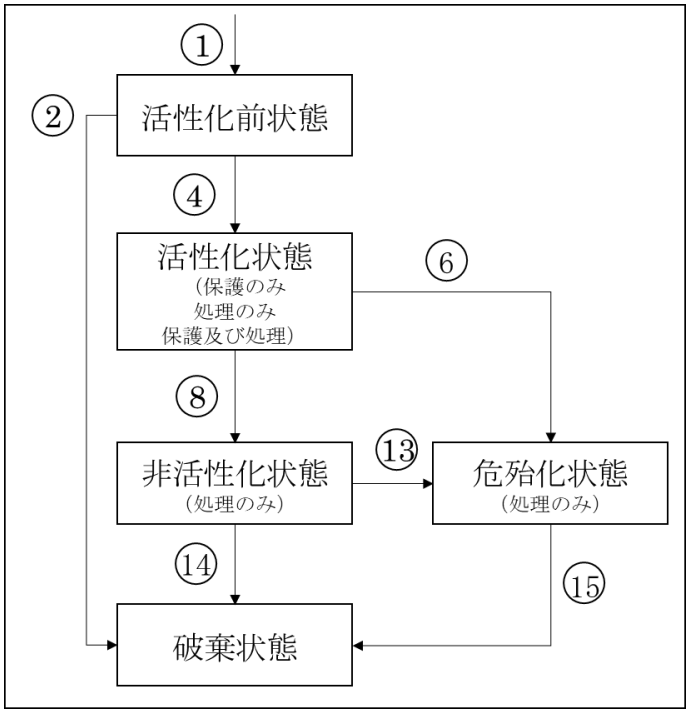
※メール送受信などを処理する通信モジュールは含まない

※CAは、外部のパブリックCAであり、CKMSには含まない

2章でのトイモデル



SP 800-57 Pt.1 Rev.5での
ライフサイクル



トイモデルでのライフサイクル

- 有効期間前の活性化前状態、有効期間内の活性化状態、有効期間後の非活性化状態に遷移(証明書の有効期限ベースに自動遷移)
- 実際に鍵が破壊されると破壊状態に遷移(手動)
- 署名プライベート鍵の危殆化が疑われる場合は危殆化状態に遷移
- 署名プライベート鍵が危殆化状態に遷移した場合、署名公開鍵も同時に危殆化状態に遷移
- 一時停止状態は設定しない
- これら以外の遷移条件は設定しない

2章でのトイモデル

【運用条件】

- 鍵生成はメール送信PCにおいて信頼できる方法で生成し、プライベート鍵はライフサイクル全般を通じてPC外部に複製されることはない。【B.04, B.22, B23】
- CAから公開鍵証明書を受信したときに、当該証明書の正当性を確認する。【B.27, B.28】
- 公開鍵証明書の有効期間が始まった鍵は自動的に活性化状態となる。【B.04, B.06, B.07】
- 署名の生成や検証は、メール送受信PCのメール管理部でのみ行う。【B.08】
- 署名付きメールを受信したとき、受信した公開鍵証明書の正当性を検証する。【B.04, B.27～B.29, B.36】
- 公開鍵証明書の有効期限切れした鍵は、自動的に非活性化状態になる。【B.04, B.09～B.11】
- 鍵の所有者が鍵を必要ないと判断したとき、手動で削除する。【B.04, B.20, B.21】
- 情報システム部を介してパブリックなCA局に署名を依頼することで、信頼できる証明書チェーンを構築し、基本的なトラストアンカー管理は更新機能などOSの機能より実現する。【B.04, B.05, B.29～B.32, B.37】
- 公開鍵証明書の運用管理は、情報システム部の担当者が行う。【B.12, B.33～B.35】
- 鍵の危殆化が疑われるときは失効処理を行う。【B.04, B.13, B.14】
- 鍵のバックアップ、アーカイブは行わない。【B.04】
- 鍵とメタデータの関連付けは公開鍵証明書により行う（暗号学的プロセス）。【B.38, B.39】
- メタデータの変更・削除・リスト化は認めない。【B.11, B.33, B.40～B.43】
- 鍵の一時停止状態は設定しない。【B.04, B.15～B.19】
- 鍵導出機能や鍵更新機能は使用しない。【B.04, B.24, B.25】
- 対象鍵は使用しない。【B.04, B.26】

2章でのトイモデル

【運用条件（続）】

- 鍵情報の機密性保護や完全性保護は、OSのアクセスコントロールシステム（ACS: Microsoft WindowsのACL、Linuxのパーミッション機能）により保護する。【B.44, B.46, B.48】
- ストレージに入力する鍵情報において、署名プライベート鍵は証明書の公開鍵とペアになっていることを確認し、メタデータは証明書の内容と一致することを確認する。【B.45】
- OSのACSにより、基本的に鍵を生成したユーザ以外、鍵情報にアクセスできない。【B.49, B.51】
- 暗号鍵のバックアップとアーカイブは実施しない。【B.52～B.61】
- 保管していた鍵が使用できなくなった場合の復元手段は用意しない。【B.50】
- 保管された鍵を保護するために鍵ラッピング鍵や鍵ペアは使用しない。【B.47】
- 署名プライベート鍵と署名公開鍵の有効期間は、公開鍵証明書の有効期間とする。【B.73】
- 署名プライベート鍵が危殆化すると署名公開鍵も同時に危殆化したとみなす。【B.74, B.75】
- 特に、危殆化しやすいと想定できるメタデータ要素はない。【B.76～B.78】
- 役割を分離して権限を必要な範囲にしている。【B.79】
- 個々のメール利用者が使用するPCにおいて、使用しているOSのログ保存機能により、署名プライベート鍵のアクセスログを管理する。アクセスログはシステム管理者しか確認できない。【B.79, B.80】
- 情報システム部の公開鍵証明書発行依頼の担当者の公開鍵証明書発行依頼ログと操作ログがPC内に保存され、操作ログへのアクセスは情報システム部の管理者しかできない。【B.79, B.80】
- 鍵の危殆化が疑われるときは失効処理を行い、鍵を再生成し、証明書を再発行する。【B.81】

2章でのトイモデル【記載例】

B.01	a) 利用するそれぞれの鍵タイプ 署名プライベート鍵、署名公開鍵 b) 鍵が生成される場所と手段 鍵の生成はメール送信者のPCで行われる c) それぞれの鍵タイプとの信頼関係で使用されるメタデータ要素（4.1節参照） 鍵の有効期間、親鍵（CAの署名公開鍵）、CAとの信頼関係 d) 鍵情報（暗号鍵やメタデータ）が存在しているそれぞれのエンティティのストレージにおける、鍵情報（暗号鍵やメタデータ）の保護方法 署名プライベート鍵はアクセスコントロールで保護される e) 配送時の鍵情報（暗号鍵やメタデータ）の保護方法 メール受信者に送信する公開鍵証明書はCAが署名している f) 鍵情報（暗号鍵やメタデータ）が配送され得る先となるエンティティの種類（例えば、ユーザ、ユーザデバイス、ネットワークデバイス） 公開鍵証明書はメールを送受信するユーザに配布
B.02	活性化前状態、活性化状態、危殆化状態、非活性化状態、破壊状態
B.03	● 署名プライベート鍵と署名公開鍵の鍵ペア生成後に活性化前状態に遷移（図2-6の①） ● 証明書に記載された有効期間の開始時に活性化前状態から活性化状態に遷移（図2-6の④） ● 活性化状態に遷移前に署名プライベート鍵に問題が生じた場合は、活性化前状態から破壊状態へ遷移（図2-6の②） ● 有効期限の終了時に活性化状態から非活性化状態に遷移（図2-6の⑧） ● 鍵の破壊条件を満たした場合に非活性化状態から破壊状態に遷移（図2-6の⑭） ● 活性化状態に遷移後に署名プライベート鍵の危殆化が疑われる事象が発生した場合は、活性化状態又は非活性化状態から危殆化状態に遷移（図2-6の⑥⑬）。署名公開鍵も同時に危殆化状態に遷移させる ● 証明書失効リスト（CRLリスト）に記載された署名公開鍵は危殆化状態に遷移（図2-6の⑥⑬） ● 危殆化処理完了時に破壊状態に遷移（図2-6の⑮） 注）本システムでは、図2-5の③⑤⑦⑨⑩⑪⑫の遷移は設定しない。

B.04	● メール送信者PCの鍵生成機能により、署名プライベート鍵と署名公開鍵を生成する。 ● 公開鍵証明書の有効期間が開始したら、鍵活性化機能により、署名プライベート鍵と署名公開鍵を活性化状態にする。 ● メール送信者PCの暗号機能により、メールのドキュメントに署名する。 ● メールを受信したら、メール送信者から送られてきたことを確認するために、メール受信者PCの暗号機能により署名の完全性を確認する。 ● 公開鍵の検証機能により、署名公開鍵に対する公開鍵証明書に対する完全性を確認し、公開鍵及びパラメタの検証を行う。 ● OSでのトラストアンカー管理機能により、ルートCAの公開鍵証明書を保管・管理する。情報の更新は、OS又はブラウザの自動アップデートにより実施する。 ● 公開鍵証明書の有効期間が終了したら、鍵非活性化機能により、署名プライベート鍵と署名公開鍵を非活性化状態にする。 ● 署名プライベート鍵と署名公開鍵は、破壊条件を満たした場合、破壊機能により、鍵を破壊する。 ● 署名プライベート鍵の危殆化が疑われるときは、鍵失効機能により、署名プライベート鍵の失効処理を行う。署名公開鍵についても同様の処理を行う。 ● 利用者の管理は情報システム部が行うものとし、そのために必要な管理機能は情報システム部管理の機器により実現する。 ● 暗号鍵とメタデータの検証及び関連付けについては、公開鍵証明書の申請段階で情報システム部がその正当性を検証するものとし、そのために必要な管理機能は情報システム部管理の機器により実現する。
B.05	● 署名公開鍵と関連メタデータの完全性は、公開鍵証明書のCA署名検証により確認する。 ● 署名プライベート鍵の機密性は、OSのファイルアクセス機能により当該鍵を作成したユーザ以外が鍵ファイルにアクセスできないように管理することで実現する。 ● CAの署名公開鍵はOSの信頼できる公開鍵証明書（トラストアンカー）からなるチェーンの有効性により完全性を確認する。 ● OSの信頼できる公開鍵証明書（トラストアンカー）の更新は、OS又はブラウザの自動アップデートにより実行される。

3章でのトイモデル

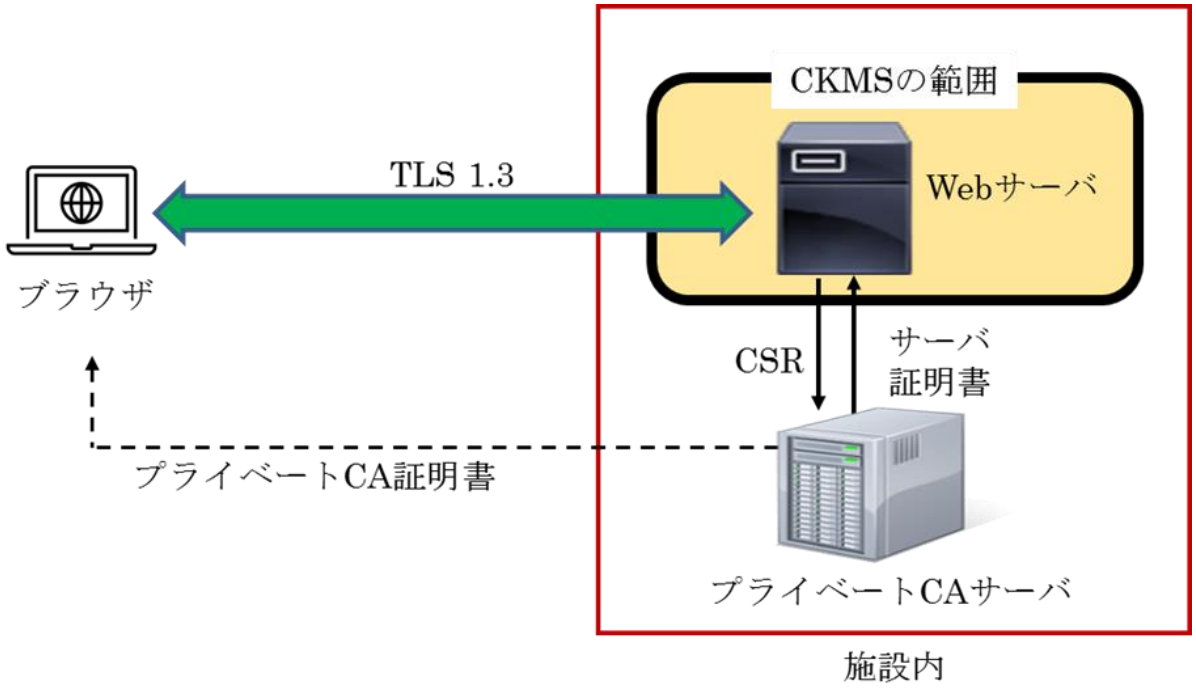
■ Webブラウザをクライアントとするクライアントーサーバシステム

【運用条件】

- 該当システムは2023年から8年間使用し2031年末破棄する予定 ⇒ 必要なセキュリティ強度は128ビット
- 「TLS暗号設定ガイドライン」の「高セキュリティ型での暗号スイート推奨設定（TLS1.3限定）」に従って設定
- サーバ証明書での署名方式はECDSA（P-256）、ハッシュ関数はSHA-256を使用

4章でのトイモデル

■ 関係者のみがアクセス可能なWebサーバシステム



【運用条件】

- Webサーバは、OSにLinux、Webサーバ機能を実現するソフトウェアにApache HTTP Server、TLS 1.3を実現するソフトウェアにOpenSSLを使用
- セキュリティプロトコルとしてTLS 1.3だけを許可
- サーバ証明書の署名方式はECDSA（P-256）を使用
- 鍵交換（鍵合意）のECDHはX25519, X448, P-256, P-386, P-521を使用
- プライベート鍵の保護のためにroot権限でプライベート鍵ファイルを保存（信頼プロセスを利用）
- 暗号学的プロセスはデジタル署名を利用して公開鍵やメタデータの完全性を保護

資料４－４

暗号鍵管理ガイダンス

（2023 年 3 月 24 日版）

目次

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、「政府機関の情報セキュリティ対策のための統一基準（令和 3 年度版）[1]」（令和 3 年 7 月 7 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 の通り、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いる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/kijyunr3.pdf

6.1.5 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、以下の措置を講ずること。

（略）

(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」を参照**した上で、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並びにそれを利用した安全なプロトコル及びその運用方法について、以下の事項を含めて定めること。

(ア) 職員等が暗号化及び電子署名に対して使用するアルゴリズム及びそれを利用した安全なプロトコルについて、**「電子政府推奨暗号リスト」に記載された暗号化及び電子署名のアルゴリズムが使用可能な場合には、それを使用**させること。

(イ) 情報システムの新規構築又は更新に伴い、暗号化又は電子署名を導入する場合には、やむを得ない場合を除き、**「電子政府推奨暗号リスト」に記載されたアルゴリズム及びそれを利用した安全なプロトコルを採用**すること。

(ウ) 暗号化及び電子署名に使用するアルゴリズムが危殆化した場合又はそれを利用した安全なプロトコルに脆弱性が確認された場合を想定した緊急対応手順を定めること。

(エ) **暗号化された情報の復号又は電子署名の付与に用いる鍵について、管理手順を定める**こと。

（以下、略）

図 1-1　政府機関の情報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]
- 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）」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 SP800-130「A Framework for Designing Cryptographic Key Management Systems」をベースに作成されている。

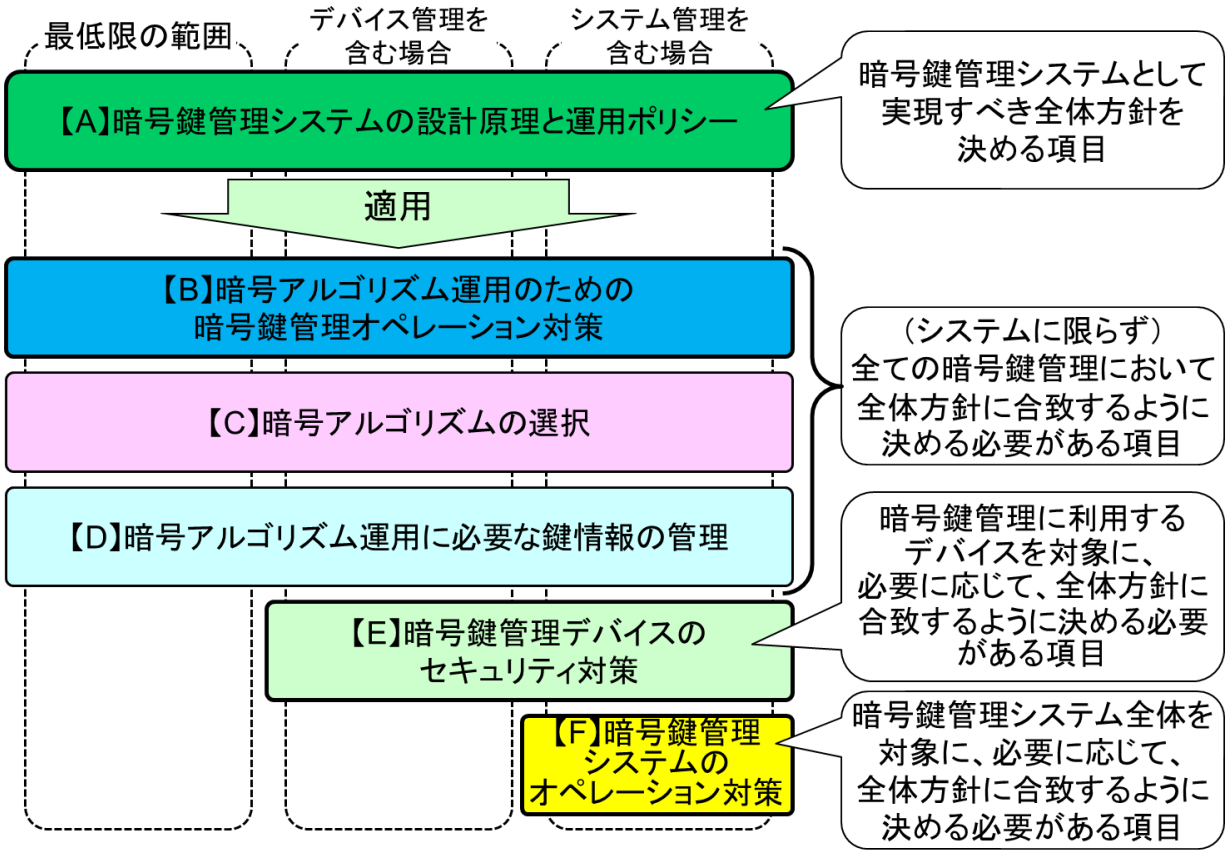


図 1-2　暗号鍵管理における目的別分類関係（「暗号鍵管理システム設計指針」より）

[2] https://www.cryptrec.go.jp/report/cryptrec-gl-3002-1.0.pdf
[3] https://www.cryptrec.go.jp/report/cryptrec-gl-3003-1.0.pdf
[4] https://www.cryptrec.go.jp/list/cryptrec-ls-0003-2022.pdf

本ガイダンスは、本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。中でも、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（ちなみに、【E】や【F】までを含む場合、「広義」の意味での暗号鍵管理を称す）。

1.2 想定読者

暗号鍵管理システム設計指針（基本編）での「暗号鍵管理についての技術的内容」での想定読者と同様、主として CKMS 設計者を想定読者としている。

1.3 構成

本設計指針は、4 章で構成されており、章立ては以下のとおりである。

2 章は「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」における項目についての解説・考慮点を示す。さらに、これらの理解を助けるため、簡単なシステム（トイモデル）を具体的に取り上げ、そのシステムで設定された構成や運用条件などを踏まえた場合の各々の項目における記載例を示す。

【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。

同様に、3 章では「暗号アルゴリズムの選択」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。

4 章では、「暗号アルゴリズム運用に必要な鍵情報の管理」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、本ガイダンスは、暗号鍵管理システム設計指針（基本編）と合わ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[5]」を利用する際に有用である。

[5] IPA, https://www.ipa.go.jp/security/vuln/ckms.html

1.4　検討体制

本ガイダンスは、2021 年度及び 2022 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（2023 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	漆嶌　賢二	GMO グローバルサイン株式会社　プロダクトマネジメント部 部長
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 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セキュリティ PF 開発課 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	西原　敏夫	シスコシステムズ合同会社　カスタマーエクスペリエンス シニアセキュリティアーキテクト
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	満塩　尚史	デジタル庁 戦略・組織グループ　セキュリティ危機管理チーム セキュリティアーキテクト

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

本章の目的・趣旨

本章は、設計指針（基本編）の 5 章に記載されている要求事項（各節での色付き枠内で示している内容）について解説したものである。

CKMS においてどのように暗号鍵が管理されるかを記載するものであり、「狭義」の意味での暗号鍵管理に相当するものである。暗号鍵の生成から廃棄までのライフサイクル全期間にわたって暗号鍵を管理するのに必要となる機能や運用方法を取り決める項目（B.01～B.81）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。

- 各暗号鍵は、どのような処理を目的として使われるのか。
- それら暗号鍵の保管場所や保管方法
- 各暗号鍵の生成から廃棄までのライフサイクルを通し、暗号鍵がどのように管理されるか。またその管理を行う上で必要となる機能群はどのようなものか。

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。

2.1 CKMS 設計

① 暗号鍵を提供するために CKMS をどのように構築するかの概要

項目	FR 番号	Framework Requirements の内容	SP800-130
B.01	FR2.4	CKMS 設計は、以下を含む CKMS システムの高レベルの概要を明記しなければならない： a) 利用するそれぞれの鍵タイプ b) 鍵が生成される場所と手段 c) それぞれの鍵タイプとの信頼関係で使用されるメタデータ要素（7.1 節表 7 2 参照） d) 鍵情報（暗号鍵やメタデータ）が存在しているそれぞれのエンティティのストレージにおける、鍵情報（暗号鍵やメタデータ）の保護方法 e) 配送時の鍵情報（暗号鍵やメタデータ）の保護方法 f) 鍵情報（暗号鍵やメタデータ）が配送され得る先となるエンティティの種類（例えば、ユーザ、ユーザデバイス、ネットワークデバイス）	2.5 節

解説・考慮点

項目 B.01 は、CKMS の設計にあたって、暗号鍵を提供するために CKMS をどのように構築するかの概要を明確化することを要求したものである。ここでは、機微なデータを保護するための暗号鍵に対する設計方針や実現目標、詳細を決める文書へのインデックス等を簡潔な概要で明記することが求められる。

本節で要求していることは、CKMS をどのような設計方針の下でどのように構築されるのかの高レベルの概要を整理し、明らかにしておくことである。ここでの「高レベルの概要」の意味は、次節以降に決める必要がある事項を検討する際に本概要で定めたことと矛盾していないことが確認できる程度に具体化した情報、ということである。

CKMS 設定の詳細については次節以降で取り扱うことになるので、ここであまり詳細に定める必要はない。

重要なことは、安全な CKMS 設計の第一歩として、B.01 において少なくとも以下のような事項を検討し、明らかにしておくことで、次節以降で詳細を定めなければいけない項目に抜けが生じないようにすることである。また、ここでの概要は CKMS の基本設定となるものであるので、次節以降では、本概要に記載したことに矛盾するような内容を定めてはならないことに注意する必要がある。

- a）と c）では、「どのような種類」の暗号鍵（及びそれに付随するメタデータ）を保護管理対象とする必要があるのかの把握
- b）では、暗号鍵が「どのように生成」されるのかの概要把握
- d）と e）では、暗号鍵が「どのような場所でどのように保護」されるのかの概要把握
- f）では、暗号鍵を「誰」が所持したり利用したりするのかの特定

その際、注意する必要があるのは、CKMS 設計としてどこまでの範囲を対象とするのかであり、それに応じて境界を定めることである。この範囲は、CKMS が取り扱う目的や設置場所、利用するエンティティやアプリケーション、プロトコルなどにより異なる。

例えば、図 2-1 のようなシステム（System A、System B、System C）間で、暗号鍵管理を含む暗号処理全体を担う各々の CKMS モジュールを経由して暗号通信をするプロトコルを利用するシステムを対象とした場合、CKMS 設計の対象範囲はシステム内での全ての CKMS モジュール（図 2-1 では CKMS Module A、CKMS Module B、CKMS Module C）の和集合、及び個々の CKMS モジュールと連携して動作する機器類やそのモジュールを利用するエンティティなどから構成される。一方、System A 内部に閉じたデータ保管を対象とした場合には、CKMS 設計の対象範囲は System A の CKMS モジュール（図 2-1 では CKMS Module A）、及び当該 CKMS モジュールと連携して動作する機器類やそのモジュールを利用するエンティティなどだけとなる。

この範囲を正しく定めておかなければ、本来管理対象とすべき暗号鍵が対象から漏れ、必要な対策が取られなかったり、逆に、不必要に多くの暗号鍵が管理対象となり、無駄な対策を実施することで余分なコストがかかったり利便性が低下したり、といった問題が生じることとなることに留意されたい。

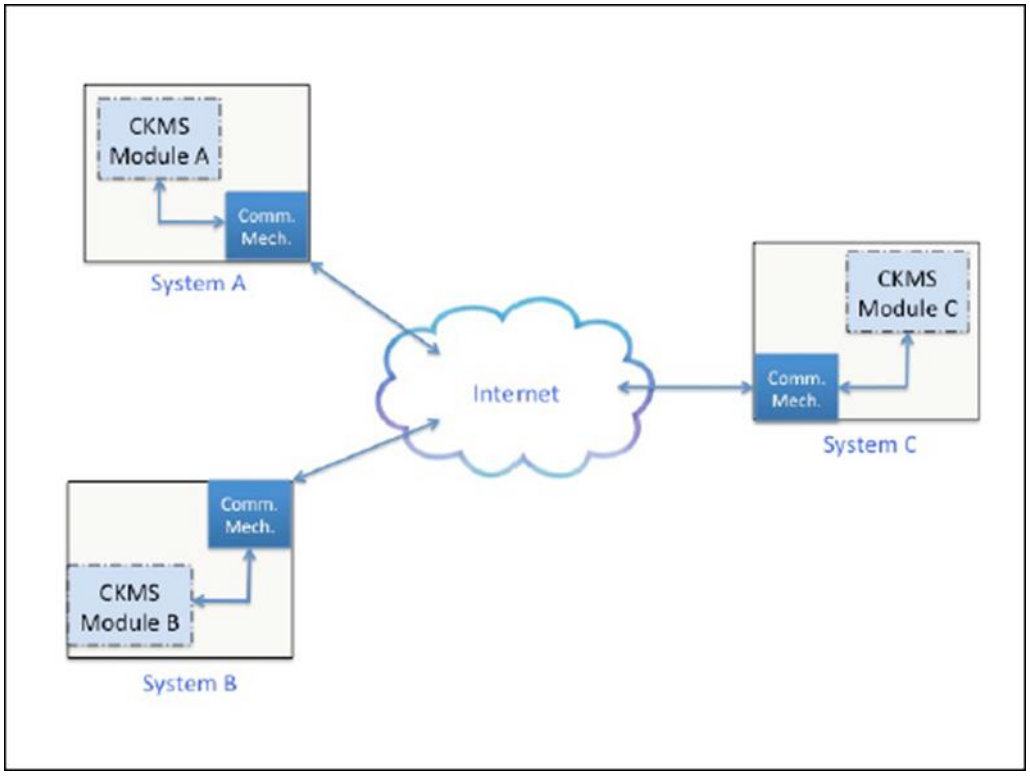


図 2-1　CKMS 概要例（NIST SP 800-130 より）

《トイモデルと記載例》

本節のトイモデルは、図 2-2～図 2-4 に示す、メールの送信元認証を S/MIME の署名付きメールで実現するシステムとする。

このシステムの CKMS 設計範囲は、図 2-2 の通り、メールの署名生成と署名検証の処理、署名生成や署名検証に使う暗号鍵の管理、公開鍵証明書の申請処理である。メールの送受信などを処理する通信モジュールは含まない。また、CA は外部のパブリック CA であり、これも CKMS には含まない。

準備として、図 2-3 の通り、メール送信者は、メール管理部において、信頼できる手段で署名プライベート鍵と署名公開鍵のペアを作成する。次いで、その署名公開鍵の CSR（証明書署名要求; Certificate Signing Request）を作成し、その CSR を組織内の情報システム部に送る。情報システム部は、証明書管理部において、CSR を確認し、確認後に CSR をパブリック CA に送り、CA が署名した公開鍵証明書を情報システム部が受信し、確認後にその公開鍵証明書をメール送信者に送信する。なお、CKMS の対象範囲はメール送受信 PC のメール管理部と情報システム部の証明書管理部だけであるので、図 2-3 での色が塗られた部分が該当する。

メールの署名と検証は図 2-4 の通りであり、メール送信者は、メール送信 PC のメール管理部において、送信するドキュメントのハッシュ値を計算し、そのハッシュ値に自分の署名プライベート鍵を使って署名し、ドキュメント、署名、自分の署名公開鍵の公開鍵証明書をメールで送信する。

メール受信者は、メール受信 PC のメール管理部において、(a)メール送信者の公開鍵証明書が有効であることを CA 証明書を使って確認し、(b)受信したメールからドキュメントを取り出し、ハッシュ値を計算し、その値とメール送信者の公開鍵証明書から取り出した公開鍵で署名が正しいか検証し、メール送信者から送られたメールであることを確認する。

なお、CKMS の対象範囲はメール送受信 PC のメール管理部となるので、図 2-4 での色が塗られた部分が該当する。また、パブリック CA は CKMS の対象範囲外としているので、CA が使う署名プライベート鍵や対応する署名公開鍵は CA が適正に管理していることを前提とし、その前提に問題がないことだけを CA 証明書によって確認する。

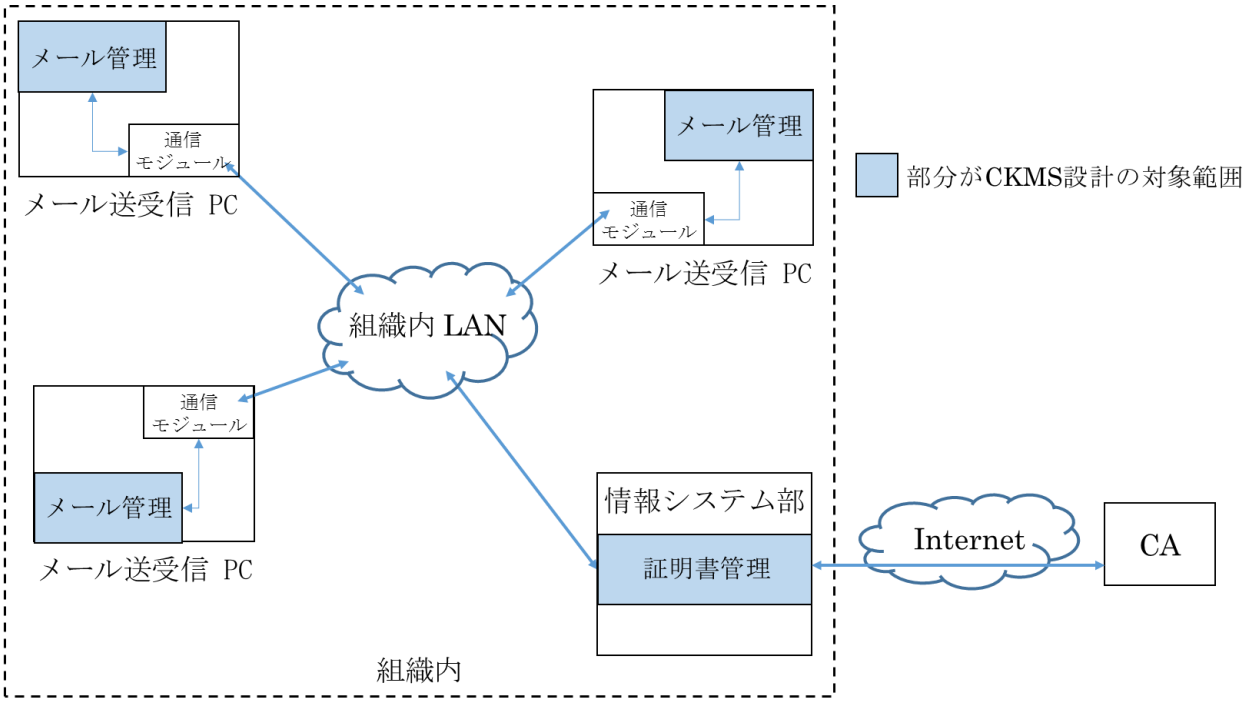


図 2-2　トイモデルでの CKMS 概要図

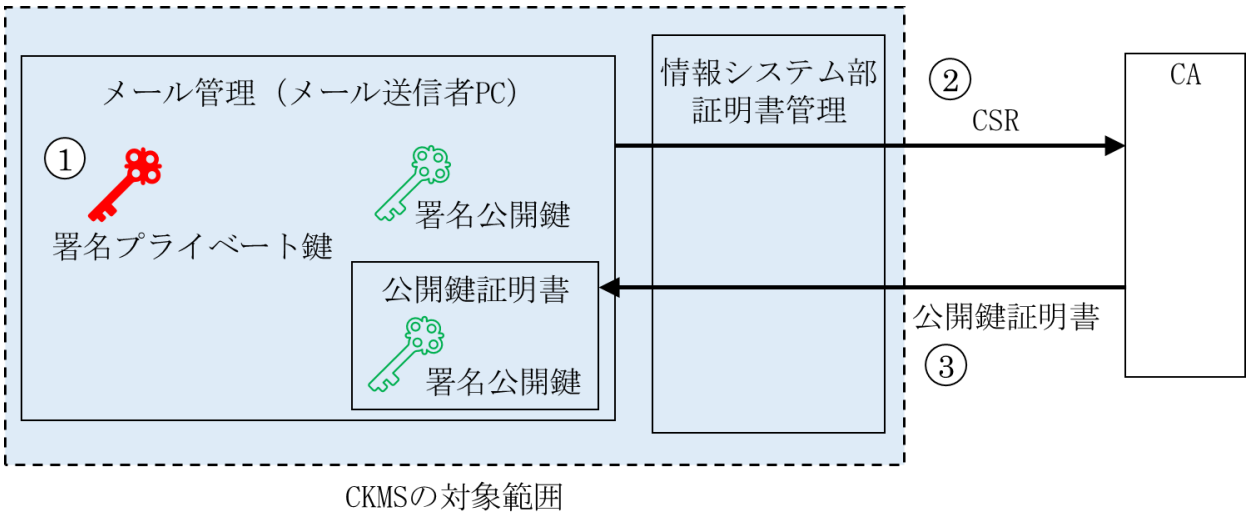


図 2-3　鍵生成、CSR 送信、証明書受信

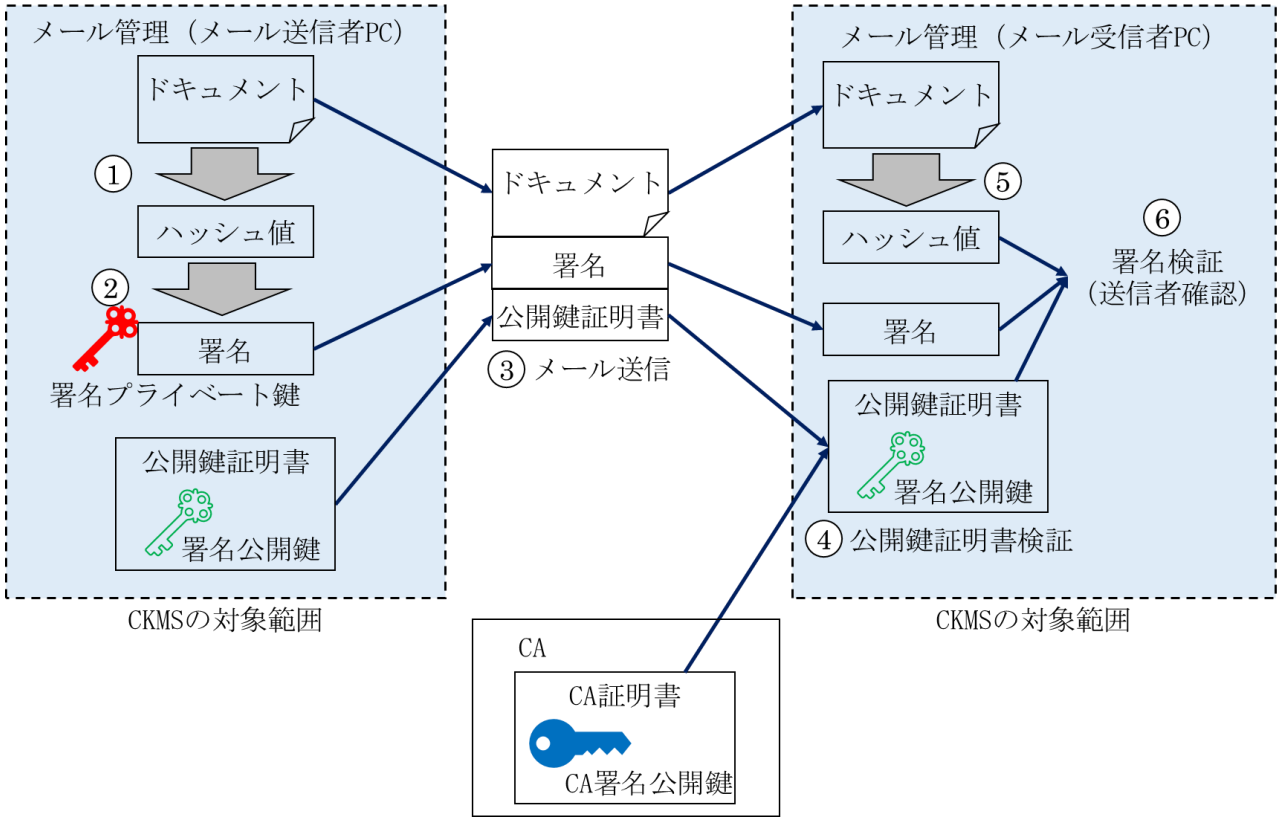


図 2-4　メールの署名と検証

以上のトイモデルにおける記載例は、以下の「署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例」のようになる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.01	a)　利用するそれぞれの鍵タイプ 署名プライベート鍵、署名公開鍵 b)　鍵が生成される場所と手段 鍵生成はメール送信者の PC で行われる c)　それぞれの鍵タイプとの信頼関係で使用されるメタデータ要素（4.1 節参照） 鍵の有効期間、親鍵（CA の署名公開鍵）、CA との信頼関係 d)　鍵情報（暗号鍵やメタデータ）が存在しているそれぞれのエンティティのストレージにおける、鍵情報（暗号鍵やメタデータ）の保護方法 署名プライベート鍵はアクセスコントロールで保護される e)　配送時の鍵情報（暗号鍵やメタデータ）の保護方法 メール受信者に送信する公開鍵証明書は CA が署名している f)　鍵情報（暗号鍵やメタデータ）が配送され得る先となるエンティティの種類（例えば、ユーザ、ユーザデバイス、ネットワークデバイス） 公開鍵証明書はメールを送受信するユーザに配布

2.2　暗号鍵のライフサイクル

①　暗号鍵のライフサイクル全体にわたって取り得る鍵状態及び遷移条件の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.02	FR6.15	CKMS 設計は、CKMS の鍵が取り得る全ての状態を明記しなければならない。	6.3 節
B.03	FR6.16	CKMS 設計は、全ての CKMS 鍵状態間の遷移、及び遷移を起こすことに関係するデータ（入力と出力）を明記しなければならない。	6.3 節

解説・考慮点

暗号鍵のライフサイクルの一般形は、SP 800-57 part1 の 7 節「鍵状態と遷移（Key States and Transitions）」に基づく。これをベースに、CKMS とそのアプリケーションに適切な鍵状態と遷移条件を選択し定義する。
項目 B.02 及び B.03 は、CKMS の設計にあたって、暗号鍵のライフサイクル全体を対象に定義した全ての鍵状態及び遷移条件を明確化することを要求したものである。ここで定義した鍵状態及び遷移条件を管理・実行するために必要な全ての管理機能を次節以降で規定することが求められる。

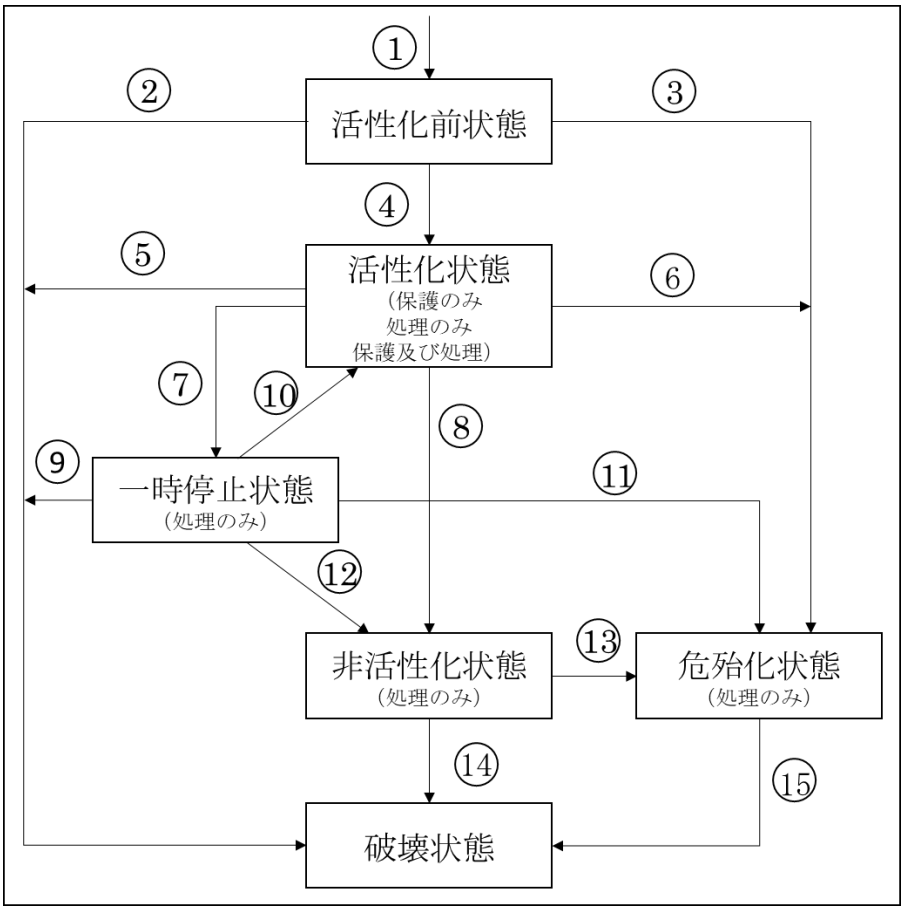
暗号鍵は、生成から破壊までの間（ライフサイクルという）にいくつかの状態を経由することとなる。これらの状態に応じて暗号鍵は異なる方法で使用される（又は使用が禁止される）ので、CKMS 設計の観点からは、どのような状態が存在し、どのような条件によって状態遷移が起きるのかを明らかにしておく必要がある。

項目 B.02 及び B.03 は、CKMS の設計にあたって、そのような暗号鍵のライフサイクル全体を対象に全ての鍵状態及び遷移条件を明確化することを要求したものである。その際、B.02 では、項目 B.01 で記載した暗号鍵が遷移する必要がある全ての状態を含まなければならない。

また、遷移の処理は次節でのライフサイクル管理機能により実現され、遷移条件に基づいて暗号鍵の状態がコントロールされるようにしなければならないので、B.02 及び B.03 で記載したことと次節での内容とが整合的であるようにしなければならないことに注意する必要がある。

暗号鍵の状態及び遷移の詳細については、NIST SP800-57 Part1 Rev5[6]の第 7 章を参照されたい。図 2-5 にその概要を記す。

[6] https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final
日本語訳：https://www.ipa.go.jp/files/000090943.pdf



【遷移】

- ✧ 正常な遷移：①→④→⑧→⑭
- ✧ それ以外はすべて何らかの異常による遷移

図 2-5　鍵状態及び遷移の例

- 活性化前状態：

  暗号鍵は生成されたが、使用が認可されていない状態のこと。鍵の所持証明や確認など、暗号鍵の検証にのみ使用することができ、それ以外の目的（例：暗号化や署名などの処理）に使ってはいけない。

  また、活性化前状態に遷移する段階（図 2-5 の①）で、暗号鍵の生成やエンティティ・所有者の確認などが行われる。

- 活性化状態：

  暗号鍵が、実際の暗号処理で使用することを認可されている状態のこと。暗号鍵のタイプに応じて、保護のみ（暗号化や署名生成）、処理のみ（復号や署名検証）、又は保護と処理の両方のいずれかを選んで指定できる。

  また、活性化前状態からの遷移は（図 2-5 の④）は、活性化前状態に入った直後に遷移する場合もあれば、何らかのトリガー（開始日時など）により遷移する場合もある。

- 非活性化状態：
  暗号鍵が、保護（暗号化や署名生成）を行うために使用してはならないが、場合によっては、暗号化された保護情報を処理（復号や署名検証）するために使用されうる状態のこと。例えば、アーカイブされている暗号鍵が非活性化状態に該当していることがあり得る。また、活性化状態は終了したが破壊状態になる前の暗号鍵も非活性化状態として取り扱われる。
  活性化状態からの遷移は（図 2-5 の⑧）は、基本的に当該鍵の有効期限が終了したタイミングで行われる。

- 破壊状態[7]：
  暗号鍵が完全に存在しなくなった状態のこと。バックアップやアーカイブされている暗号鍵についても（もしあれば同時に）破壊することが求められる。
  暗号鍵が破壊されると、それ以降、当該暗号鍵の危殆化を心配する必要はなくなる。一方で、当該暗号鍵で保護（暗号化）を行った情報を処理（復号）することはできなくなり、当該情報を永遠に喪失する結果を招くことになるので、誤って暗号鍵の破壊をしてしまうことがないようにしておくことも重要である。

- 危殆化状態：
  暗号鍵が、実際の暗号処理で安全に利用できる保証がなくなった状態、又はその可能性が生じた状態のこと。例えば、暗号鍵の漏えいや暴露、認可されていないエンティティによる不正アクセスなどが生じた時（又はその疑いが生じた時）に危殆化状態への遷移（図 2-5 の③⑥⑪⑬）を実施する。
  危殆化状態となった暗号鍵（及びその鍵ペア）は失効させる必要があり、その後、保護（暗号化や署名生成）を行うために使用されてはならない。また、暗号化された保護情報を処理（復号や署名検証）する場合には、その情報の正当性や完全性が疑わしくなっていることを十分に認識したうえで、どのように取り扱うのかを考える必要がある。例えば、危殆化の発生前から物理的に保護されているか、信頼できるタイムスタンプの利用など別の保護手段により保護されているか、などの視点を踏まえて、当該情報の正当性や完全性を受け入れるか受け入れないかを判断することが必要となる。

- 一時停止状態：
  暗号鍵の利用が一時的に認められなくなっている状態のこと。一時停止状態にある暗号鍵は、保護（暗号化や署名生成）を行うために使用されてはならない。
  一般に、一時停止状態への遷移（図 2-5 の⑦）には、i）暗号鍵の危殆化が疑われ、その状況を調査するための時間を確保しつつ、万が一の危殆化の影響が発生しないようにするケースと、ii）当該暗号鍵を所有するエンティティが一定期間利用しない（長期休暇など）こ

[7] 設計指針（基本編）では「破壊状態」、SP 800-57 の日本語訳では「破棄状態」と記載されているが、本ガイダンスでは「破壊状態」で統一する。「暗号鍵を破棄」といった場合、「（再利用できないように）暗号鍵の存在自体を完全に消す」場合と「（簡単には使えないように）暗号鍵を読み出せなくする」場合が考える。破壊状態とみなすのは前者だけであることに留意されたい。後者は、簡単には暗号鍵を読み出せなくなっているというだけでデータとしては残ったままということであり、このような状態は破壊状態とはみなさない。

とが明らかであり予防保全を行うケース、のどちらかの理由に起因して実施されることが考えられる。i）の場合、調査の結果により、危殆化の恐れが払拭されれば活性化状態に遷移（図 2-5 の⑩）することもあり得るが、そうでなければ危殆化状態に遷移（図 2-5 の⑪）するのが普通である。ii）の場合は、利用しない期間が明ければ（休み明けにより利用再開など）即座に活性化状態に遷移（図 2-5 の⑩）するのが一般的である。

《トイモデルと記載例》

本節のトイモデルも 2.1 節と同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムとする。

図 2-6 の通り、署名プライベート鍵は署名生成（保護）のために利用するため、有効期間前の活性化前状態、有効期間内の活性化状態、有効期間後の非活性化状態に遷移する。その後、実際に署名プライベート鍵が破壊されると破壊状態に遷移する。また、署名プライベート鍵の危殆化が疑われる場合は危殆化状態に遷移させる。なお、本システムでは一時停止状態は設定しない。

署名プライベート鍵のペアとなる署名公開鍵についても、有効期間前の活性化前状態、有効期間内の活性化状態、有効期間後の非活性化状態に遷移する。その後、実際に署名プライベート鍵が破壊されると破壊状態に遷移する。また、署名プライベート鍵が危殆化状態に遷移した場合、署名公開鍵も同時に危殆化状態に遷移させる。これら以外の遷移条件は設定しないこととする。

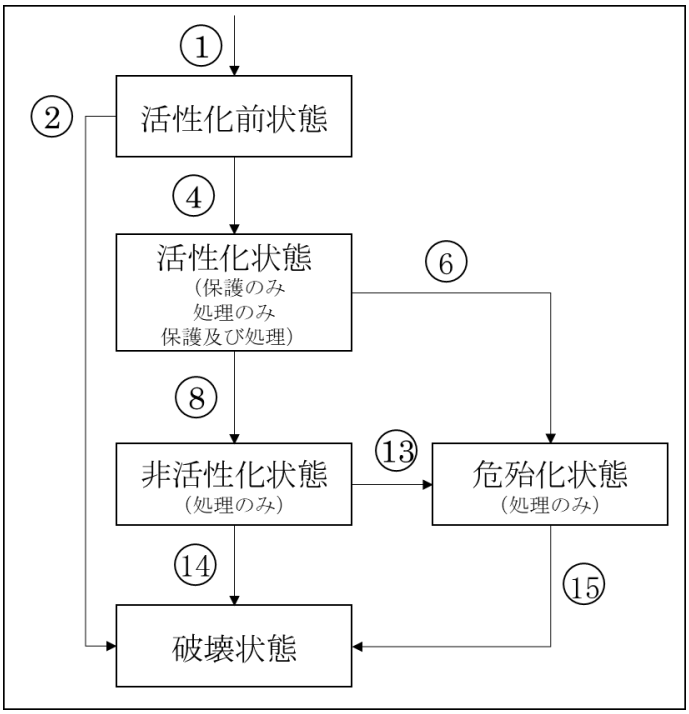


図 2-6　トイモデルの鍵状態及び遷移

以上のトイモデルにおける記載例は、以下の「署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例」のようになる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.02	活性化前状態、活性化状態、危殆化状態、非活性化状態、破壊状態
B.03	● 署名プライベート鍵と署名公開鍵の鍵ペア生成後に活性化前状態に遷移（図 2-6 の①） ● 証明書に記載された有効期間の開始時に活性化前状態から活性化状態に遷移（図 2-6 の④） ● 活性化状態に遷移前に署名プライベート鍵に問題が生じた場合は、活性化前状態から破壊状態へ遷移（図 2-6 の②） ● 有効期限の終了時に活性化状態から非活性化状態に遷移（図 2-6 の⑧） ● 鍵の破壊条件を満たした場合に非活性化状態から破壊状態に遷移（図 2-6 の⑭） ● 活性化状態に遷移後に署名プライベート鍵の危殆化が疑われる事象が発生した場合は、活性化状態又は非活性化状態から危殆化状態に遷移（図 2-6 の⑥⑬）。署名公開鍵も同時に危殆化状態に遷移させる ● 証明書失効リスト（CRL リスト）に記載された署名公開鍵は危殆化状態に遷移（図 2-6 の⑥⑬） ● 危殆化処理完了時に破壊状態に遷移（図 2-6 の⑮） 注）本システムでは、図 2-5 の③⑤⑦⑨⑩⑪⑫の遷移は設定しない。

2.3 暗号鍵のライフサイクル管理機能

① 鍵情報に対する管理のために実行される機能の全体像

項目	FR 番号	Framework Requirements の内容	SP800-130
B.04	FR6.17	CKMS 設計は、実装されサポートされる鍵情報（暗号鍵及びメタデータ）の管理機能を明記しなければならない。	6.4 節
B.05	FR6.18	CKMS 設計は、CKMS に実装されるそれぞれの鍵情報（暗号鍵及びメタデータ）の管理機能のパラメタに適用される完全性、機密性、及びソース認証（source-authentication）の処理（service）を特定しなければならない。	6.4 節

解説・考慮点

項目 B.04 は、CKMS の設計にあたって定義した暗号鍵のライフサイクルにおける鍵状態及び遷移条件を管理・実行するために必要な全ての管理機能を明確化し、実装することを要求したものである。対象となる管理機能は本節の②以降である。
項目 B.05 は、管理機能に共通して入出力されるデータについて、完全性や機密性、ソース認証が必要となるものがあれば、それらを明確化することを要求したものである。これには、エンティティの認証及び認可が含まれることもある。

項目 B.04 の目的は、暗号鍵のライフサイクル管理機能として②以降で対象となる機能を全て明記することによって、詳細を定めなければいけない項目に抜けが生じないようにすることである。なお、それぞれの機能の詳細については次節以降で取り扱うことになるのでここであまり詳細に定める必要はないが、前節にも記載した通り、ライフサイクル管理機能が暗号鍵のライフサイクルを実現するための手段であるので、その基本方針となる B.02 及び B.03 で記載したことと整合的であるようにしなければならない。

項目 B.05 の目的は、管理機能に入出力される個々の鍵情報（暗号鍵及びメタデータ）が、完全性や機密性、ソース認証のどの処理（service）に使われるものなのかを明記することにより、鍵情報が誤った使い方をされていないことを確認することである。したがって、B.01 で記載した暗号鍵全てを包含していることが求められる。また、②以降での処理（service）についての項目において、B.05 と矛盾していないことを確認する必要がある。

②　鍵活性化機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.06	FR6.22	CKMS 設計は、それぞれの鍵タイプがどのように活性化されるか、及び鍵が活性化される状況を明記しなければならない。	6.4.3 節
B.07	FR6.23	それぞれの鍵タイプに対して、CKMS 設計は、鍵活性化の通知の要求事項を明記しなければならない。それには、どの当事者に通知されるか、どのように通知されるか、どのセキュリティ処理（services）が通知に適用されるか、及び通知の期間が含まれる。	6.4.3 節

解説・考慮点

暗号鍵の活性化前状態から活性化状態への遷移を提供する機能である。
項目 B.06 及び B.07 は、CKMS の設計にあたって、鍵活性化の手順や遷移条件、通知方法等、鍵活性化機能への要求事項を明確化することを求めたものである。

項目 B.06 は、活性化前状態から活性化状態への遷移（図 2-5 の④）を実現するための手順や遷移条件を具体化することを求めたものであり、B.03 で記載した活性化状態への遷移条件と整合的であるようにしなければならない。

項目 B.07 は、暗号鍵が活性化状態になった時に、当該暗号鍵を利用するエンティティ（利用者）に、その暗号鍵が暗号処理に利用できるようになったことを通知する方法を具体化することを求めたものである。例えば、データ暗号化対称鍵であれば当該鍵の利用エンティティだけに秘密裏に通知できる手段を使って通知する必要がある一方、署名公開鍵であれば多くのエンティティに周知できる手段を使う必要があるかもしれない。

活性化状態に遷移したことが通知されない場合、当該暗号鍵を利用するエンティティ（利用者）はいつからその暗号鍵が使えるのかが分からないため、基本的には活性化状態へ遷移する前にい

つから使えるのか通知することが必要であると考えるのがよい。ただし、通知手段としては、利用者にも知らせる形で行われる場合もあれば、利用者に感知させることなく機器間で自動的に行われる場合もある。

③　暗号機能の実行場所の特定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.08	FR6.69	CKMS 設計は、サポートされている全ての暗号機能、及びそれらの暗号機能が CKMS のどこで実行されるか（例えば、CA、ホスト、又はエンドユーザシステム）を明記しなければならない。	6.4.27 節

解説・考慮点

データへの暗号学的保護を実際に提供する機能であり、署名生成、署名検証、暗号化、復号、鍵ラッピング、鍵アンラッピング、MAC 生成、及び MAC 検証を含む。
項目 B.08 は、CKMS の設計にあたって、暗号機能がどこで実行されるのか明確化することを要求したものである。

項目 B.08 は、暗号機能がどこに存在し、どこで実行されるのかを把握しておくことを求めたものである。暗号機能が実行されるところでは必ず暗号鍵が平文の形で使われることになるため、暗号鍵が保管される場所と並んで、もっとも暗号鍵の危殆化が発生しやすい場所である。したがって、暗号機能がある場所や実行場所を把握しておくことで、暗号鍵が狙われるリスクを低減させるために重点的に対策・保護すべき場所の絞り込みに活用できる。

例えば、暗号モジュールの内部で暗号機能が実行されることとなれば、暗号鍵が平文の形で使われるのは暗号モジュール内に限定される。

④　鍵非活性化機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.09	FR6.24	CKMS 設計は、各鍵タイプに対して、鍵の非活性化がどのように決定されるのか（例えば、暗号鍵有効期間（cryptoperiod）による、使用回数による、又はデータ量による）を明記しなければならない。	6.4.4 節
B.10	FR6.25	CKMS 設計は、それぞれの鍵タイプがどのように非活性化されるか（例えば、非活性化日時、使用回数、又は保護されたデータの量に基づいて、手動で行われるのか自動で行われるのか）を明記しなければならない。	6.4.4 節

B.11	FR6.26	CKMS 設計は、それぞれの鍵タイプの非活性化日時がどのように変更できるかを明記しなければならない。	6.4.4 節
B.12	FR6.27	それぞれの鍵タイプに対して、CKMS 設計は、鍵タイプの非活性化の事前通知の要求事項を明記しなければならない。それには、CKMS がサポートするどの役割に通知されるか、どのように通知されるか、どのセキュリティ処理（services）が通知に適用されるか、及び通知の期間が含まれる。	6.4.4 節

解説・考慮点

暗号鍵の非活性化状態への遷移を提供する機能である。CKMS セキュリティポリシーには当該ポリシーがカバーするあらゆる鍵タイプについて最大許容暗号鍵有効期間を記載すべきであり、その期間を超えた暗号鍵有効期間を設定してはならない。
項目 B.09～B.12 は、CKMS の設計にあたって、鍵非活性化の手順や遷移条件、変更方法、通知方法等、鍵非活性化機能への要求事項を明確化することを求めたものである。

セキュリティ対策の一環として、多くの場合、暗号鍵が無期限に使われることはなく、あるタイミングで暗号鍵の交換を行う。暗号鍵の活性化状態に存在している時間のことを暗号鍵有効期間という。この時間は、扱う情報の資産価値、求められる情報の機密性や完全性、CKMS への脅威、暗号鍵の交換に伴うメリットとデメリットの比較などに基づいて決められる。

暗号鍵有効期間が経過した暗号鍵は非活性化状態に遷移することで、新たな保護（暗号化や署名生成）を行うことはできなくなる。

項目 B.09 と B.10 は、管理対象となる各々の暗号鍵がどのような条件を満たしたときにどのような方法で非活性化状態に遷移するのかを決めるためのものである。例えば、遷移条件では、暗号鍵有効期間のほか、使用回数やデータ量などを使うケースもある。いずれの条件を使うとしても、CKMS セキュリティポリシーや CKMS 設計などで最大許容暗号鍵有効期間が決まっている場合には、その最大期間と矛盾しないようにしなければならない。また、遷移方法では、暗号鍵が自動的に利用できなくなるように非活性化状態に直接遷移する（図 2-5 の⑧）ような場合もあれば、例えば、一時停止状態に自動的に遷移（図 2-5 の⑦）した後、管理者等の確認処理を経て非活性化状態に遷移する（図 2-5 の⑫）ようなやり方も想定される。

項目 B.11 は、非活性化状態への遷移条件の変更を例外的に認めるかどうか、また、認める場合でもどのような条件下で認めるのかを決めておくものである。基本的には例外を認めない方がよいと考えられるが、システム上の要請などにより例外的に遷移条件の変更を容認する必要がある場合にはそのことを明確にしておくこと、また例外がなし崩しにならないような明確な条件として定めておくことが重要である。

なお、これらは B.03 の非活性化状態への遷移条件と整合していなければならない。

項目 B.12 は、暗号鍵が非活性化状態になるより前に、当該暗号鍵を利用するエンティティ（利用者）に、その暗号鍵が暗号処理に利用できなくなることを予告する方法を具体化することを求

めたものである。非活性化状態に遷移することが事前に通知されることなく、非活性化状態に遷移した場合、当該暗号鍵を利用するエンティティ（利用者）にとっては突然その暗号鍵が使えなくなることを意味し、それが故障などの予期せぬ原因によって生じた事態だと誤認する恐れがある。このような事態を避けるのが事前通知の役割であり、何らかの方法で通知する仕組みを設けたほうがよい。ただし、通知手段としては、利用者にも知らせる形で行われる場合もあれば、利用者に感知させることなく機器間で自動的に行われる場合もある。

⑤　鍵失効機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.13	FR6.28	CKMS 設計は、いつ、どのように、どのような状況で失効が実行され、失効情報を依拠する当事者が利用可能になるかを明記しなければならない。	6.4.5 節
B.14	FR6.108	CKMS 設計は、使用される又は使用できる鍵失効メカニズム及び関連付けられた依拠するエンティティへの通知メカニズムを明記しなければならない。	6.8.3 節

解説・考慮点

暗号鍵有効期間より前に当該暗号鍵の使用を終了させる必要が生じた場合に行われ、暗号鍵の危殆化状態への遷移を提供する機能である。この機能が実行されると、過去に保護された情報の処理のための使用に対しても完全なセキュリティは保証されないので、当該暗号鍵を速やかに置き換える能力とその鍵を使用する当事者に危殆化／失効を通知する能力を備えているべきである。
CKMS の設計にあたって、項目 B.13 は、鍵失効の遷移条件及び通知方法といった鍵失効機能への要求事項を明確化することを求めたものである。B.14 は、実際に利用する具体的な鍵失効メカニズム及び通知メカニズムを明確化することを要求したものである。

非活性化状態への遷移は、暗号鍵有効期間が満了したなどの予め決められた条件に従って実行される。一方、危殆化状態への遷移は、暗号鍵の漏えいなどによる予期せぬ原因により、当該暗号鍵の安全性が担保できないと判断された場合に実行される。つまり、暗号鍵の安全性低下が遷移の主要因になっているかどうかが、非活性化状態への遷移と危殆化状態への遷移の違いである。
鍵失効機能は、危殆化状態への遷移を実行するための機能のことである。

項目 B.13 は、どのようなことが発生したら危殆化状態と見なすのかといった鍵失効の遷移条件と誰にどのように危殆化状態に遷移した事実を知らせるのかといった通知方法など、鍵失効機能への要求事項を明確化することを求めている。これは、B.03 の危殆化状態への遷移条件と整合していなければならない。B.14 は、実際に利用する具体的な鍵失効メカニズム及び通知メカニズムを明確化することを要求している。

鍵失効の遷移条件及び具体的な鍵失効メカニズムは、「いつ、どのように、どのような状況で危殆化した（と推定される）鍵情報を失効させ、利用できなくするか」の視点で検討される。

一方で、この機能が実行されると、それ以降の保護（暗号化や署名生成）のセキュリティが担保されないだけでなく、過去に保護された情報の処理（復号や署名検証）に対しても完全なセキュリティは保証されなくなる。このため、危殆化した暗号鍵を速やかに置き換え、新しい暗号鍵で保護が再開できるようにすることはもとより、当該暗号鍵を使用して過去に保護された情報を処理するエンティティ（利用者）全員に完全なセキュリティが保証されない可能性があることの注意喚起を行う必要がある。鍵失効の通知が重要であるのはこのためである。

失効情報を依拠する当事者及び依拠するエンティティへの通知メカニズムは、「危殆化した（と推定され、失効した）暗号鍵を使用して過去に保護された情報について完全なセキュリティが保証されない可能性があることをどのように関係者に通知するか」の視点で検討される。例えば、危殆化／失効の通知方法としては、危殆化鍵リスト、証明書失効リスト（CRLs)、ホワイトリスト、クエリホワイトリスト、OCSP（Online Certificate Status Protocol）がある。

⑥　暗号鍵の一時停止機能及び再活性化機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.15	FR6.29	CKMS 設計は、どのように、どのような状況で鍵が一時停止されるかを明記しなければならない。	6.4.6 節
B.16	FR6.30	CKMS 設計は、どのように一時停止情報を依拠又は通信する当事者が利用可能になるかを明記しなければならない。	6.4.6 節
B.17	FR6.32	CKMS 設計は、どのように一時停止された鍵によるセキュリティ処理（services）の実行を防止するのかを明記しなければならない。	6.4.6 節
B.18	FR6.31	CKMS 設計は、どのように、どのような状況で一時停止された鍵が再活性化されるかを明記しなければならない。	6.4.6 節
B.19	FR6.33	CKMS 設計は、どのように再活性化情報を依拠又は通信する当事者が利用可能になるのかを明記しなければならない。	6.4.6 節

解説・考慮点

暗号鍵の一時停止状態への遷移を提供する一時停止機能、及び活性化状態への遷移を再度提供する再活性化機能のことであり、これらの機能は必ずセットで用いられる。
項目 B.15 及び B.16 は、CKMS の設計にあたって、暗号鍵の一時停止の遷移条件及び通知方法といった一時停止機能への要求事項を明確化することを求めたものである。B.17 は、一時停止された暗号鍵が利用されないように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。B.18 及び B.19 は、暗号鍵を再活性化するための遷移条件及び通知方法といった再活性化機能への要求事項を明確化することを求めたものである。

なお、鍵状態において一時停止状態を定義しない場合には、これらの項目は対象外である。

一時停止状態は、暗号鍵の利用が一時的に認められなくなっている状態であり、i）暗号鍵の危殆化が疑われ、その状況を調査するための時間を確保しつつ、万が一の危殆化の影響が発生しないようにするケースと、ii）当該暗号鍵を所有するエンティティが一定期間利用しない（長期休暇など）ことが明らかであり予防保全を行うケース、のどちらかの理由で使われることが想定されている。前者の「暗号鍵の危殆化の疑い」には、暗号鍵の誤使用や誤配置などに暗号鍵の危殆化につながりかねない状況を含んでもよい。

なお、鍵状態として必ず一時停止状態を用意する必要があるわけではなく、i）のケースでは活性化状態から危殆化状態に直接遷移するようにし、ii）のケースでは活性化状態のままにしておく、という管理の仕方をしても構わない。もし一時停止状態を定義しない場合には、ここでの項目は全て対象外となる。

一時停止状態を用意することのメリットは、i）のケースでは、調査の結果で危殆化の恐れが払拭された場合には、同じ暗号鍵を再活性化することで継続利用が可能になり、暗号鍵の更新に関連する処理を実施する必要はないことである。もし危殆化状態に遷移させていた場合、危殆化の恐れが払拭された場合でも暗号鍵の更新・再設定が必要となる。また、ii）のケースでは、長期利用しないエンティティになりすまされて不正利用される事態を防止できることである。

一方、一時停止状態と用意する場合には、どのようなことが発生したら一時停止させるのかといった一時停止状態への遷移条件（図 2-5 の⑦）と、どのようなことが満たされたら一時停止を解除するのかといった活性化状態への遷移条件（図 2-5 の⑩）を必ずセットで準備しなければならない。また、一時停止状態になった暗号鍵は保護（暗号化や署名生成）を行うために使用できないので、当該暗号鍵を利用する処理が行われることはないことを関係するエンティティ（利用者）全員に注意喚起する必要がある。

項目 B.15 と B.18 は上述の一時停止状態への遷移条件と活性化状態への遷移条件を、B.16 と B.19 は関係エンティティ(利用者)への通知方法を明確化することを求めたものである。B.17 は、一時停止された暗号鍵が利用されないようにするための要求事項の明確化のことである。これらの要件は B.03 の一時停止状態への遷移条件及び再活性化の遷移条件と整合していなければならない。

⑦　鍵情報の破壊機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.20	FR6.38	CKMS 設計は、どのように、どのような条件で鍵が意図して破壊されるか、及び破壊がコンポーネントへの局所的（local）なものであるか CKMS 全体への共通的（universal）なものであるかを明記しなければならない。	6.4.9 節

B.21	FR6.39	それぞれの鍵タイプに対して、CKMS 設計は、鍵破壊の事前通知の要求事項を明記しなければならない。それには、どの当事者に通知されるか、どのように通知されるか、どのセキュリティ処理（services）が通知に適用されるか、及び通知の時期が含まれる。	6.4.9 節

解説・考慮点

暗号鍵の破壊状態への遷移を提供する機能である。
項目 B.20 は、CKMS の設計にあたって、鍵情報を破壊するための条件及び具体的な破壊方法、並びに当該鍵情報がどこに存在するかを含めて明確化することを要求したものである。B.21 は、鍵情報が破壊されたことの通知方法といった破壊機能への要求事項を明確化することを求めたものである。

暗号鍵が使えない状態になっていたとしても、どこかに当該暗号鍵のデータが残っていれば、攻撃や不正持出など、何らかの理由によりそのデータが漏えいするリスクを完全には排除できない。逆に言えば、暗号鍵の漏えいリスクを完全に排除するためには、当該暗号鍵が完全に存在しなくなった状況にするしかない。当然、実際に利用中の暗号鍵だけでなく、バックアップやアーカイブされたものも含めて完全に存在しない状況にする必要がある。つまり、バックアップやアーカイブしている場合で鍵の破壊を行うケースでは、鍵の破壊[8]に関して B.20 の記載内容と B.54 や B.56 の記載内容が整合していなければならない。

一方、暗号鍵を破壊してしまえば、例え当該暗号鍵の所有者であったとしても、当該暗号鍵で保護（暗号化）を行った情報を処理（復号）することはできなくなり、その情報は永遠に失われる結果を招くことになる。したがって、誤って暗号鍵の破壊をしてしまうことがないようにしておくことも重要である。
項目 B.20 では、CKMS の設計にあたって、鍵情報を破壊するための条件などの他、破壊した暗号鍵の影響範囲がどこまで及ぶのかを明確にしておくことを求めている。例えば、利用者 A だけが使う暗号鍵であれば、その暗号鍵が破壊されたことによって影響を受けるのは利用者 A だけに限定される。つまり、局所的（local）な影響である。一方、CKMS 全体を管理するルート鍵が破壊されると CKMS 全体で当該ルート鍵が利用できなくなるので、その影響は共通的（universal）であると言える。
なお、ここでの要件は B.03 の破壊状態への遷移条件と整合していなければならない。

B.21 は、鍵情報が破壊されたことの通知方法の要求事項を明確化することを求めたものである。鍵情報の破壊は、当該暗号鍵が使えなくなるだけでなく、当該暗号鍵で保護（暗号化）を行った情報も喪失することを意味するので、注意喚起の意味でも関連するエンティティ（利用者）に事前通知しておくことが特に重要になる。

[8] SP 800-88 Rev.1「Guidelines for Media Sanitization（媒体のデータ抹消処理（サニタイズ）に関するガイドライン）」中に暗号鍵の破壊方法についての記載（5 章参照）があるので、必要に応じて参照されたい。
https://csrc.nist.gov/publications/detail/sp/800-88/rev-1/final
日本語訳：https://www.ipa.go.jp/files/000094547.pdf

⑧　鍵生成機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.22	FR6.19	CKMS 設計は、それぞれの鍵タイプに対して、CKMS で使用される鍵生成手段を明記しなければならない。	6.4.1 節
B.23	FR6.20	CKMS 設計は、対称鍵及びプライベート鍵を生成するのに使用される元となる乱数生成器を明記しなければならない。	6.4.1 節

解説・考慮点

CKMS の設計にあたって、項目 B.22 は暗号鍵を生成する手段について、B.23 は暗号鍵を生成する際に利用する乱数生成器について明確化することを要求したものである。一般に、鍵生成手段は暗号鍵と対になる暗号アルゴリズムの仕様に依存し、暗号目的として設計された乱数生成器の使用を要求する。

暗号アルゴリズムの安全性は、利用する暗号鍵が予測できないことを前提としている。逆に言えば、利用する暗号鍵が何らかの方法で予測できるのであれば、どんなに強力な暗号アルゴリズムを使っていたとしても安全性は担保されない。そのため、予測できない暗号鍵を生成する手段を利用することが極めて重要になる。

ここでの「予測できない」の意味は、①出力が一様分布になっていること、②十分なエントロピー量が確保されていること、の両方を満たすことである。

①の出力が一様分布になっているということは、生成される可能性がある暗号鍵の種類が 2^{128} 個あるならば、どの鍵も一様に $1/2^{128}$ の確率で生成されるということである。このような性質を持つように作られていると信頼できる鍵生成方法として、SP 800-90 や SP 800-133 などの乱数生成器を使う方法と、SP 800-56 などのような鍵導出関数を使う方法がある。

②の十分なエントロピー量を確保されているということは、極めて多数の種類の暗号鍵が生成可能であるということを意味する。例えば、鍵長 128 ビットの暗号鍵であれば 2^{128} 個の異なる暗号鍵が利用できることが期待されるので、利用する鍵生成手段が 2^{128} 個の異なる暗号鍵を生成するように作られていれば「十分なエントロピー量を確保されている」という。一方、例えば、1,000 個の異なる暗号鍵しか生成しないような鍵生成手段であれば、例え出力が一様分布になっていたとしても「エントロピー量が十分に確保されていない」と判断される。これは、暗号鍵の全数探索で暗号解読しようとした場合、前者は平均 2^{127} 回のトライアルが必要となるが、後者は平均 500 回のトライアルをすればよいためである。

項目 B.22 は暗号鍵を生成する手段について、B.23 は暗号鍵を生成する際に利用する乱数生成器について明確化することを要求したものである。ここでのポイントは「信頼できる鍵生成手段」や「信頼できる乱数生成器」を使い、十分なエントロピー量が確保されていることを確認することにある。

⑨　鍵導出機能／鍵更新機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.24	FR6.36	CKMS 設計は、鍵を導出又は更新するために使用される全てのプロセス、及び鍵が導出又は更新される状況を明記しなければならない。	6.4.8 節
B.25	FR6.37	それぞれの鍵タイプに対して、CKMS 設計は、鍵の導出又は更新の事前通知の要求事項を明記しなければならない。それには、どの当事者に通知されるか、どのように通知されるか、どのセキュリティ処理（services）が通知に適用されるか、及び通知の期間が含まれる。	6.4.8 節

解説・考慮点

暗号鍵の生成では、鍵生成機能を利用する代わりに、鍵導出機能や鍵更新機能を利用することがある。
項目 B.24 及び B.25 は、CKMS の設計にあたって、鍵導出機能／鍵更新機能が利用される条件や鍵導出方法、通知方法等といった鍵導出機能／鍵更新機能への要求事項を明確化することを求めたものである。
なお、これらの機能は採用必須の機能ではなく、CKMS の設計で採用しなければ検討対象外である。逆に、採用してはならないという要求事項の場合もあり得る。

鍵導出機能では、一部が秘密であるような独立した他の情報（他の暗号鍵、共有秘密やパスワードなど）から不可逆な形で暗号鍵が導出されるプロセスを実行する。例えば、鍵確立プロトコルでは互いの共有秘密から共有鍵を導出する。

鍵更新機能では、「元鍵」から「別鍵」を計算で導出し、導出した「別鍵」で「元鍵」を置き換えるプロセスを実行する。なお、別鍵を導出する際に他の秘密データを使用しない場合には、元鍵と更新方法を知っている攻撃者が将来にわたるあらゆる時期の更新した別鍵を知りうるというセキュリティリスクにさらされる。

暗号鍵の予測可能性の観点でいえば、二つの点で、鍵生成機能を利用して生成される暗号鍵よりも予測がしやすくなっている可能性がある。①鍵導出機能や鍵更新機能により作られる暗号鍵は、「（乱数生成器ではなく）ある種の計算方法」に従って生成されることから、入力データと出力データ（生成された暗号鍵）との間に相関が残っている可能性がある、②入力データの種類によっては十分なエントロピー量が確保できない可能性がある。例えば、パスワードなどから鍵を導出するからといって、十分なエントロピー量の確保できるような複雑なパスワードを利用させることは困難を伴う。

これらの弱点による暗号鍵の予測可能性を少しでも低減するには、信頼できる鍵導出方法や鍵更新方法を使うのが望ましい。例えば、SP 800-108 や SP 800-132、SP 800-135 などである。

一方、鍵導出機能や鍵更新機能を使うことのメリットは、i）ローカルで（鍵生成機能を使うことなく）鍵更新ができる、ii）鍵確立プロトコルを利用しなくてもそれぞれのエンティティ（利用

者）が独自に同じ秘密データを使って同じ暗号鍵を生成することができる、という点である。前者は特に大規模 CKMS の場合や CKMS につながっていない機器などの場合に効果を発揮する可能性があり、後者は鍵確立時の漏えいリスクの低減に役立つ。

項目 B.24 及び B.25 は、CKMS の設計にあたって、鍵導出機能／鍵更新機能が利用される条件や鍵導出方法、通知方法等といった鍵導出機能／鍵更新機能への要求事項を明確化することを求めたものである。ここでのポイントは、B.24 は主に「信頼できる鍵導出方法や鍵更新方法」を使うことで暗号鍵の予測可能性のリスクを低減していることの確認を行うことである。B.25 は鍵導出や鍵更新を行う事前通知がないまま新しい暗号鍵に変わった場合、当該暗号鍵を利用するエンティティ（利用者）にとっては突然以前の暗号鍵が使えなくなることを意味し、それが故障などの予期せぬ原因によって生じた事態だと誤認する恐れがある。このような事態を避けるのが事前通知の役割であり、何らかの方法で通知する仕組みを設けたほうがよい。ただし、通知手段としては、利用者にも知らせる形で行われる場合もあれば、利用者に感知させることなく機器間で自動的に行われる場合もある。

なお、鍵導出機能や鍵更新機能は採用が必須の機能ではないので、CKMS 設計でこれらの機能を採用しないと決めたのであれば検討対象外となる。さらには、採用してはならないという要件を採用する場合もあり得る。

⑩　対称鍵の検証機能への要求事項／⑪　公開鍵の検証機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.26	FR6.66	CKMS 設計は、どのように、どこで、どのような状況で、対称鍵やそのメタデータが検証されるかを明記しなければならない。	6.4.24 節
B.27	FR6.63	CKMS 設計は、どのように、どこで、どのような状況で、公開鍵ドメインパラメタが検証されるかを明記しなければならない。	6.4.21 節
B.28	FR6.64	CKMS 設計は、どのように、どこで、どのような状況で、公開鍵が検証されるかを明記しなければならない。	6.4.22 節
B.29	FR6.65	CKMS 設計は、どのように、どこで、どのような状況で公開鍵証明書パスが検証されるかを明記しなければならない。	6.4.23 節

解説・考慮点

＜対称鍵の検証機能＞
対称鍵及びそのメタデータに対するテストを実行する機能である。
項目 B.26 は、CKMS の設計にあたって、対称鍵の検証機能が利用される条件や検証方法等といった対称鍵の検証機能への要求事項を明確化することを求めたものである。

＜公開鍵の検証機能＞
公開鍵についてある種の正当性チェックを実行して公開鍵が数学的に正しいことを保証し、公開ドメインパラメタについてもドメインパラメタが数学的に正しいことの保証を提供する機能である。
CKMS の設計にあたって、項目 B.27～B.29 は、公開鍵の検証機能が利用される条件や検証方法等といった公開鍵の検証機能への要求事項を明確化することを求めたものである。

生成されたり、共有されたりした暗号鍵が正当なものであることは、暗号アルゴリズムを利用する上での前提条件である。したがって、実際の暗号処理に利用する前には正当性を確認しておくことが必要である。

ただし、その確認を行うタイミングは、利用する暗号鍵の種類や利用方法、利用環境などによってさまざまである。例えば、タイミングだけ見ても、①暗号鍵が生成されたタイミングで実施、②暗号処理を行う直前に実施、③暗号鍵が活性化状態に遷移するタイミングで実施、④定期的に実施、などが考えられる。検証方法についても、a）ハッシュ値でのチェック、b）署名でのチェック、c）物理的手段でのチェック、d）相互通信可否でのチェック、e）CKMS サーバでのチェック、などが考えられる。

項目 B.26～B.29 は、いずれも検証機能が利用される条件や検証方法等といった、暗号鍵の検証に必要な要求事項を明確化することを求めている。具体的には、B.26 では対称鍵とそれに関連するメタデータが、B.27 では B.28 での公開鍵が利用するドメインパラメタが、B.28 では公開鍵が、それぞれ検証の対象として、「いつ（タイミング）、どこで（検証場所）、どのように（検証方法）」検証を行うのかを洗い出すことである。

同様に、B.29 では、B.28 での公開鍵とセットで作られる公開鍵証明書について、トラストアンカーから始まる公開鍵証明書のチェーン（証明書パス）を使った有効性検証を対象としている。

⑫　トラストアンカー管理機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.30	FR6.70	CKMS 設計は、サポートされている全てのトラストアンカー管理機能を明記しなければならない（[RFC6024] を参照）。	6.4.28 節
B.31	FR6.71	CKMS 設計は、依拠するエンティティがトラストアンカーについてのソース認証（source authentication）及び完全性検証を実行できるように、どのようにそれらのトラストアンカーがセキュアに配付されるかを明記しなければならない。	6.4.28 節
B.32	FR6.72	CKMS 設計は、依拠するエンティティのシステムのトラストアンカーストアに対して、認可された追加、変更、削除のみが行えること	6.4.28 節

		を保証するために、どのように依拠するエンティティのシステムでトラストアンカーが管理されるかを明記しなければならない。	

解説・考慮点

トラストアンカーなしでは信頼できるかわからない公開鍵に対して、信頼を確立するために使用されるトラストアンカーを保管・管理する機能である。
CKMS の設計にあたって、項目 B.30 はトラストアンカー管理機能についてどのようなものを受け入れるのかといった要求事項を、B.31 及び B.32 はトラストアンカーを完全かつセキュアに配送・保管・追加・削除等といったメンテナンスを行うためのトラストアンカー管理機能への要求事項を明確化することを求めたものである。

公開鍵暗号・署名では、プライベート鍵とそれに対応する公開鍵は一対一対応しているが、問題はそのプライベート鍵の所有者が「正当な」エンティティであることの保証が「公開鍵」からは得られないことである。例えば、Eve が自分自身のプライベート鍵に対応する公開鍵を「Alice の公開鍵」であると偽って公開していた場合に、Bob がその公開鍵が「Alice のものではない」と見破ることはほとんどできない。つまり、何も対策をしていないままの公開鍵は基本的に「信頼できるかわからない公開鍵」である。

そこで、信頼できる第三者であるトラストアンカーを用意し、そのトラストアンカーが「信頼できるかわからない公開鍵」にお墨付きを与えることで「信頼できる公開鍵」にする仕組みが必要になり、それを実現したのが公開暗号基盤（PKI）である。つまり、PKI では、トラストアンカーが公開鍵の信頼性の起点となることを意味し、その信頼性のうえで公開鍵暗号・署名の安全性が確保されていることになる。したがって、トラストアンカーの完全性確保は CKMS のセキュリティにとって死活的に重要である。

項目 B.30～B.32 は PKI でのトラストアンカーの完全性確保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。ここでのポイントは、具体的に、B.30 はどんなトラストアンカーを PKI の利用エンティティのシステムに用意するのか、B.31 はそのシステムに対してトラストアンカーの情報をどのように安全に配布するのか、さらに B.32 はそのシステムでのトラストアンカーの情報をどのように安全に管理するのか、といったことを確認することである。

例えば、トラストアンカー管理機能として OS での証明書管理機能を使い（B.30）、CKMS サーバから準備するトラストアンカーの公開鍵証明書（プライベート CA ルート証明書）を当該 CKMS サーバからダウンロードさせ、パスワード及びハッシュ値チェックで OK になった場合にそのルート証明書を証明書管理機能が管理するトラストアンカーストアに登録し（B.31）、OS での証明書管理機能でルート証明書の管理を行う（B.32）といったことである。

⑬　公開鍵の有効期間延長機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.33	FR6.34	CKMS 設計は、どのように、どのような条件で公開鍵の有効期間が延長できるかを明記しなければならない。なお、延長後の有効期間の合計が CKMS セキュリティポリシー等で定める公開鍵の最大許容暗号鍵有効期間を超える場合には、当該公開鍵の延長を行ってはならない。	6.4.7 節
B.34	FR6.35	それぞれの鍵タイプに対して、CKMS 設計は、鍵タイプの有効期間延長の事前通知の要求事項を明記しなければならない。それには、どの当事者に通知されるか、どのように通知されるか、どのセキュリティ処理（services）が通知に適用されるか、及び通知の期間が含まれる。	6.4.7 節

解説・考慮点

新しい有効期限を設定した「同じ公開鍵」を含む新しい公開鍵証明書を発行することで、以前の有効期間を超えて既存の公開鍵に対する新しい有効期間を確立するための機能である。なお、延長後の有効期間の合計が CKMS セキュリティポリシー等で定める公開鍵の最大許容暗号鍵有効期間を超える場合には、当該公開鍵の延長を行ってはならない。
CKMS の設計にあたって、項目 B.33 及び B.34 は、公開鍵の有効期間延長機能が利用される条件や事前通知方法等といった公開鍵の有効期間延長機能への要求事項を明確化することを求めたものである。

セキュリティ上の観点から、公開鍵証明書には有効期間が記載されている。有効期間が経過すると非活性化状態に自動的に遷移して当該公開鍵は失効し、また新しい公開鍵が鍵生成機能を使って生成されるのが一般的である。この場合、B.33 と B.34 は検討対象外となる。

しかし、新しい公開鍵を生成した場合、関連する鍵情報の更新も合わせて必要となったり、過去に署名したものに対する署名検証ができなくなったりするといったデメリットもある。こういったデメリットを回避する方法として、当該公開鍵の有効期間を延長するというやり方がある。これは、公開鍵の有効期間延長機能を使って、新しい有効期間を設定した「同じ公開鍵」を含む新しい公開鍵証明書を発行することによる実現する。

そもそも公開鍵証明書の有効期間は、トラストアンカーが「信頼できるかわからない公開鍵」を「信頼できる公開鍵」として利用できるお墨付きを与えている期間である。言い換えれば、その期間内は、公開鍵の安全性の担保をトラストアンカーが代理で受け持つということである。このことを踏まえれば、「有効期間を延長」することができるかどうかはトラストアンカーがその分の「責任を継続して負う」ことができるかどうかに依存する。項目 B.33 は、この「責任を継続して負う」ことができる条件を明確化することを求めたものである。

なお、CKMS セキュリティポリシー等で定める公開鍵の最大許容暗号鍵有効期間は、セキュリ

ティ上の要件として定められたものであるので、これに違反するような延長は認められないことに注意されたい。例えば、最大許容暗号鍵有効期間が 3 年である場合、1 年間有効の公開鍵証明書は 2 度延長することはできるが、3 度目の延長はできない。

公開鍵証明書の有効期間を延長するためには、その有効期間が切れる前に新たな有効期間を設定した公開鍵証明書の発行が必要となる。その手続きを行わせるためのトリガーとなるのが事前通知であり、項目 B.34 で事前通知を行うための条件の明確化を求めている。通知手段としては、利用者にも知らせる形で行われる場合もあれば、利用者に感知させることなく機器間で自動的に行われる場合もある。

⑭　所有者登録機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.35	FR6.21	CKMS 設計は､鍵と所有者の識別子を結び付けるプロセスを含めて、所有者登録に関わる全てのプロセスを明記しなければならない。	6.4.2 節

解説・考慮点

セキュリティエンティティ（個人、組織、デバイス、又はプロセス）及びメタデータを伴う暗号鍵の最初の登録を行うための機能である。
項目 B.35 は、CKMS の設計にあたって、所有者登録機能への要求事項を明確化することを求めたものである。

暗号鍵は、正しいエンティティ（利用者）に届ける必要がある。もし登録時点で誤った所有者登録が行われてしまうと、誤ったエンティティと暗号鍵が結び付くことになり、その後の暗号鍵のライフサイクルが正しく実行されたとしても全く安全性が担保されない。

したがって、所有者登録では、①どのようにエンティティ（利用者）が正しいことを確認するか、②どのようにそのエンティティと暗号鍵とを結びつけるか、を把握しておくことが重要である。典型的には、エンティティの対称鍵、公開鍵又はプライベート鍵の初期セットと、エンティティ識別子及びメタデータとも結び付ける登録プロセスが存在する。

項目 B.35 は、これら把握しておくべきことを明らかにすることを求めている。

⑮　プライベート鍵所持の検証機能への要求事項／⑯　プライベート鍵の検証機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.36	FR6.68	CKMS 設計は、どのように、どこで、どのような状況で、プライベート鍵とそのメタデータの所持が検証されるかを明記しなければならない。	6.4.26 節
B.37	FR6.67	CKMS 設計は、どのように、どこで、どのような状況で、プライベート鍵又は鍵ペア、あるいはそのメタデータが検証されるかを明記しなければならない。	6.4.25 節

解説・考慮点

＜プライベート鍵所持の検証機能＞
公開鍵の所有者であると主張する者が対応するプライベート鍵を所持していることの保証を得るために、公開鍵を受領したエンティティによって使用される機能である。
項目 B.36 は、CKMS の設計にあたって、プライベート鍵所持の検証機能が利用される条件や検証方法等といったプライベート鍵所持の検証機能への要求事項を明確化することを求めたものである。

＜プライベート鍵の検証機能＞
プライベート鍵に対してある種のテストを実行し、鍵ペアの仕様を満たすことの保証を提供するための機能である。
項目 B.37 は、CKMS の設計にあたって、プライベート鍵の検証機能が利用される条件や検証方法等といったプライベート鍵の検証機能への要求事項を明確化することを求めたものである。

PKI に則り、トラストアンカーが「信頼できるかわからない公開鍵」を「信頼できる公開鍵」として利用できるお墨付きを与えるためには、その公開鍵に対応する「正しい」プライベート鍵（及びそれに付随するメタデータ）を「正当な」エンティティが所有していることを確認する必要がある。

その一方、プライベート鍵は基本的に唯一のエンティティのみが秘密に所持することが求められるため、当該公開鍵に対応する「正しい」プライベート鍵（及びそれに付随するメタデータ）を「正当な」エンティティが所有していることを、トラストアンカーにその中身を直接示して証明するわけにはいかない。

そこで、項目 B.36 は、プライベート鍵（及びそれに付随するメタデータ）（と称するデータ）を所持していることを「正しい」エンティティが所有していることを確認するための検証方法の明確化を求めたものである。また、B.37 は B.36 での検証対象となったプライベート鍵（及びそれに付随するメタデータ）（と称するデータ）と（お墨付きを与える対象の）公開鍵とが一対一の

正しい関係にあり、結果として「正しい」プライベート鍵（及びそれに付随するメタデータ）であることを確認するための検証方法の明確化を求めたものである。

つまり、B.36 及び B.37 に両方を実施することで、トラストアンカーにプライベート鍵（及びそれに付随するメタデータ）の中身を直接示すことなく、当該公開鍵に対応する「正しい」プライベート鍵（及びそれに付随するメタデータ）を「正当な」エンティティが所有していることを証明する。なお、これらの機能は、プライベート鍵の所有者又はプライベート鍵の所有者の代理として振舞う信頼される第三者のみが実行できる。

⑰　暗号鍵とメタデータの関連付け機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.38	FR6.40	使用されているそれぞれの鍵タイプに対して、CKMS 設計は、何のメタデータが鍵と関連付けられているか、どのようにメタデータが鍵と関連付けられているか、及びメタデータが鍵と関連付けられる状況を明記しなければならない。	6.4.10 節
B.39	FR6.41	使用されているそれぞれの鍵タイプに対して、CKMS 設計は、どのように次のセキュリティ処理（services）（保護）が関連付けられたメタデータに適用されるかを明記しなければならない：ソース認証（source authentication）、完全性、及び機密性。	6.4.10 節

解説・考慮点

暗号鍵に関連付けられているメタデータ要素がある場合、それらを関連付けるための機能である。なお、暗号鍵とメタデータの関連付けは、当該鍵情報の生成時のほか、配送時、登録時、保管時など、暗号鍵有効期間を通じて完全性を維持する必要がある。
CKMS の設計にあたって、項目 B.38 及び B.39 は、関連付け機能が利用される対象や条件、関連付けの方法等といった関連付け機能への要求事項を明確化することを求めたものである。

鍵情報は、暗号鍵とメタデータとで構成される。4.1 節に記載があるように、メタデータとは、暗号鍵を適切に管理するために、その暗号鍵に関連付けられている情報である。例えば、暗号鍵のパラメタ、保護方法や有効期間などである。「暗号鍵管理システム設計指針（基本編）」では、メタデータの典型的な要素として 23 種類が記載されている。鍵タイプによってはメタデータとの組み合わせによりその暗号鍵の完全性を確保する場合もあるため、とりわけそのような鍵タイプの暗号鍵ではメタデータの管理も安全な鍵情報の管理を実施するために重要となる。

項目 B.38 と B.39 は、各々の鍵タイプに対して、暗号鍵に関連付けられているメタデータがあるのか、ある場合にはどのようなメタデータが関連付けられているのか、どのように関連付けられているのか、そして、何のためにそれが行われているのか、を把握することにポイントがあり、それを実現するために必要な機能要件を明確化することである。

ちなみに、関連付けを提供する保護メカニズムには、暗号学的プロセスを使用する場合と信頼プロセスを使用する場合とがある。直感的には、前者は、暗号鍵とメタデータの組で計算されたデジタル署名など、暗号アルゴリズムによって関連付けが保証される。後者は、信頼されるエンティティからのメタデータの対面手渡しやセキュアなストレージでの保管など、物理的な手段で関連付けが保証される。

留意するポイントとしては、B.38 と B.39 は「関連付け機能への要求事項」であるため、詳細な内容までを求めているわけではなく、そういった具体的な内容は D.02 以降で取りまとめられる。B.38 と B.39 は、D.02 以降に記載された暗号鍵とメタデータの関連付けを行う上で必要となる機能が用意されているか、整合的であるかの観点で確認することが重要である。

⑱　メタデータの変更機能への要求事項／⑲　メタデータの削除機能への要求事項／⑳　暗号鍵のメタデータリスト化機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.40	FR6.42	CKMS 設計は、関連付けられたメタデータが変更される状況を明記しなければならない。	6.4.11 節
B.41	FR6.43	CKMS 設計は、鍵と関連付けられたメタデータが削除される状況を明記しなければならない。	6.4.12 節
B.42	FR6.44	CKMS 設計は、関連付けられたメタデータを削除するために使われる手法を明記しなければならない。	6.4.12 節
B.43	FR6.45	それぞれの鍵タイプに対して、CKMS 設計は、どのメタデータが認可されたエンティティによってリスト化が可能かどうかを明記しなければならない。	6.4.13 節

解説・考慮点

＜メタデータの変更機能＞
認可されたエンティティが、暗号鍵と関連付けられている既存の書き込み可能なメタデータを変更するために使用する機能である。
項目 B.40 は、CKMS の設計にあたって、メタデータの変更機能が利用できる対象や条件、認可されていないエンティティの利用防止策等といったメタデータの変更機能への要求事項を明確化することを求めたものである。

＜メタデータの削除機能＞
認可されたエンティティが、暗号鍵に関連付けられたメタデータを削除するために使用する機能である。
CKMS の設計にあたって、項目 B.41 は、メタデータの削除機能が利用できる対象や条件、認

可されていないエンティティの利用防止策等といったメタデータの削除機能への要求事項を明確化することを求めたものである。B.42 は、具体的な削除方法の明確化することを要求したものである。

＜暗号鍵のメタデータリスト化機能＞
エンティティに認可されている暗号鍵のメタデータのリスト化を当該エンティティが実行するための機能である。
項目 B.43 は、CKMS の設計にあたって、メタデータリスト化機能が利用できる対象や条件といったメタデータリスト化機能への要求事項を明確化することを求めたものである。

前の要求事項でも記載したように、鍵タイプによってはメタデータとの組み合わせによりその暗号鍵の完全性を確保する場合もあるため、とりわけそのような鍵タイプの暗号鍵ではメタデータの管理も安全な鍵情報の管理を実施するために重要となる。つまり、暗号鍵だけでなく、メタデータも不用意に変更されたり削除されたりしないことが必要である。
そのためには、メタデータの完全性に影響するような処理を行えるエンティティを制限し、かつ認可されたエンティティであっても許可された範囲内、例えば自分の管理下にあるメタデータに対してのみ変更や削除、参照が行えるようにしておくことが望ましい。

項目 B.40 は変更権限を持つエンティティのみが決められた範囲内で変更機能を利用できるように、B.41 と B.42 は削除権限を持つエンティティのみが決められた範囲内で決められた方法による削除機能を利用できるように、それぞれのアクセス制御を行うために必要な要求事項の明確化を求めている。B.43 は、メタデータを一覧として参照できる権限を持つエンティティが決められた範囲内のメタデータのみ参照できるようにアクセス制御を行うために必要な要求事項の明確化を求めている。

《トイモデルと記載例》

本節のトイモデルも、2.1 節のトイモデルと同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムであり、暗号鍵のライフサイクルは 2.2 節のトイモデルで定めたものとする。
また、システムの運用条件を以下のように設定する。【】内は記載例のどの項目に影響を与えているのかを示している。

- 鍵生成はメール送信 PC において信頼できる方法で生成し、プライベート鍵はライフサイクル全般を通じて PC 外部に複製されることはない。【B.04, B.22, B23】
- CA から公開鍵証明書を受信したときに、当該証明書の正当性を確認する。【B.27, B.28】
- 公開鍵証明書の有効期間が始まった鍵は自動的に活性化状態となる。【B.04, B.06, B.07】
- 署名の生成や検証は、メール送受信 PC のメール管理部でのみ行う。【B.08】
- 署名付きメールを受信したとき、受信した公開鍵証明書の正当性を検証する。【B.04, B.27～B.29, B.36】

- 公開鍵証明書の有効期限切れした鍵は、自動的に非活性化状態になる。【B.04, B.09～B.11】
- 鍵の所有者が鍵を必要ないと判断したとき、手動で削除する。【B.04, B.20, B.21】
- 情報システム部を介してパブリックな CA 局に署名を依頼することで、信頼できる証明書チェーンを構築し、基本的なトラストアンカー管理は更新機能など OS の機能より実現する。【B.04, B.05, B.29～B.32, B.37】
- 公開鍵証明書の運用管理は、情報システム部の担当者が行う。【B.12, B.33～B.35】
- 鍵の危殆化が疑われるときは失効処理を行う。【B.04, B.13, B.14】
- 鍵のバックアップ、アーカイブは行わない。【B.04】
- 鍵とメタデータの関連付けは公開鍵証明書により行う（暗号学的プロセス）。【B.38, B.39】
- メタデータの変更・削除・リスト化は認めない。【B.11, B.33, B.40～B.43】
- 鍵の一時停止状態は設定しない。【B.04, B.15～B.19】
- 鍵導出機能や鍵更新機能は使用しない。【B.04, B.24, B.25】
- 対象鍵は使用しない。【B.04, B.26】

以上のトイモデルにおける記載例は、以下の「署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例」のようになる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.04	● メール送信者 PC の鍵生成機能により、署名プライベート鍵と署名公開鍵を生成する。 ● 公開鍵証明書の有効期間が開始したら、鍵活性化機能により、署名プライベート鍵と署名公開鍵を活性化状態にする。 ● メール送信者 PC の暗号機能により、メールのドキュメントに署名する。 ● メールを受信したら、メール送信者から送られてきたことを確認するために、メール受信者 PC の暗号機能により署名の完全性を確認する。 ● 公開鍵の検証機能により、署名公開鍵に対する公開鍵証明書に対する完全性を確認し、公開鍵及びパラメタの検証を行う。 ● OS でのトラストアンカー管理機能により、ルート CA の公開鍵証明書を保管・管理する。情報の更新は、OS 又はブラウザの自動アップデートにより実施する。 ● 公開鍵証明書の有効期間が終了したら、鍵非活性化機能により、署名プライベート鍵と署名公開鍵を非活性化状態にする。 ● 署名プライベート鍵と署名公開鍵は、破壊条件を満たした場合、破壊機能により、鍵を破壊する。 ● 署名プライベート鍵の危殆化が疑われるときは、鍵失効機能により、署名プライベート鍵の失効処理を行う。署名公開鍵についても同様の処理を行う。 ● 利用者の管理は情報システム部が行うものとし、そのために必要な管理機能は情報システム部管理の機器により実現する。 ● 暗号鍵とメタデータの検証及び関連付けについては、公開鍵証明書の申請段階で情報システム部がその正当性を検証するものとし、そのために必要な管理機能は情報システム部管理の機器により実現する。

	参考： ● 一時停止状態は設定しないので、一時停止機能及び再活性化機能は使わない。 ● 鍵生成では、鍵生成機能のみを利用するものとし、鍵導出機能／鍵更新機能は使わない。 ● 対称鍵の生成・利用は行わないので、対称鍵の検証機能は使わない。 ● 公開鍵の有効期間延長は認めないので、公開鍵の有効期間延長機能は使わない。 ● 署名プライベート鍵が利用できなくなることによる影響は当該鍵の所有者だけであり、影響が局所的であるので、バックアップとアーカイブの処理は実施せず、これらの機能は使わない。
B.05	● 署名公開鍵と関連メタデータの完全性は、公開鍵証明書の CA 署名検証により確認する。 ● 署名プライベート鍵の機密性は、OS のファイルアクセス機能により当該鍵を作成したユーザ以外が鍵ファイルにアクセスできないように管理することで実現する。 ● CA の署名公開鍵は OS の信頼できる公開鍵証明書（トラストアンカー）からなるチェーンの有効性により完全性を確認する。 ● OS の信頼できる公開鍵証明書（トラストアンカー）の更新は、OS 又はブラウザの自動アップデートにより実行される。
B.06	CA の署名した公開鍵証明書が、OS の信頼できる公開鍵証明書（トラストアンカー）からなるチェーンの有効性確認で正当であり、かつ記載された有効期間になったら、署名プライベート鍵と署名公開鍵は活性化状態に遷移する。
B.07	鍵活性化の通知は行わない。 メール送信者は CA が署名した公開鍵証明書の有効期間により活性化状態であることを認識できる。メール受信者は送信者の公開鍵証明書が有効期間内であることにより活性化状態であることを認識できる。
B.08	メールの署名は、メール送信 PC のメール管理部の管理下で実行される。 メールの署名検証は、メール受信 PC のメール管理部の管理下で実行される。
B.09	活性化状態の署名プライベート鍵と署名公開鍵は、公開鍵証明書の有効期間が終了すると非活性化状態へ遷移する。
B.10	署名プライベート鍵や署名公開鍵は、公開鍵証明書の有効期間が終了すると OS の機能及びメールアプリケーションの機能で、送信メールに署名ができなくなり、メールの署名検証も有効期限切れ表示になり、非活性化状態へ自動的に遷移する。
B.11	非活性化日時の変更は不可なので、対象外。
B.12	公開鍵証明書の有効期間が終了する一月前に、情報システム部の公開鍵証明書発行依頼の担当者が、メール送信者に証明書の有効期間の終了が近づいたので、新たに署名プライベート鍵と署名公開鍵を生成し、証明書を再申請するように通知する。
B.13	● 以下の状況のいずれかが発生した場合は失効処理を行う。 ➢ PC が保護されない状況で PC 利用者以外のエンティティがアクセス可能であった場合

	➢ 署名プライベート鍵に不正なアクセスがあったことを示すログが検出された場合 ➢ メール送信者が認識していない署名付きメールの送信ログが確認された場合 ➢ その他、署名プライベート鍵の紛失・漏えいが疑われる事象が発生した場合 ● 上記の事象が生じた場合、該当する署名プライベート鍵の所有者は情報システム部に連絡し、情報システム部は当該署名プライベート鍵に対応する署名公開鍵の公開鍵証明書の失効処理を CA に依頼する。PC 利用者は当該署名プライベート鍵の使用を停止する。 ● メール受信者は B.14 の方法により配布される証明書失効リストを受信したら、OS の機能により該当する証明書を失効させる。
B.14	公開鍵証明書の失効を依頼された CA は、当該証明書に対する証明書失効リストを作成する。 通知は、CA から得た証明書失効リストを関係するエンティティに配布することにより行う。また、証明書に記載された OCSP にアクセスすることで失効情報を得ることも可能である。
B.15	B.02 の通り、一時停止状態への遷移は設けないため、対象外。
B.16	同上
B.17	同上
B.18	同上
B.19	同上
B.20	署名プライベート鍵・署名公開鍵が非活性化状態になって一定期間が経過し鍵を必要とするケースはないと判断したとき、または危殆化状態の署名プライベート鍵・署名公開鍵について危殆化状態での管理を行う必要がないと判断したとき、署名プライベート鍵を OS の機能を使用して当該鍵の所有者が手動で削除することで当該鍵の破壊を行う。署名プライベート鍵の破壊は局所的（Local）なものである。 署名公開鍵については公開鍵証明書にて管理されており、公開鍵証明書の有効期間の経過後、OS の証明書管理機能により自動的に削除され、破壊される。署名公開鍵の破壊は局所的（Local）なものである。
B.21	鍵の破壊は B.20 で慎重に判断するので、通知は必要としないため、対象外。
B.22	署名プライベート鍵及び署名公開鍵の鍵生成は FIPS 186-4 で規定されている方法で生成する。
B.23	乱数生成器は SP 800-90 で規定されている方法を使用し、乱数生成器で使用するエントロピーは Linux の乱数生成の疑似デバイスである/dev/random から得る。
B.24	鍵導出機能や鍵更新機能は使用していないため、対象外。
B.25	同上
B.26	対称鍵は利用しないため、対象外。
B.27	● CA から公開鍵証明書を受信したとき、受信した PC で、署名公開鍵が FIPS 186-4 で規定されたドメインパラメタを使用していることを検証する。

	● 署名付きメールを受信したとき、受信した PC で、一緒に受信した公開鍵証明書に記載された署名公開鍵が FIPS 186-4 で規定されたドメインパラメタを使用していることを検証する。
B.28	● CA から公開鍵証明書を受信したとき、受信した PC で、その証明書に記載された署名公開鍵が、生成した署名プライベート鍵とペアになる公開鍵であることを検証する。 ● 署名付きメールを受信したとき、受信した PC で、一緒に受信した公開鍵証明書を検証することにより、公開鍵を検証する。
B.29	● CA から公開鍵証明書を受信したとき、受信した PC で、当該証明書について OS の信頼できる証明書パスを検証する。 ● 署名付きメールを受信したとき、受信した PC で、一緒に受信した公開鍵証明書について OS の信頼できる証明書パスを検証する。 ● 証明書失効リストを受信したときに、受信した PC で、当該リストを検証するために必要となる公開鍵証明書について OS の信頼できる証明書パスを検証する。
B.30	トラストアンカー管理機能は OS の証明書管理機能を使用する。
B.31	トラストアンカーの配布は独自に行わない。OS に標準で保存されているトラストアンカーを使用する。トラストアンカーは信頼できる OS の更新機能により自動更新する。
B.32	トラストアンカーは信頼できる OS の更新機能により、自動的に追加、変更、削除を行う。緊急にトラストアンカーを修正したい場合、システム管理者が管理者権限により、OS の証明書管理機能で追加、変更、削除を行う。一般の利用者による追加、変更、削除はできない設定にする。
B.33	公開鍵の有効期間の延長は認めないため、対象外。
B.34	同上
B.35	メール送信者の CSR は組織内の情報システム部の公開鍵証明書発行依頼の担当者が所有者情報などを目視確認後、情報システム部から CA に CSR を送信し、CA から受信した公開鍵証明書は情報システム部が受信し、公開鍵証明書発行依頼の担当者が確認後にメール送信者に送られる。
B.36	メール受信者は、メールの署名を検証することで、一緒に受信した公開鍵証明書に記載された公開鍵とペアになる署名プライベート鍵をメール送信者が所持していることを検証する。
B.37	新規公開鍵証明書の発行のために情報システム部の公開鍵証明書発行依頼の担当者が CSR を受け取ったときに、その担当者は CSR に保存されたメタデータと署名を確認し、CSR 作成者がプライベート鍵の所有者であることを確認する。
B.38	鍵の有効期間、所有者情報（Subject）、key usage などのメタデータが、署名公開鍵に関連付けられ、署名公開鍵と一緒に公開鍵証明書に記載される。これらのメタデータと署名プライベート鍵とは、対応する署名公開鍵を介して関連付けられる。
B.39	署名公開鍵に関連付けられたメタデータは、公開鍵証明書に記載され、CA の署名により完全性が保護される。署名プライベート鍵は、ペアとなる署名公開鍵の保証により、完全性の関連付けが保証される。

B.40	メタデータの変更は認めないため、対象外。
B.41	メタデータの削除は認めないため、対象外。
B.42	同上
B.43	鍵メタデータのリスト化は認めないため、対象外。

2.4　鍵情報の保管方法

①　保管中の鍵情報のセキュリティを確保するための手段の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.44	FR6.73	CKMS 設計は、鍵情報（暗号鍵やメタデータ）をストレージに入れるエンティティの ID 認証及び認可検証に使用される手段を明記しなければならない。	6.5 節
B.45	FR6.74	CKMS 設計は、ストレージに入力する鍵情報（暗号鍵やメタデータ）の完全性検証に使用される手段を明記しなければならない。	6.5 節
B.46	FR6.75	CKMS 設計は、保管された対称鍵、プライベート鍵及びメタデータの機密性保護に使用される手段を明記しなければならない。	6.5 節
B.47	FR6.76	鍵ラッピング鍵（又は鍵ペア）が保管された鍵を保護するために使用される場合、CKMS 設計は、鍵ラッピング鍵（又は鍵ペア）を保護し、その使用を制御するために使用される手段を明記しなければならない。	6.5 節
B.48	FR6.77	CKMS 設計は、保管された鍵情報（暗号鍵及びメタデータ）の完全性保護に使用される手段を明記しなければならない。	6.5 節
B.49	FR6.78	CKMS 設計は、保管された鍵へのアクセスがどのように制御されるかを明記しなければならない。	6.5 節
B.50	FR6.79	CKMS 設計は、全ての保管された鍵を訂正又は復元するために使用される手法を明記しなければならない。	6.5 節

解説・考慮点

鍵情報の保管にあたっては、全ての暗号鍵は完全性保護を、加えて対称鍵及びプライベート鍵は機密性保護も必要とする。
また、認可されたユーザのみが保管された鍵情報にアクセスできるようにすべきであり、そのためのアクセスコントロールが必要である。
CKMS の設計にあたって、項目 B.44 及び B.45 は鍵情報を保管する時点での完全性保護を実現するための要求事項を、B.46～B.49 は保管中の鍵情報の完全性及び機密性の保護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。B.50 は、保管中の鍵情報が破損した際

の対策を明確化することを要求したものである。

暗号アルゴリズムを安全に使う上で、(i) 利用権限を有する正しいエンティティ（利用者）のみが暗号鍵を利用できること、(ii) 暗号鍵の完全性が確保されていること、(iii) 加えて対称鍵及びプライベート鍵は機密性が確保されていること、は絶対条件である。また、暗号鍵の可用性の観点では、(iv) 利用権限を有する正しいエンティティ（利用者）は、暗号鍵の有効期間内はいつでも当該暗号鍵が利用できること、が求められる。

したがって、暗号鍵の保管にあたって、上記の要件を満たすように、鍵情報のセキュリティを確保するための手段を決める必要がある。なお、ここで決める手段は、後述の②以降で利用する手法を実現するものでなければならないことに留意されたい。

a）当該暗号鍵をストレージなどに保管する時：

暗号鍵を保管する時点で、「偽物の暗号鍵」が正しく保管されてしまうと、その後の保護手段がいくら強固で正しく動作したとしても全く意味をなさない。そのようなことが起きないようにするには、「格納権限がある信頼できるエンティティ（利用者）」によって「正しい暗号鍵」が「正しく保管」されることが重要である。つまり、「格納権限がある信頼できるエンティティ(利用者)」であることを確認するための認証認可機能、「正しい暗号鍵」であることを確認するための完全性検証機能、さらに対称鍵やプライベート鍵では秘密裏に「正しく保管」するための機密性保護機能が求められる。これらの機能を実現する具体的な手段を決めるよう求めたものが検討課題 B.44～B.46 に該当する。

検討課題 B.47 は、暗号鍵の機密性保護のために当該暗号鍵の再暗号化を行う場合、再暗号化で利用する暗号鍵（鍵ラッピング鍵）も安全に管理・運用されていることが必要となるので、そのための必要となる機能を実現する具体的な手段を決めるよう求めている。

b）ストレージなどに保管された当該暗号鍵を利用する時：

(i) の要件を満たすためには、利用権限を有する正しいエンティティ（利用者）のみが保管された暗号鍵にアクセスできることを保証するためのアクセス制御機能・認証認可機能が求められる。この機能を実現する具体的な手段を決めるよう求めたものが検討課題 B.49 となる。

また、ストレージなどでの保管中も、誤操作や故障、改ざん攻撃などによって、暗号鍵の完全性が損なわれないように保護することが必要である。例えば、誤り検知や攻撃検知などによって予期せぬ理由で暗号鍵が書き変わらないようにする、利用前には読み出した暗号鍵の完全性の検証を行う、などの対策が考えられる。このような、保管中の鍵情報の完全性保護機能を実現する具体的な手段を決めるよう求めたものが検討課題 B.48 である。

検討課題 B.50 は、何らかの理由で保管された暗号鍵の完全性が損なわれた場合に、当該暗号鍵を正しい状態に復旧するための取られる方法を具体的に決めるよう求めたものである。例えば、誤りが検知された暗号鍵について訂正可能であれば適切な訂正を行い、訂正不能な場合には当該暗号鍵はしないようにするとか、暗号鍵のバックアップやアーカイブを行い、必要に応じて暗号鍵の復元に行うなどが考えられる。

最近はクラウドサービスを利用するケースも増えており、通常、クラウド上にあるデータに対する暗号処理もクラウド内で行われる。その際に利用する暗号鍵の管理・保管方法として、クラウド事業者に完全に任せる方式からクラウド利用者が自ら管理する方式までいくつかのやり方がある。代表的な方法として、クラウド事業者が全ての鍵管理を行う方式（Cloud Native Encryption Services）、鍵生成と管理はクラウド利用者が行うが暗号鍵の保管はクラウド事業者で行う方式（BYOK：Bring Your Own Key[9]）、クラウド利用者が全ての鍵管理を行う方式（HYOK：Hold Your Own Key[10]）などがある。

これらの方式のうち、どのような方式を取るかによって暗号鍵に対するクラウド利用者の管理度合いやクラウド事業者に求めるセキュリティレベルが変わる。一般に、クラウド事業者に鍵管理を依存する方式になるほど暗号鍵管理に関する負荷を低減させることができる一方、その安全性はクラウド事業者に依存するようになる。反対に、クラウド利用者が自ら鍵管理を行う方式になるほど、安全性をクラウド事業者に依存しなくてすむようになるが、オンプレミスで暗号鍵を管理するのと同様の負荷が求められるようになる。

これらの詳細については、日本クラウドセキュリティアライアンスが発行している Cloud Data Protection[11]を参照されたい。

項目 B.44～B.50 は、暗号鍵の管理を CKMS 運用者が自ら実施することを前提としたときの暗号鍵の保管方法について CKMS 設計で考慮すべき項目となっている。したがって、クラウドサービスを利用する場合には、どの部分の暗号鍵の管理をクラウド事業者に任せ、どの部分をクラウド利用者自ら管理するのかを切り分けること（暗号鍵管理についての責任分界点を明確化すること）が最初にすることである。そのうえで、クラウド利用者自ら管理する必要がある部分については他と同様に検討し、クラウド事業者に任せる部分についてはクラウド事業者が提供する機能や利用するサービス内容などをわかる範囲で記載すればよい。

例えば、クラウド事業者が全ての鍵管理を行う方式であれば、B.44 や B.49 のアクセス制御に係る部分を重点的に検討し、利用権限を有する正しいエンティティ（利用者）のみが暗号鍵を使えるようにすればよい。その他の項目については、クラウド事業者に管理を委ねる部分になる。

一方、BYOK の場合は、B.44 と B.49 に加え、クラウド側に暗号鍵を移す際の方法（B.45）や暗号鍵の復元方法（B.50）についても検討することが必要となる。

---

9 利用者が暗号鍵を作成してクラウドサービスの CKMS に持ち込む方式

10 利用者が管理する CKMS をクラウドサービスが利用する方式。オンプレミスで実現する場合やクラウド事業者が機能提供する場合、その併用など、いろいろな実現形態がある。Microsoft Azure Double Key Encryption、Google Cloud External Key Manager、Salesforce Cache-Only Key、Microsoft Azure Dedicated HSM、AWS CloudHSM など

11 日本クラウドセキュリティアライアンス、Cloud Data Protection、https://www.cloudsecurityalliance.jp/site/wp-content/uploads/2021/08/Cloud_Data_Protection2_V10.pdf

②　運用中の鍵情報の保管場所及び保護方法の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.51	FR6.46	それぞれの鍵タイプに対して、CKMS 設計は、以下のことを明記しなければならない：それぞれの鍵タイプとそのメタデータが保管される状況、鍵とメタデータの保管場所、及び鍵とメタデータの保護方法。	6.4.14 節

解説・考慮点

運用中の鍵情報がどこに存在し、どのような保護状態に置かれているのかを全て明らかにしておく必要がある。
項目 B.51 は、CKMS の設計にあたって、鍵情報の保管場所や保護方法などの要求事項を明確化することを求めたものである。運用中の鍵情報は、B.51 で定めた保管場所以外に置かれないようにしなければならない。

運用中の暗号鍵は、ストレージなどに保管された状態から読み込まれ、メモリ上などに保存された状態で暗号処理に使われるのが一般的である。その際、暗号処理のスループットを上げるため、暗号鍵が平文の形で置かれることが多く、暗号鍵の漏えいリスクが高い場所の一つになっている。

そのため、暗号鍵の漏えいリスクの低減策として、運用中の暗号鍵がどこに存在し、どのように保護されているのかを把握しておくことが重要であり、項目 B.51 はそれらの情報を具体的に決めるよう求めたものである。例えば、一言に「メモリ上に保存される」といっても、他のアプリケーションなどからも読み出せる汎用のメモリ上に置かれるのか、外部からの侵入が厳しく制限される暗号モジュール内のメモリ上に置かれるのかによって、その暗号鍵が置かれている保護状態は全く異なる。具体的には、前者は OS レベルでのアクセス制御による保護であるのに対して、後者は暗号モジュールが提供するアクセス制御のほか、当該暗号モジュールへの侵入検知による物理的防護などの保護が受けられる可能性がある。

なお、クラウドサービスでの運用中の鍵情報の管理方法について検討する必要があるかどうかは、暗号鍵の保管方法と同様、暗号鍵を管理する主体がクラウド事業者なのかクラウド利用者なのかによって異なる。つまり、前者の場合、運用中の暗号鍵を安全に管理するのはクラウド事業者の責任となるので項目 B.51 は対象外としてよい。一方、後者の場合は、運用中の暗号鍵を安全に管理するのはクラウド利用者の責任となるので B.51 も検討しておく必要がある。

③　鍵情報のバックアップ方法の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.52	FR6.47	CKMS 設計は、どのように、どこで、どのような状況において鍵及びそのメタデータがバックアップされるかを明記しなければならない。	6.4.15 節
B.53	FR6.48	CKMS 設計は、バックアップされた鍵情報（暗号鍵及びメタデータ）の保護のためのセキュリティポリシーを明記しなければならない。	6.4.15 節
B.54	FR6.49	CKMS 設計は、鍵情報（暗号鍵及びメタデータ）のバックアップ中のセキュリティポリシーがどのように実装されるかを明記しなければならない。例えば、バックアップされた鍵情報（暗号鍵及びメタデータ）の配送及び保管中における、機密性とマルチパーティコントロールの要求事項の実装方法。	6.4.15 節

解説・考慮点

CKMS の設計にあたって、項目 B.52 は、鍵情報のバックアップを行うための条件を明確化することを、B.53 及び B.54 はバックアップされる鍵情報のセキュリティ、特に機密性保護を確保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。
なお、鍵情報のバックアップを実施しない場合には、検討対象外である。

運用中の暗号鍵が喪失、改変、又はその他の理由で利用不能状態になったときに、当該暗号鍵を復元できるようにするため、安全な設備・メディアなどにバックアップをする場合がある。バックアップをすることにより、暗号鍵が喪失、改変又はその他の理由で利用不能になった場合であっても、当該暗号鍵で保護されている情報まで喪失する事態を避けることが可能になる。一方、暗号鍵の複製を作ることでもあるので、暗号鍵の漏えいリスクを高めることにつながる。

したがって、ここでの項目を検討する前に、まずはバックアップを行うことによるメリットとデメリットを天秤にかけて、暗号鍵のバックアップを行うかどうかを決定することが重要である。もしバックアップは行わないと決定した場合には、ここでの項目 B.52～B.54 は対象外となる。

一方、バックアップを行うと決定した場合には、できる限り、暗号鍵の複製を作ることによる漏えいリスクを低減する対策を検討する必要があり、そのためには、項目 B.52～B.54 の内容に沿って具体的なバックアップの条件や実施方法などを取りまとめておくことが重要である。特に、B.52 はバックアップを実施する条件を、B.53 はバックアップされる暗号鍵の保護方針を明確化することを求めており、バックアップを行う上での全体方針を決めるものである。この方針によって、バックアップのセキュリティが決まると言っても過言ではない。

B.54 は、B.53 の方針に沿った具体的なバックアップの実現方法を明確化することを求めている。

なお、B.52～B.54 では明確には求められていないが、バックアップした暗号鍵が使用されるこ

とがなくなったとき、当該暗号鍵は復元できないように破壊されるべきである。バックアップストレージメディアに保管されている場合にはメディア内の暗号鍵を、コピーが存在する場合には当該コピーも含めて破壊することを、B.53 でのセキュリティポリシーの中に明記することを強く推奨する。

④　鍵情報のアーカイブ方法の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.55	FR6.50	CKMS 設計は、どのように、どこで、どのような状況で鍵情報（暗号鍵やメタデータ）がアーカイブされるかを明記しなければならない。	6.4.16 節
B.56	FR6.51	CKMS 設計は、鍵情報（暗号鍵やメタデータ）のセキュアな破壊、又は新しい保管メディアに書き込まれた後の古い保管メディアのセキュアな破壊のための手法を明記しなければならない。	6.4.16 節
B.57	FR6.52	CKMS 設計は、アーカイブ鍵の暗号鍵有効期間（cryptoperiod）の期限切れ後に、鍵情報（暗号鍵やメタデータ）がどのように保護されるかを明記しなければならない。	6.4.16 節

解説・考慮点

CKMS の設計にあたって、項目 B.55 は、鍵情報のアーカイブを行うための条件を明確化することを、B.56 はアーカイブされた鍵情報を破棄するための要求事項を明確化することを、B.57 は機密性保護の継続性を確保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。
なお、鍵情報のアーカイブを実施しない場合には、検討対象外である。

バックアップが、「運用中の暗号鍵」が喪失、改変又はその他の理由で利用不能状態になった時の「復元対策」であるのに対して、アーカイブは、「運用中の暗号鍵かどうかに関わりなく」、法律や規則等で要求される期間中、必要に応じて、当該暗号鍵を復元できるようにしておく「長期保管対策」である。

バックアップもアーカイブも、暗号鍵の複製を作ることで漏えいリスクを高めることにつながるという点では同じであるが、バックアップが比較的頻繁に復元を行うことが想定されているのに対して、アーカイブは長期保管用ストレージ設備などに保管され、限定的な条件下でのみ復元されることが想定されている点が異なる。

アーカイブは、適用される法律や規則等も考慮して最小限の範囲で実施し、暗号鍵の複製を作ることによる漏えいリスクを低減する対策を検討する必要がある。

なお、バックアップと比較して、復元頻度が少なく、長期保管が想定されることから、可用性よりも機密性保護を優先して対策を考えるべきである。例えば、鍵分割による保護、アーカイブ鍵による暗号鍵の再暗号化、物理的に保護される装置内での保管、などがある。また、アーカイ

ブする必要がなくなれば、当該暗号鍵が復元できないようにそのアーカイブは破壊されるべきである。

項目 B.55 は、暗号鍵のアーカイブを行うための条件や保護方針を明確化することである。

鍵情報のアーカイブを実施しない場合には、ここでの項目 B.55～B.57 は対象外となる。一方、アーカイブを実施する場合には、アーカイブされた暗号鍵は、アーカイブされている期間、物理的又は暗号学的に保護されなければならない。また、保管メディアにアーカイブする作業中に、アーカイブする暗号鍵が露見しないように対策を取る必要がある。これらの方針を明確化することにより、アーカイブを行う上での全体方針を決めるものである。この方針によって、アーカイブのセキュリティが決まると言っても過言ではない。

B.56 は、アーカイブされた暗号鍵、もしくは当該暗号鍵がアーカイブされている保存媒体そのものを破壊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。また、B.57 は、アーカイブ鍵による暗号鍵の再暗号化によって機密性保護を行っているケースにおいて、アーカイブ鍵のほうが先に有効期間切れになった時に、機密性保護の継続性を確保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。例えば、アーカイブ鍵の暗号鍵有効期間が期限切れになる前の再暗号化や、新しいセキュアな保存メディアへの移動などである。

⑤　鍵情報の復元方法の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.58	FR6.53	CKMS 設計は、鍵情報（暗号鍵やメタデータ）の CKMS 復元ポリシーを明記しなければならない。	6.4.17 節
B.59	FR6.54	CKMS 設計は、鍵情報（暗号鍵やメタデータ）の復元ポリシーを実装及び実行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。	6.4.17 節
B.60	FR6.55	CKMS 設計は、どのように、どのような状況で鍵情報（暗号鍵やメタデータ）がそれぞれの鍵データベース又はメタデータ保管設備から復元されるかを明記しなければならない。	6.4.17 節
B.61	FR6.56	CKMS 設計は、鍵情報（暗号鍵やメタデータ）が復元中にどのように保護されるかを明記しなければならない。	6.4.17 節

解説・考慮点

CKMS の設計にあたって、項目 B.58～B.61 は、バックアップやアーカイブされた鍵情報を復元するための条件や復元方法、要求事項を明確化することを求めたものである。

バックアップやアーカイブされている暗号鍵を復元することは、当該暗号鍵のコピーを利用可能にする行為であるから、万が一にも不適正に復元が行われれば、即、当該暗号鍵の漏えいにつ

ながるほど危険な行為となる。したがって、バックアップやアーカイブからの復元が不適正に行われることがないように、厳格な復元ルールを規定し、そのルールが全て満たされていることを検証された後に、認可されたエンティティ（利用者）によって復元できるようにすべきである。

項目 B.58 は、復元ポリシーとして復元を実行するために必要なルールを定めることを求めている。B.59 はそのルールを守られるように具体的にどのように実装し、実行するのかを具体化すること、B.60 は復元処理を行う際の保管メディアへのアクセス条件を明確化することを求めている。また、B.61 は、予定外の場所で暗号鍵が露見しないような対策を求めたものである。

なお、ここで実現されるセキュリティ水準は、バックアップやアーカイブで実現されるセキュリティ水準と整合的であることが重要である。また、バックアップとアーカイブのどちらも実施しない場合には、本項目 B.58～B.61 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、2.1 節のトイモデルと同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムであり、暗号鍵のライフサイクルは 2.2 節のトイモデルで定めたものとする。

なお、このトイモデルでは、暗号鍵が喪失、改変、又はその他の理由で利用不能状態になったとしても影響範囲が限定的（署名生成ができなくなるだけでその他に影響しない）であるため、鍵情報のバックアップとアーカイブのどちらも実施しないこととする。鍵情報の喪失や破損が発生したときは、新たな暗号鍵を再生成し公開鍵証明書を発行し直す。このため、バックアップとアーカイブに関連する項目は対象外となる。

また、暗号鍵の保管に関する運用条件を以下のように設定する。【】内は記載例のどの項目に影響を与えているのかを示している。

- 鍵情報の機密性保護や完全性保護は、OS のアクセスコントロールシステム（ACS: Microsoft Windows の ACL、Linux のパーミッション機能）により保護する。【B.44, B.46, B.48】
- ストレージに入力する鍵情報において、署名プライベート鍵は証明書の公開鍵とペアになっていることを確認し、メタデータは証明書の内容と一致することを確認する。【B.45】
- OS の ACS により、基本的に鍵を生成したユーザ以外、鍵情報にアクセスできない。【B.49, B.51】
- 暗号鍵のバックアップとアーカイブは実施しない。【B.52～B.61】
- 保管していた鍵が使用できなくなった場合の復元手段は用意しない。【B.50】
- 保管された鍵を保護するために鍵ラッピング鍵や鍵ペアは使用しない。【B.47】

以上のトイモデルにおける記載例は、以下の「署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例」のようになる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.44	PC にログイン ID 及びパスワードを設定し、その ID 及びパスワードによる利用者認証を通過したエンティティに対してのみ、OS のアクセスコントロールシステム（ACS: Microsoft Windows の ACL、Linux のパーミッション機能）による自分が管理する鍵情報へのアクセス権限を付与する。 なお、システム管理者は、管理者 ID 及び管理者パスワードによる利用者認証でログインした場合に限り、例外的に全ての利用者の鍵情報に対してアクセス権限（Microsoft Windows の管理者権限、Linux の root 権限）が付与される。
B.45	ストレージに入力する署名プライベート鍵は、公開鍵証明書に記載された公開鍵とペアになっていることを確認し､当該証明書の CA 署名を検証することで完全性を検証する。
B.46	署名プライベート鍵と関連するメタデータは B.44 に記載した ACS により機密性を保護する。
B.47	保管された鍵を保護するために鍵ラッピング鍵や鍵ペアを使用しないため、対象外。
B.48	署名プライベート鍵と関連するメタデータは B.44 に記載した ACS により完全性を保護する。
B.49	署名プライベート鍵は B.44 に記載した利用者認証及び ACS を使用することで、システム管理者を除き、当該鍵を作成したユーザ以外はアクセスできない。
B.50	署名プライベート鍵の訂正や復元するための手段は用意しないため、対象外。
B.51	署名プライベート鍵、署名公開鍵や CA 署名公開鍵は署名するユーザの PC の内臓ストレージに保管する。これらの鍵は B.44 に記載した ACS で保護される。また、署名や検証するときに一時的に当該 PC のメモリ上に置かれるが、これは OS のプロセス間メモリ保護機能により保護される。
B.52	鍵情報のバックアップは実施しないため、対象外。 署名プライベート鍵や署名公開鍵が利用不能になった場合、その鍵の失効処理を行う。また、必要に応じて、新たな鍵ペアを再生成し公開鍵証明書を発行する。
B.53	同上
B.54	同上
B.55	鍵情報のアーカイブは実施しないため、対象外。
B.56	同上
B.57	同上
B.58	鍵情報のバックアップとアーカイブのどちらも実施しないため、対象外。
B.59	同上
B.60	同上
B.61	同上

2.5　鍵情報の鍵確立方法

①　鍵確立機能の利用局面の特定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.62	FR6.57	CKMS 設計は、どのように、どのような状況で鍵及びそのメタデータが確立されるかを明記しなければならない。	6.4.18 節

解説・考慮点

項目 B.62 は、CKMS の設計にあたって、鍵確立機能を利用する状況を特定することを要求したものである。

鍵確立（key establishment）機能とは、2 つ又はそれ以上のエンティティ間で暗号鍵をセキュアに共有するプロセスのことであり、方法として以下の 2 つがある。

- 鍵配送（key transport）:
  一方のエンティティが共有する暗号鍵を生成し、当該暗号鍵及び（あれば）メタデータを他方のエンティティに配付する。
- 鍵合意（key agreement）:
  両方のエンティティが共有鍵を導出するために使用される情報を共有し、当該情報から暗号鍵を導出する。

鍵確立が行われるタイミングは、「盗聴」という手段－すなわち、鍵確立に係る正当なエンティティに検知されない手段－で第三者が暗号鍵を窃取できる唯一のタイミングである。したがって、鍵確立を行うとき、(i) 確立される暗号鍵が誤りなく、正しく共有されること（暗号鍵の完全性）が求められるだけではなく、(ii) 関係するエンティティが全員正当であることの確認と（iii）セキュアな通信路での通信による暗号鍵の機密性保護が極めて重要である。

項目 B.62 は、CKMS の設計にあたって鍵確立機能を利用する状況を特定することで、鍵確立がいつ、どのように行われるのかを把握し、以降の②～⑤についての検討を忘れないようにすることを目的としている。②～⑤はいずれも（i）～（iii）の目的を達成するために必要な要求事項に関連するものである。

なお、鍵情報の鍵確立を使う状況がなければ、2.5 節の全て、すなわち B.62～B.70 全てが検討対象外となる。

②　鍵配送における鍵情報のセキュリティを確保するための要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.63	FR6.80	CKMS 設計は、配送中の対称鍵及びプライベート鍵の機密性保護に使用される手段を明記しなければならない。	6.6.1 節
B.64	FR6.81	CKMS 設計は、配送された鍵の完全性保護に使用される手段、及びエラー検出後にどのように鍵が再構築又は置き換えられるのかを明記しなければならない。	6.6.1 節
B.65	FR6.82	CKMS 設計は、配送される鍵素材（keying material）の鍵受信者に、どのように鍵送信者の識別子（ID）が認証されるかを明記しなければならない。	6.6.1 節

解説・考慮点

CKMS の設計にあたって、項目 B.63 は鍵配送における機密性保護のための要求事項を、B.64 は完全性保護のための要求事項を、B.65 は鍵送信者を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。
なお、鍵情報の鍵配送を使う状況がなければ検討対象外である。

鍵情報の鍵配送には、郵便・宅配便などの物理的手段を使う場合と、ネットワークを介する電子的手段を使う場合がある。いずれの手段であっても、全ての暗号鍵は完全性保護を、加えて対称鍵及びプライベート鍵は機密性保護も必要とする。

機密性保護の観点からは、物理的保護が行われる手段か、対称鍵ラッピング鍵又はひとつ以上の非対称配送鍵ペアが関わる鍵配送手段が使用される。前者であれば信頼できる仲介者が必要であり、後者であれば当該ラッピング鍵や配送鍵が配送に関わるエンドエンティティによって保護されたうえで信頼できる鍵配送手段が必要となる。項目 B.63 はこれらの要件に対応するために利用する手段を明確化することを目的としたものである。

完全性保護の観点からは、暗号鍵の送信者が信頼できることと、暗号鍵に改ざんやエラーがないことが求められる。したがって、配送された暗号鍵の受信者に対して、期待する認可された鍵送信者から当該暗号鍵が来たことを保証できることが必要である。また、暗号鍵の完全性検証を実行し、訂正可能な破損が検出された場合には適切な訂正を行い、訂正不能な破損が検出された場合には使用前に新しい又は訂正された暗号鍵を再確立する必要がある。
項目 B.64 は暗号鍵に改ざんやエラーがないことを確認するために利用する手段を、B.65 は鍵送信者が正当であることを鍵受信者が確認できるための方法を、それぞれ明確化することを目的としたものである。

なお、鍵情報の鍵配送を使う状況がなければ、B.63～B.65 は検討対象外である。

③　鍵合意における鍵情報のセキュリティを確保するための要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.66	FR6.83	CKMS 設計は、CKMS にサポートされるそれぞれの鍵合意スキームを明記しなければならない。	6.6.2 節
B.67	FR6.84	CKMS 設計は、鍵合意に参加するそれぞれのエンティティがどのように認証されるかを明記しなければならない。	6.6.2 節

解説・考慮点

CKMS の設計にあたって、項目 B.66 は鍵合意プロセスの手法について明確化することを、B.67 はエンティティの認証方法について明確化することを要求したものである。
なお、鍵情報の鍵合意を使う状況がなければ検討対象外である。

セキュアな鍵合意プロセスを利用する場合、そのプロセスに関与するそれぞれのエンティティは合意鍵を導出するために使われるある種の情報を提供しあうことで自ら合意鍵を生成することができるが、当該プロセスに関与していないエンティティは提供しあっている情報全て得たとしても合意鍵を得ることができない。逆に言えば、不正なエンティティに合意鍵を窃取されないためには認可されていないエンティティが鍵合意プロセスに不正に入り込むことを防止することが絶対条件となる。そのため、典型的には、鍵合意プロセスに参加する各エンティティは他方のエンティティ識別子の保証を必要とする。

項目 B.66 は鍵合意プロセスの手法について明確化することでセキュアな鍵合意プロセスであることを把握するため、また B.67 はエンティティの認証方法について明確化することで不正なエンティティに入り込むのを防止することを目的としたものである。

なお、鍵情報の鍵合意を使う状況がなければ、B.66 と B.67 は検討対象外である。

④　鍵確認機能を利用するための要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.68	FR6.86	CKMS 設計は、それぞれの鍵確認が実行される状況を明記しなければならない。	6.6.3 節
B.69	FR6.85	CKMS 設計は、他方のエンティティと正しい鍵を確立したことを確認するために使用されるそれぞれの鍵確認手段を明記しなければならない。	6.6.3 節

解説・考慮点

CKMS の設計にあたって、項目 B.68 は鍵確認を行うための条件を明確化することを、B.69 は

鍵確認の手法について明確化することを求めたものである。
なお、鍵確立した鍵情報の鍵確認を行う状況がなければ検討対象外である。

鍵確認機能は、鍵確立機能で共有された暗号鍵について、それぞれのエンティティが、実際に他方のエンティティが正しい暗号鍵を確立したことの確認をするために使用する機能である。

実際の暗号処理で利用する前に共有された暗号鍵を鍵確認することにより、何らかの理由で誤った暗号鍵が生成されたり、暗号鍵がうまく共有できなかったりした場合であっても、実害が発生する前に当該暗号鍵の利用を止めることが可能となる。

項目 B.68 はどのような状況のときに鍵確認が行うのかを把握することを、B.69 は鍵確認の手法について明確化することを求めたものである。

なお、鍵確立した鍵情報の鍵確認を行う状況がなければ、B.68 と B.69 は検討対象外である。

⑤　利用する鍵確立プロトコルの決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.70	FR6.87	CKMS 設計は、鍵確立と保管の目的のために CKMS によって採用されている全てのプロトコルを明記しなければならない。	6.6.4 節

解説・考慮点

項目 B.70 は、CKMS の設計にあたって、利用する鍵確立プロトコルを全て明確化することを要求したものである。ここで、定められた以外の鍵確立プロトコルを利用してはならない。

鍵確立で利用する通信路はセキュアであることが求められるので、CKMS 設計ではセキュアなプロトコルであることが確認されたものだけを使うべきである。項目 B.70 は、このことを確認する目的で、利用する鍵確立プロトコルを全て明確化することを求めたものである。

セキュアな鍵確立プロトコル、あるいは鍵確立機能を含んだプロトコルとして、代表的なものとして以下のものある。

- Internet Key Exchange（IKE）
- Transport Layer Security（TLS）
- Kerberos
- Over-The-Air-Rekeying（OTAR）Key Management Messages
- Secure Shell（SSH）

なお、鍵確立プロトコル自体は相互接続性の要求が強く求められることもあることから、「必ずしもセキュアとは言えないが相互接続性を実現するために必要」とされる手順や暗号アルゴリズ

ムもデフォルトで選択できるようになっていることがある。しかし、これらの手順や暗号アルゴリズムは真に必要な場合を除いて利用するべきではないので、意図せずに誤って利用することがないよう、CKMS 設計の段階でデフォルトでは使えないように設定しておき、真に必要な場合には「例外として意図的に設定変更する」ようにすることが重要である。

鍵確立プロトコルでの設定については、安全性と相互接続性のバランスを踏まえた推奨の設定ガイダンスが公開されているものも多い。そのようなガイダンスでは、「相互接続性を実現するために含まれたセキュアとは言えない手順や暗号アルゴリズム」はデフォルトでは使えないようにするための設定オプションが記載されている。

したがって、それらのガイダンスが存在する場合にはその設定に従うことで、セキュアな鍵確立プロトコルを確保できる。例えば、「TLS 暗号設定ガイダンス」や「SP 800-57 Part 3」などを参照されたい。

《トイモデルと記載例》

2.1 節と同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムであり、暗号鍵のライフサイクルは 2.2 節で定めたトイモデルでは、暗号鍵の鍵配送及び鍵合意は使用しないため、B.62～B.70 は対象外となる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.62	B.02 の通り、暗号鍵の鍵配送及び鍵合意は使用しないため、対象外。
B.63	同上
B.64	同上
B.65	同上
B.68	同上
B.69	同上
B.70	同上

2.6　鍵情報の喪失・破損時の BCP 対策

①　鍵情報の喪失・破損に対する BCP 対策の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.71	FR10.11	CKMS 設計は、暗号鍵及びそのメタデータをバックアップ及びアーカイブするための手続きを明記しなければならない。	10.7 節
B.72	FR10.12	CKMS 設計は、保管又は伝送された破損した鍵情報（暗号鍵及びメタデータ）を復元又は置き換えるための手続きを明記しなければならない。	10.7 節

解説・考慮点

CKMS の設計にあたって、項目 B.71 は、BCP 対策として必要な鍵情報のバックアップを行うための手続きや要求事項を明確化することを、B.72 は BCP 対策として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、これらの項目で定めたことは B.52～B.61 の上位規定として機能し、B.52～B.61 の内容が B.71 及び B.72 の内容に矛盾してはならない。

鍵情報（暗号鍵やメタデータ）が喪失又は破損した場合で、バックアップもアーカイブもされていなかった場合、当該暗号鍵で保護されているデータの喪失につながる可能性がある。

とりわけ重大な災害は、多数の運用中の鍵情報の喪失又は破損を一気に引き起こす可能性が高い。この場合の BCP 対策として、鍵情報のバックアップやアーカイブは有効な手段であり、鍵情報の正当な復元を行うことで保護されているデータの喪失を防止することができる。その他のよくあるケースとしては、エンティティ（利用者）の誤操作や過失などによって、当人の鍵情報が破損したり紛失したりする場合があり、必要に応じて、バックアップからの復旧が求められることがある。

一方、バックアップやアーカイブからは要求したエンティティに対して復元した鍵情報を提供することになることから、万が一にも、そのエンティティが正しいエンティティでなかったり、正当な権限を持っていなかったりした場合は、復元要求を拒絶しなければならない。

また、鍵情報の喪失・破損の原因が紛失や攻撃など人為的な要因に起因する場合は、特に鍵情報の外部への流出などが否定できず、結果として当該暗号鍵で暗号化されていたデータの危殆化につながる可能性がある。この場合には、バックアップから単に当該鍵情報の正当な復元を行うだけでは不十分であり、当該暗号鍵の利用停止や失効処理、潜在的なリスク評価、新しい暗号鍵への置き換え及びデータの再暗号化といった、0 節の対応を含む一連の BCP 対策が必要となる。

そのため､鍵情報の喪失や破損時の BCP を実現するためにどのような対策が必要かを検討し、その結果、鍵情報のバックアップやアーカイブを行うこととした場合には、バックアップやアーカイブの方針をまず定める必要がある。この方針を具体化するものとして、項目 B.71 では、BCP 対策として必要な鍵情報のバックアップを行うための手続きや要求事項の明確化を､B.72 は BCP 対策として復旧を行うための手続きや要求事項の明確化を求めている。なお、これらの項目で定めたことは B.52～B.61 の上位規定として機能することから、B.52～B.61 の内容は B.71 及び B.72 の内容に沿って設定されなければならず､また矛盾していないことを確認することが重要である。

《トイモデルと記載例》

2.1 節と同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムであり、暗号鍵のライフサイクルは 2.2 節で定めたトイモデルにおいて、2.3 節で定めた運用条件を行った場合､BCP を実現するための鍵情報のバックアップとアーカイブのどちらも実施しないことになっている。また、鍵情報の喪失や破損が発生したときは新たな暗号鍵を再生成し公開鍵証明書

を発行し直す。

このようなトイモデルでは、鍵情報の喪失・破損時の BCP 対策が必要ないため、B.71 と B.72 は対象外となる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.71	B.02 の通り、鍵情報のバックアップとアーカイブのどちらも実施しないため、対象外。
B.72	同上

2.7　鍵情報の危殆化時の BCP 対策

①　暗号鍵の危殆化に対する BCP 対策の決定／②　メタデータの危殆化に対する BCP 対策の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.73	FR6.102	CKMS 設計は、システムによって使用されているそれぞれの鍵タイプの受け入れ可能な暗号鍵有効期間（cryptoperiod）又は利用制限（usage limit）の範囲を明記しなければならない。	6.8.1 節
B.74	FR6.103	それぞれの鍵に対し、CKMS 設計は、セキュリティがその鍵に依存する他の鍵タイプを明記しなければならず、また初期鍵の危殆化が発生した時にそれに依存する鍵がどのように置き換えられるかを明記しなければならない。	6.8.1 節
B.75	FR6.104	CKMS 設計は、鍵が危殆化したときに他の危殆化した鍵を特定できるための手段を明記しなければならない。例えば、鍵導出鍵が危殆化したとき、導出された鍵をどのように特定するのか？	6.8.1 節
B.76	FR6.105	導入されたそれぞれの鍵タイプに対して、CKMS 設計は、どのメタデータ要素が危殆化（機密性、完全性、又はソース）しやすいのかを明記しなければならない。	6.8.2 節
B.77	FR6.106	CKMS 設計は、鍵のそれぞれの危殆化しやすいメタデータ要素に危殆化（機密性、完全性、又はソース）が起こったときに、起こり得るセキュリティ結果を明記しなければならない。	6.8.2 節
B.78	FR6.107	CKMS 設計は、それぞれの危殆化しやすいメタデータ要素での危殆化からどのように回復できるかを明記しなければならない。	6.8.2 節

解説・考慮点

そもそも全ての潜在的なセキュリティ問題を CKMS が防止し鍵情報の危殆化が発生しないようにすることは現実的でないことを前提として、CKMS の設計においては鍵情報の危殆化を速

やかに検知できるようにすべきである。
鍵情報の危殆化が検知された場合、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

＜暗号鍵の危殆化＞
CKMS の設計にあたって、項目 B.73 は暗号鍵に対する暗号鍵有効期間の設定や利用範囲の制限について明確化することを、B.74 は危殆化した暗号鍵（の鍵タイプ）だけでなく連鎖的に影響を受ける可能性がある別の暗号鍵（の鍵タイプ）を含めてどのような 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。B.75 は、危殆化した暗号鍵から連鎖的に影響を受ける別の暗号鍵の特定方法の明確化を要求したものである。

＜メタデータの危殆化＞
CKMS の設計にあたって、項目 B.76 はメタデータの中でも危殆化が起こりやすい又は関連する暗号鍵の危殆化につながりやすいものがどれであるかを明確化することを、B.77 はメタデータに危殆化が起きた時にどのような影響が出るかを明確化することを要求したものである。B.78 はどのような 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。

暗号鍵の危殆化の影響は鍵タイプ及び鍵の用途に依存し、以下の結果をもたらし得る。

- 機密性の喪失
- 完全性の喪失
- 認証の喪失
- 否認防止の喪失
- これらの喪失の組み合わせ

一般に、暗号鍵が危殆化した場合には、当該暗号鍵の利用を停止し、新しい暗号鍵に置き換えるとともに、すでに暗号処理（暗号化や署名生成）が行われた情報に対しては個別にその正当性の判断を行うことになる。これらは、危殆化状態への遷移に相当し、B.13 や B.14 で決められた手段が取られる。

しかしながら、危殆化した暗号鍵の使われ方によっては、当該暗号鍵で保護されたデータに対してだけでなく、当該暗号鍵が保護する他の多くの暗号鍵についても危殆化を連鎖的に引き起こす可能性があることに留意されたい。例えば、システムマスター鍵など、上位の暗号鍵が危殆化すればシステム全体に影響が及ぶ可能性があり、鍵ラッピング鍵が危殆化すれば当該鍵で暗号化された鍵情報に影響が及ぶ。このような場合、危殆化した暗号鍵の失効・置き換えはもとより、

当該暗号鍵に依存して影響を受ける他の暗号鍵も可能な限り速やかに失効・置き換えを行うべきである。なお、暗号鍵の置き換えに伴い、データの再暗号化が必要となる場合もある。

また、暗号鍵の生成や更新に鍵導出手段を使っている場合、鍵導出手段に入力する元の暗号鍵が危殆化すると、それ以降に導出される暗号鍵も全て危殆化している。したがって、このような場合には、鍵導出手段を使うのではなく、改めて新しい暗号鍵を独自に生成し直す必要がある。

そこで、項目 B.74 と B.75 は、暗号鍵が危殆化した場合にどの程度の他の暗号鍵に影響を与える可能性があるかを、暗号鍵ごとに把握しておき、さらに BCP 対策として影響を受ける可能性がある暗号鍵の更新方法までを決めておくことを求めている。具体的には、B.74 は、危殆化した暗号鍵から連鎖的に影響を受ける可能性がある別の暗号鍵を含めてどのような BCP 対策を行い、暗号鍵を使う処理を再開するのかを明確化することを、B.75 は危殆化した暗号鍵から派生して生成される別の暗号鍵の特定方法の明確化を要求したものである。なお、鍵導出機能などを使わない場合には B.75 は対象外である。

暗号鍵の危殆化の影響を小さくするために、使用するそれぞれの暗号鍵に対して適切な暗号鍵有効期間の設定や利用範囲の制限をすることで、暗号鍵の危殆化のリスクを低減することも重要である。一般的には、対称鍵ラッピング鍵、鍵配送鍵、及び鍵合意鍵の暗号鍵有効期間を実用的な最短期間にしておくことがよい。この他、鍵導出鍵とマスタ鍵も定期的に変更したほうがよい。項目 B.73 は、暗号鍵の鍵タイプごとに有効期限や利用範囲を具体的に定めることを求めており、その範囲内で該当する暗号鍵を生成・利用するようにすることで、暗号鍵の危殆化の影響を小さくすることを目的としている。

メタデータの危殆化は、メタデータ要素及びその使われ方に依存して、暗号鍵の危殆化や当該暗号鍵によって保護されるデータの危殆化につながる可能性がある。項目が B.76 と B.77 は、メタデータのうち、どのデータが危殆化しやすいのか、さらにそのデータの危殆化が起きたら関連する暗号鍵にどのようなことが起きうるのかを具体的に把握することを求めている。B.78 は、危殆化状態からの復旧方法についての具体化を求めており、例えば、暗号鍵自体は危殆化していないことが確認でき、危殆化したメタデータの内容だけを更新すれば、当該暗号鍵の利用を再開できるようなケースで利用することを想定している。

③　役員・従業員によるセキュリティ危殆化に対する BCP 対策の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.79	FR6.117	CKMS 設計は、それぞれのサポートされる役割に提供される、あらゆる役員・従業員による危殆化の検知機能を明記しなければならない。	6.8.7 節
B.80	FR6.118	CKMS 設計は、それぞれのサポートされる役割に提供される、あらゆる役員・従業員による危殆化を最小化する機能を明記しなければならない。	6.8.7 節

B.81	FR6.119	CKMS 設計は、それぞれのサポートされる役割に提供される、CKMS 危殆化からの回復能力を明記しなければならない。	6.8.7 節

解説・考慮点

CKMS の設計にあたって、項目 B.79 及び B.80 は役員・従業員によるセキュリティ危殆化への事前対策としての要求事項を明確化することを求めたものである。B.81 は危殆化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。

CKMS のセキュアな運用に責任のある人間が、与えられた権限を悪用し、自らそのセキュリティを危殆化させる場合がある。そのような事態への対策としては、基本的に権限必要最小限ルールの徹底が重要であり、必要な人に必要な権限しか与えない、権限を悪用していないかを監査する、操作ログを隠蔽できないようにする、といった事前対策が重要となる。

また、役員・従業員によるセキュリティ危殆化が発生した場合には、あらかじめ決められた情報セキュリティポリシー及び CKMS 機能に基づいて、以下のような回復手続きで対応・復旧することが重要である。さらに、再発防止策としてのセキュリティポリシーや運用規程等の改訂もあり得る。

- システムの完全なシャットダウン
- 新しい暗号鍵によるバックアップ設備及びシステムの活性化
- 起こり得るセキュリティ障害についての現在及び潜在的ユーザへの通知
- 危殆化した暗号鍵へのフラグ付け・失効処理

項目 B.80 は上記に示したような事前対策に関する方針を実現するための手段を具体的に示すことを、B.81 は危殆化が検知された後にどのような BCP 対策を行い、暗号鍵を使う処理を再開するかを明確化することを要求したものである。

《トイモデルと記載例》

本節のトイモデルも、2.1 節のトイモデルと同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムであり、暗号鍵のライフサイクルは 2.2 節のトイモデルで定めたものとする。

また、システムの運用条件を以下のように設定する。【】内は記載例のどの項目に影響を与えているのかを示している。

- 署名プライベート鍵と署名公開鍵の有効期間は、公開鍵証明書の有効期間とする。【B.73】
- 署名プライベート鍵が危殆化すると署名公開鍵も同時に危殆化したとみなす。【B.74, B.75】
- 特に、危殆化しやすいと想定できるメタデータ要素はない。【B.76～B.78】
- 役割を分離して権限を必要な範囲にしている。【B.79】

- 個々のメール利用者が使用する PC において、使用している OS のログ保存機能により、署名プライベート鍵のアクセスログを管理する。アクセスログはシステム管理者しか確認できない。【B.79, B.80】
- 情報システム部の公開鍵証明書発行依頼の担当者の公開鍵証明書発行依頼ログと操作ログが PC 内に保存され、操作ログへのアクセスは情報システム部の管理者しかできない。【B.79, B.80】
- 鍵の危殆化が疑われるときは失効処理を行い、鍵を再生成し、証明書を再発行する。【B.81】

以上のトイモデルにおける記載例は、以下の「署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例」のようになる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.73	署名プライベート鍵と署名公開鍵の有効期間は、公開鍵証明書に記載された有効期間と一致する。システムの運用標準では 1 年プラス 10 日とする。
B.74	署名公開鍵は、署名プライベート鍵の危殆化に依存する。 署名プライベート鍵に危殆化の疑いが発生したときは、当該鍵ペアの両方の失効処理を行い、新たな鍵ペアを再生成し公開鍵証明書を発行し直すことにより、鍵の置き換えを行う。
B.75	署名プライベート鍵が危殆化すると署名公開鍵も同時に危殆化したとみなす。
B.76	公開鍵証明書に記載されているメタデータは、CA 署名により完全性が保護される。また、保存されるメタデータは B.44 に記載した ACS により完全性を保護されている。そのため、危殆化しやすいメタデータ要素は想定しない。
B.77	B.76 により、対象外。
B.78	同上
B.79	● 本システムでの役割として、以下の通りである。 ➢ メール利用者（メール送信者・メール受信者） ➢ 当該 PC ごとのシステム管理者（情報システム部内で PC ごとに担当者を割り当てる） ➢ 情報システム部の公開鍵証明書発行依頼の担当者 ➢ 情報システム部の公開鍵証明書発行依頼の管理者 ● メール利用者が使用する PC の署名プライベート鍵のアクセスログを保存する。システム管理者がアクセスログを毎月確認し、不正な署名生成が行われていないかどうかを確認する。 ● 公開鍵証明書発行依頼の担当者は、公開鍵証明書発行依頼ログを毎月確認し、不正な公開鍵証明書発行依頼が行われていないかどうかを確認する。 ● 公開鍵証明書発行依頼の管理者は、発行依頼の担当者の操作ログを毎月確認し、不正な公開鍵証明書発行依頼が行われていないかどうかを確認する。
B.80	● メール利用者が署名プライベート鍵を複製するなど、署名プライベート鍵に対するアクセスは全てアクセスログに記録される。

	● メール利用者は、署名プライベート鍵のアクセスログにアクセスできない。アクセスできるのは、システム管理者だけである。 ● PC は当該利用者が安全に管理する。 ● システム管理者は、メール利用者の許可又は別途規定に基づく手続きによる場合を除き、メール利用者の署名プライベート鍵にアクセスしてはならない。 ● 公開鍵証明書発行依頼の担当者が行う発行依頼は全て操作ログに記録される。 ● 公開鍵証明書発行依頼の担当者は、操作ログにアクセスできない。アクセスできるのは、公開鍵証明書発行依頼の管理者だけである。
B.81	● メール利用者が使用する PC のアクセスログにより署名プライベート鍵の危殆化を検知したシステム管理者は、CA に該当する鍵ペアの失効処理を依頼し、CA から得た証明書失効リストを関係するエンティティに配布する。 ● 公開鍵証明書発行依頼の管理者が、証明書発行依頼の処理に使用している PC の操作ログにより不正な証明書発行を検知した場合は、CA に該当する署名公開鍵の失効処理を依頼し、CA から得た証明書失効リストを関係するエンティティに配布する。 ● 失効処理した鍵ペアのメール利用者は新たな鍵ペアを再生成し、公開鍵証明書を新規発行する。

3 暗号アルゴリズムの選択

本章の目的・趣旨

本章は、設計指針（基本編）の 6 章に記載されている要求事項（各節での色付き枠内で示している内容）について解説したものである。

CKMS 設計では、要求される保護レベル（セキュリティ強度）を満たすように暗号アルゴリズムと鍵長を決定しなければならない。セキュリティ強度は、扱う情報の資産価値、求められる情報の機密性や完全性、保護する期間（保護終了年）を踏まえて決定すべきである。

決定したセキュリティ強度、暗号アルゴリズムと鍵長を明記することにより、どの程度安全に情報が保護されているかを確認することも可能になる。なお、本章で扱う暗号アルゴリズムとしては主に公開鍵暗号、デジタル署名、共通鍵暗号、ハッシュ関数を想定としているが、高機能暗号や秘密分散など、新しいタイプの暗号アルゴリズムを含めることもできる。

3.1 暗号アルゴリズムのセキュリティ

① 要求される保護レベル（セキュリティ強度）に対応した暗号アルゴリズムの決定

項目	FR 番号	Framework Requirements の内容	SP800-130
C.01	FR2.1	CKMS 設計は、システムによって使用される全ての暗号アルゴリズムとそれぞれのアルゴリズムでサポートされる全ての鍵長を明記しなければならない。	2.1 節
C.02	FR2.2	CKMS 設計は、鍵と鍵に結び付けられたメタデータを保護するために導入されているそれぞれの暗号技術について推定されるセキュリティ強度を明記しなければならない。	2.1 節

解説・考慮点

暗号アルゴリズムの選定方法について取り扱う。
CKMS がライフサイクル全体にわたって管理及び保護している暗号鍵を使用することで要求される保護レベル（セキュリティ強度）を満たすことができる暗号アルゴリズムを選定することが求められる。
項目 C.01 及び C.02 は、CKMS の設計にあたって、要求される保護レベル以上を実現していることを確認するために、採用している暗号アルゴリズム（鍵長を含む）及びセキュリティ強度の明確化を求めたものである。

本節で求めているのは、要求される保護レベル（セキュリティ強度）を決定し、それに対応した暗号アルゴリズム及び鍵長を選択し、明記することである。

具体的には、C.01 では、決定したセキュリティ強度に応じて使用可能な設定を行ったりして、当該システムで利用可能な全ての暗号アルゴリズムとサポートされる全ての鍵長を洗い出すことを求めている。また、C.02 では、C.01 に明記した暗号アルゴリズムで使われる暗号鍵とメタデ

ータを保護するために使用している暗号技術についても、どの程度のセキュリティ強度を有しているかを評価することを求めている。これは、C.01 で使う暗号アルゴリズムと鍵長が適切に選択されていたとしても、そこで利用する鍵情報が適切なセキュリティ強度で保護されていなければ、必要とされるセキュリティ強度が達成されないためである。

上記の要求を満たすためには、まず CKMS 設計では最初に必要なセキュリティ強度を決める必要がある。セキュリティ強度の決定では、扱う情報の資産価値、情報の機密性や完全性などのほか、該当システムの利用期間の終了年も重要な要因の一つとなる。なお、扱う情報によっては、当該情報の保護がシステムの終了年以降も必要な場合がある。例えば、攻撃者が暗号化された通信データを先に窃取しておいて解読が可能になった時期に復号を行う攻撃（Store (Harvest)-then-decrypt、Store now & decrypt later、Retrospective Decryption ともいう）を考慮する必要があるケースである。その場合、システムの終了年ではなく、当該情報の保護が必要な期間を基準にセキュリティ強度を決めるべきである。

具体的に必要なセキュリティ強度の決定にあたっては、「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[12]」又は「暗号鍵設定ガイダンス[13]」を参考にされたい。

また、扱う情報の資産価値（重要性）では、扱う情報の機密性・完全性・可用性に危殆化が発生したときに、どの程度の影響を与えるかを段階的に評価し、その影響度が大きいほど資産価値が高いと判断される。この評価はリスク分析の一環として行われることが多く、資産価値が高い情報と評価されると、その情報を扱うシステムではより強いセキュリティが求められることがある。このような場合には、通常のセキュリティ強度よりも高い強度を設定することが同時に求められる場合もあることに留意されたい。必要があれば、「中小企業の情報セキュリティ対策ガイドライン[14]」や「政府機関等の対策基準策定のためのガイドライン（令和 3 年度版）改定版[15]」なども参考にされたい。

なお、情報資産は時間の経過により、陳腐化する、逆に価値が上がることもある。そのような情報資産の重要性の変化があるかどうかも含めて検討し、適切なセキュリティ強度を設定すべきである。

「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」は、CRYPTREC 暗号リスト[16]に掲載されている暗号技術を利用する際に、適切なセキュリティ強度を実現するためのアルゴリズム及び鍵長の選択方法を規定したものである。したがって、利用する鍵長についてこの設定基準に合致しない鍵長を用いた場合には、電子政府推奨暗号リストの暗号技術を利用しているとは見なされないことに留意されたい。また、政府機関等のサイバーセキュリティ対策のための統一基準において適用対象となる電子政府システム（暗号化機能・電子署名機能の導入を行うものに限る。）の調達・開発・運用に関わる場合には、この設定基準に従う必要がある。

---

12 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、https://www.cryptrec.go.jp/list/cryptrec-ls-0003-2022.pdf

13 暗号鍵設定ガイダンス、https://www.cryptrec.go.jp/report/cryptrec-gl-3003-1.0.pdf

14 中小企業の情報セキュリティ対策ガイドライン第 3 版、https://www.ipa.go.jp/files/000055520.pdf

15 政府機関等の対策基準策定のためのガイドライン（令和 3 年度版）改定版、https://www.nisc.go.jp/pdf/policy/general/guider3_2.pdf

16 電子政府における調達のために参照すべき暗号リスト（CRYPTREC 暗号リスト）https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2012r7.pdf

表 3-1 は、本設定基準において、システムの想定運用終了・廃棄年又は利用期間の終了年を基準に必要なセキュリティ強度要件を示したものである。例えば、128 ビットセキュリティは 2022～2040 年が利用可能であり、2041～2050 年が移行完遂期間になっているので、2050 年までに運用を停止するシステムであれば 128 ビットセキュリティでもよいが、2051 年以降も何らかの形で運用するシステムでは 2050 年までに 192 ビットセキュリティ以上のセキュリティ強度に移行することが必要となる。

表 3-1　セキュリティ強度要件の基本設定方針

<table>
<tr><th colspan="2">想定運用終了・廃棄年／利用期間</th><th>2022～2030</th><th>2031～2040</th><th>2041～2050</th><th>2051～2060</th><th>2061～2070</th></tr>
<tr><td rowspan="2">112 ビット
セキュリティ</td><td>新規生成
((a)参照)</td><td rowspan="2">移行完遂
期間
((c)参照)</td><td>利用不可</td><td rowspan="2">利用不可</td><td rowspan="2">利用不可</td><td rowspan="2">利用不可</td></tr>
<tr><td>処理
((b)参照)</td><td>許容</td></tr>
<tr><td rowspan="2">128 ビット
セキュリティ</td><td>新規生成
((a)参照)</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">移行完遂
期間
((c)参照)</td><td>利用不可</td><td rowspan="2">利用不可</td></tr>
<tr><td>処理
((b)参照)</td><td>許容</td></tr>
<tr><td rowspan="2">192 ビット
セキュリティ</td><td>新規生成
((a)参照)</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td></tr>
<tr><td>処理
((b)参照)</td></tr>
<tr><td rowspan="2">256 ビット
セキュリティ</td><td>新規生成
((a)参照)</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td></tr>
<tr><td>処理
((b)参照)</td></tr>
</table>

(a) **新規に暗号保護を適用**する（例えば、暗号化や署名生成を実行する）際は、原則として、2040 年までは 128 ビット以上のセキュリティ強度のものを**選択すべきである**。2041 年以降は 192 ビット以上のセキュリティ強度のものを**選択すべきである**。

(b) **保護済みのデータに対して処理を実行**する（例えば、復号や署名検証を実行する）際は、2040 年までは 128 ビット以上、2041 年以降は 192 ビット以上のセキュリティ強度のものを**選択すべきである**。ただし、保護済みのデータに対する正当性を担保又は確認するための何らかの技術的又は運用的な対策やルール等（暗号技術によるものとは限らない）を併用している場合に、2031 年以降も 2040 年までの必要な範囲内で 112 ビットセキュリティ強度のものを選択することを許容する。同様に、2051 年以降も 2060 年までの必要な範囲内で 128 ビットセキュリティ強度のものを選択することを許容する。

(c) 移行完遂期間内に、よりセキュリティ強度の高い暗号技術又は鍵長への移行を完遂させることを前提として、利用する暗号処理が短期間で完結する場合（例：エンティティ認証）、又は既存の電子政府システムの継続利用やそれらとの互換性・相互接続性維持などの必要がある

場合には、2030 年までは 112 ビットセキュリティ強度のものを、2050 年までは 128 ビットセキュリティ強度のものを選択することを許容する。

「暗号鍵設定ガイダンス」は、安全な暗号技術の導入の観点から、暗号技術を利用する際の鍵長の選択方法に関する一般的な考え方を解説したものであり、必要な鍵長を判断する上での目安となるものである。表 3-2 は、本ガイダンスにおいて、1982 年当時に DES が有していたのと同程度のセキュリティ強度を実現するために必要と推定されるビットセキュリティを示したものであり、表 3-2 でのビットセキュリティを下限のセキュリティ強度として、一定のセキュリティマージン（数十ビット）を追加したそれ以上のセキュリティ強度で設定することが望ましいと記載されている。例えば、2040 年に該当システムの利用を終了するのであれば最低で 104 ビット以上、できれば 128 ビット以上のセキュリティ強度を設定するのがよい。

表 3-2　1982 年の DES と同等のセキュリティを提供すると推定される（＝その後 10～15 年程度安全と期待される）ビットセキュリティ

年	1982	2030	2040	2050	2060	2070
ANSSI（2014）	56	81 ～ 96	86 ～ 104	91 ～ 112	96 ～ 120	101 ～ 128
Lenstra（2001）	56	93	101	109	－	－
Lenstra（2004）	56	88	95	102	－	－

必要なセキュリティ強度を決定したら、暗号アルゴリズムと鍵長を決定する。その際、使用する全ての暗号アルゴリズムと鍵長が必要とするセキュリティ強度を上回る強度となるように選択しなければならない。公開鍵暗号、デジタル署名、共通鍵暗号、ハッシュ関数については、CRYPTREC 暗号リストの電子政府推奨暗号リスト（又は推奨候補暗号リスト）に掲載されている暗号アルゴリズムから選択することを推奨する。また、セキュリティ強度と鍵長の関係は、「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」又は「暗号鍵設定ガイダンス」の 2.2 節「暗号技術の推定セキュリティ強度表現－ビットセキュリティ」を参照する。

《トイモデルと記載例》

本節のトイモデルは、Web ブラウザをクライアントとするクライアントーサーバシステムである。暗号プロトコルとして TLS 通信を使用する。

そこで、このトイモデルでは、セキュリティ強度は「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」に従い、また TLS サーバでの暗号スイートの設定では「TLS 暗号設定ガイドライン[17]」の「高セキュリティ型での暗号スイート推奨設定（TLS1.3 限定)」に従って設定している。

まず、該当システムは 2023 年から 8 年間使用し 2031 年末破棄する予定である。この場合、表 3-1 により必要なセキュリティ強度は 128 ビットとなり、TLS1.3 の中で利用する暗号アルゴリ

[17] TLS 暗号設定ガイドライン、https://www.cryptrec.go.jp/report/cryptrec-gl-3001-3.0.1.pdf

ズムでの鍵長は 128 ビットセキュリティを満たす鍵長を利用することになる。

TLS1.3 を利用する場合における高セキュリティ型の暗号スイートは、

TLS_AES_256_GCM_SHA384、TLS_CHACHA20_POLY1305_SHA256、
TLS_AES_128_GCM_SHA256、TLS_AES_128_CCM_SHA256、
TLS_AES_128_CCM_8_SHA256

であり、個々の暗号スイートの最後の SHA384/ SHA256 は鍵生成 HKDF で使用するハッシュ関数を示している。

なお、TLS 暗号設定ガイドラインにおける高セキュリティ型での推奨設定では、暗号アルゴリズム自体の安全性に問題がないものだけが選定されているので、暗号アルゴリズムそのものの選択に関しては気にする必要はない。

また、このサーバで使われるサーバ証明書は、セキュリティ強度が 128 ビット以上必要なので、ここでは署名方式として ECDSA P-256、証明書に使用するハッシュ関数として SHA-256 を使用する。

このトイモデルを対象とした場合、C.01 については「TLS 暗号設定ガイドライン」の高セキュリティ型での推奨設定に従った内容となる。C.02 については、鍵とメタデータを保護するための暗号技術としては、認証局が署名したサーバ証明書によって「サーバ公開鍵」と「有効期間」とが結び付けられているので、推定セキュリティ強度は認証局の署名で使われている暗号アルゴリズムと鍵長から導かれるセキュリティ強度である。

以上のトイモデルにおける記載例は、以下の「クライアントーサーバシステムにおける記載例」のようになる。

クライアントーサーバシステムにおける記載例

C.01	署名 ECDSA P-256 - 鍵長 256 ビット SHA-256 鍵合意 ECDHE P-256 - 鍵長 256 ビット X25519 - 鍵長 255 ビット データ暗号化／復号 AES-128-GCM - 鍵長 128 ビット AES-256-GCM - 鍵長 256 ビット AES-128-CCM - 鍵長 128 ビット AES-128-CCM-8 - 鍵長 128 ビット CHACHA20-POLY1305 - 鍵長 256 ビット

	鍵生成（HKDF） HMAC-SHA-256 HMAC-SHA-384
C.02	サーバ公開鍵とメタデータ（有効期間）の結び付きの保護は、認証局の署名により行っている。そこで利用している暗号技術は、ECDSA P-256 と SHA-256 であるので、セキュリティ強度は 128 ビットセキュリティである

4 暗号アルゴリズム運用に必要な鍵情報の管理

本章の目的・趣旨

本章は、設計指針（基本編）の 7 章に記載されている要求事項（各節での色付き枠内で示している内容）について解説したものである。

「鍵情報の管理」の主要な効果の 1 つは、該当システムで使用している暗号鍵について、鍵タイプとメタデータに応じて分類した上で保護方法などを明記し管理することにより、その暗号鍵が安全に管理されていることを明確にすることである。

暗号鍵の管理方法は、扱う情報の資産価値や対策コストを考慮して CKMS 設計者が決定することとなるが、利用用途や目的等に応じた鍵タイプごとに、その特徴に合わせた相応しい鍵管理を実施することが必要になる。また、鍵タイプによってはメタデータとの組み合わせによりその暗号鍵の完全性を確保する場合もあるため、とりわけそのような鍵タイプの暗号鍵ではメタデータの管理も安全な鍵情報の管理を実施するために重要となる。

このように分類して鍵情報を管理することで、CKMS 設計者が明示的に管理すべき暗号鍵を漏れなく洗い出し、それらの暗号鍵を安全に管理していることを明確にすることができる。また、ある暗号鍵の危殆化が疑われるときに、その暗号鍵の管理状況の確認や危殆化による影響リスクの判断等に利用することができ、さらに、危殆化が発生した場合に敏速な対応も可能になる。

本章では、暗号鍵の安全な管理を行うために必要な、鍵タイプ、鍵のメタデータの分類方法、またそれらの保護方針や記載方法を解説する。

4.1 鍵情報の種類

SP800-130 に記載されている鍵タイプとメタデータを以下のように解説している。CKMS 設計者は、明示的に管理すべき暗号鍵すべてについて、暗号鍵の利用用途や目的に応じて、それぞれの暗号鍵がどの鍵タイプに属するか、管理対象となる鍵情報が何であるかを分類することが必要になる。

暗号鍵は、以下の通り、特性と用途（＋オプション）に応じて分類され、これらの組み合わせで「鍵タイプ」が定義される。

特性	公開（Public）	一般に公開できる情報
	プライベート（Private）	一人のユーザのみが秘密に保持する情報
	対称（Symmetric）	送信者と受信者が共通して秘密に保持する情報

オプション	静的（Static）	長期的に固定した情報
	一時的（Ephemeral）	1つのセッションやトランザクションでのみ使われる情報

用途	データの暗号化／復号（Encryption/Decryption）
	鍵ラッピング（Key Wrapping）
	鍵配送（Key Transport）

	鍵合意（Key Agreement）
	署名（Signature）
	認証（Authentication）
	認可（Authorization）
	乱数生成（Random Number Generator（RNG））
	マスタ鍵（Master Key）

SP800-130 で分類する鍵タイプは以下の通りである。

1）署名プライベート鍵（Private Signature Key）
2）署名公開鍵（Public Signature Key）
3）認証対称鍵（Symmetric Authentication Key）
4）認証プライベート鍵（Private Authentication Key）
5）認証公開鍵（Public Authentication Key）
6）データ暗号化／復号対称鍵（Symmetric Data Encryption/Decryption Key）
7）鍵ラッピング対称鍵（Symmetric Key Wrapping Key）
8）乱数生成対称鍵（Symmetric RNG Key）
9）乱数生成プライベート鍵（Private RNG Key）
10）乱数生成公開鍵（Public RNG Key）
11）マスタ対称鍵（Symmetric Master Key）
12）鍵配送プライベート鍵（Private Key Transport Key）
13）鍵配送公開鍵（Public Key Transport Key）
14）鍵合意対称鍵（Symmetric Key Agreement Key）
15）鍵合意静的プライベート鍵（Private Static Key Agreement Key）
16）鍵合意静的公開鍵（Public Static Key Agreement Key）
17）鍵合意一時的プライベート鍵（Private Ephemeral Key Agreement Key）
18）鍵合意一時的公開鍵（Public Ephemeral Key Agreement Key）
19）認可対称鍵（Symmetric Authorization Key）
20）認可プライベート鍵（Private Authorization Key）
21）認可公開鍵（Public Authorization Key）

メタデータは、CKMS によって明示的に記録され管理されている特定の暗号鍵に関連付けられている情報として定義されるものであり、特性、制約、受け入れられるユーザ及び適用可能なパラメタを指定する。メタデータの各ユニットはメタデータ要素と呼ばれる。SP800-130 で取り上げる典型的なメタデータ要素は以下の通りである。

a）鍵ラベル（Key Label）
b）鍵識別子（Key Identifier）
c）所有者識別子（Owner Identifier）
d）鍵ライフサイクル状態（Key Lifecycle State）
e）鍵フォーマット指定子（Key Format Specifier）

f) 鍵生成に使用した製品（Product used to create the Key）
g) 鍵を使用する暗号アルゴリズム（Cryptographic Algorithm using the Key）
h) スキーム又は暗号利用モード（Scheme or Modes of Operation）
i) 鍵パラメタ（Parameters for the Key）
j) 鍵長（Length of the Key）
k) 鍵／アルゴリズム組のセキュリティ強度（Security Strength of the Key/Algorithm Pair）
l) 鍵タイプ（Key Type）
m) 鍵に対する適切なアプリケーション（Appropriate Applications for the Key）
n) 鍵セキュリティポリシー識別子（Key Security Policy Identifier）
o) 鍵アクセスコントロールリスト（Key Access Control List（ACL））
p) 鍵使用カウント（Key Usage Count）
q) 親鍵（Parent Key）
r) 鍵機微性（Key Sensitivity）
s) 鍵保護（Key Protections）
t) メタデータ保護（Metadata Protections）
u) 信頼関係保護（Trusted Association Protections）
v) 日時（Date Times）
w) 失効理由（Revocation Reason）

解説・考慮点

暗号鍵は、「特性」に応じて、対称鍵、公開鍵、プライベート鍵に大きく分類できる。

対称鍵は、暗号化に使用する暗号鍵と復号に使用する暗号鍵が同一であり、情報を所有又は共有するエンティティ（情報所有者や、情報送信者と情報受信者）のみが秘密裏に利用する。

公開鍵とプライベート鍵はペアで使用する。暗号の場合、公開鍵は暗号化に使用し、プライベート鍵は復号に使用する。署名と検証の場合、公開鍵は検証に使用し、プライベート鍵は署名に使用する。公開鍵は、不特定多数のエンティティに知られても良いが、プライベート鍵は復号や署名する単独のエンティティ（一人のユーザ）以外には知られることが無いようにする。

時間的な「オプション」として、暗号鍵を長期に渡り利用する場合は静的な暗号鍵と呼び、短期間しか使わない場合は一時的な暗号鍵と呼ぶ。例えば、接続ごとに(EC)DH[18]で鍵交換する場合、一回の接続（1 つのセッションやトランザクション）でしか使用しない鍵であるため、一時的な鍵である。

「用途」は、その暗号鍵の利用用途や目的を表している。この用途と、上記の特性とオプションに応じて、「暗号鍵管理システム設計指針（基本編）」に記載されている 21 種類の鍵タイプに分類でき、それぞれの鍵タイプの役割は、以下の通りである。

[18] 一時的(EC)DH は、一時的（Ephemeral）であることを示すため(EC)DHE と記述することもある。

1) 署名プライベート鍵：デジタル署名を生成するためのプライベート鍵である。署名は公開鍵アルゴリズムが使用され、この鍵と鍵ペアとなる 2)の署名検証公開鍵が存在する。適切に扱うことができれば、署名プライベート鍵は、メッセージやドキュメント、保存されたデータのソース認証と完全性認証を提供するほか、それらの否認防止をサポートするためにも、使用することができる。
2) 署名公開鍵：デジタル署名を検証するための鍵である。公開鍵アルゴリズムで使用される鍵ペアの公開鍵であり、メッセージ、ドキュメント又は保存されたデータのソース認証と完全性認証を提供するほか、それらの否認防止をサポートすることを目的としたデジタル署名を検証するために使用される。
3) 認証対称鍵：対称鍵アルゴリズムと共に使用され、通信セッション、メッセージ、文書又は保存されたデータの ID 認証と完全性認証を提供する。対称鍵アルゴリズムの認証暗号利用モードでは、一つの鍵が認証と暗号化の両方に使用されることになる。（SP 800-175B を参照）。
4) 認証プライベート鍵：エンティティの身元の保証（つまり、ID 認証）を提供するための鍵である。この認証は公開鍵アルゴリズムが使用され、この鍵と鍵ペアとなる 5)の認証公開鍵が存在する。認証された通信セッション又は何らかのアクションを実行するための認可を確立するときに使用される。
5) 認証公開鍵：エンティティの身元の保証（つまり、ID 認証）を提供するための公開鍵アルゴリズムで使用される鍵ペアの公開鍵である。認証された通信セッション又は何らかのアクションを実行するための認可を確立するときに使用される。
6) データ暗号化／復号対称鍵：対称鍵アルゴリズムを用いて、データの機密性保護（平文データの暗号化）をするための鍵である。同じ鍵が、機密性保護を解除（暗号文データの復号）するためにも使用される。対称鍵アルゴリズムの認証付き秘匿モードでは、一つの鍵がソース認証と暗号化の両方に使用される。
7) 鍵ラッピング対称鍵：対称鍵アルゴリズムを用いて、他の鍵を暗号化するための鍵である。鍵暗号化鍵と呼ばれることもある。鍵の暗号化に使用された鍵ラッピング鍵は、暗号化処理を元に戻す（つまり、暗号化された鍵を復号する）ためにも使用される。鍵を使用するアルゴリズムによっては、完全性保護を提供するために鍵を使用することもできる。
8) 乱数生成対称鍵：対称暗号方式を使用して乱数を生成するための鍵である。
9) 乱数生成プライベート鍵：公開鍵アルゴリズムを使って乱数を生成するためのプライベート鍵である。この鍵と鍵ペアとなる 10)の乱数生成公開鍵が存在する。
10) 乱数生成公開鍵：乱数を生成するための鍵である。9)の乱数生成プライベート鍵のペアとなる乱数生成公開鍵である。
11) マスタ対称鍵：対称暗号化方式を使用して他の対称鍵（データ暗号化鍵や鍵ラッピング鍵など）を導出するための鍵である。マスタ鍵は、鍵導出鍵とも呼ばれる。
12) 鍵配送プライベート鍵：公開鍵暗号アルゴリズムを使用してペアとなる公開鍵で暗号化

された鍵の復号に使用される鍵である。この鍵と鍵ペアとなる 13)の鍵配送公開鍵が存在する。鍵配送鍵は、通常、対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び場合によっては他の鍵材料（例えば、初期ベクトル）を確立するために使用される。

13) 鍵配送公開鍵：公開鍵アルゴリズムを使用して鍵を暗号化するために使用される鍵である。12)の鍵配送プライベート鍵のペアとなる公開鍵である。これらの鍵ペアは、対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び場合によっては他の鍵材料（例えば、初期ベクトル）を確立するために使用される。確立された鍵の暗号化形態は、後で鍵配送プライベート鍵を使用して復号するために保存できる。

14) 鍵合意対称鍵：対称鍵合意アルゴリズムを使用して、対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び他の鍵材料（例えば、初期ベクトル）を確立するための鍵である。

15) 鍵合意静的プライベート鍵：公開鍵アルゴリズムを使用して対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び他の鍵材料（例えば、初期ベクトル）を確立するための長期的なプライベート鍵である。この鍵と鍵ペアとなる 16)の鍵合意静的公開鍵が存在する。

16) 鍵合意静的公開鍵：公開鍵アルゴリズムを使用して対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び他の鍵材料（例えば、初期ベクトル）を確立するための長期的な公開鍵である。15)の静的鍵合意プライベート鍵のペアとなる公開鍵である。

17) 鍵合意一時的プライベート鍵：公開鍵アルゴリズムを使用して対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び他の鍵材料（例えば、初期ベクトル）を確立するための短期的なプライベート鍵である。対称鍵や鍵材料を確立するために一度だけ使用される。この鍵と鍵ペアとなる 18)の鍵合意一時的公開鍵が存在する。

18) 鍵合意一時的公開鍵：公開鍵アルゴリズムを使用して対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び他の鍵材料（例えば、初期ベクトル）を確立するための短期的な公開鍵である。対称鍵や鍵材料を確立するために一度だけ使用される。17)の鍵合意一時的プライベート鍵のペアとなる公開鍵である。

19) 認可対称鍵：対称暗号方式を使用してエンティティに権限を付与するための鍵である。認可鍵は、認可されたエンティティへのアクセス権限の監視と付与を担当する責任を負うエンティティ、及びリソースへのアクセスを求めるエンティティに知らされる。

20) 認可プライベート鍵：権限に対する所有者の権利を証明するために使用（例えば、デジタル署名を使用）される非対称鍵（公開鍵）の鍵ペアのプライベート鍵である。この鍵と鍵ペアとなる 21)の認可公開鍵が存在する。

21) 認可公開鍵：関連する認可プライベート鍵を知っているエンティティに対する権限を検証するために使用される非対称鍵（プライベート鍵）の鍵ペアの公開鍵である。20)の認可プライベート鍵のペアとなる公開鍵である。

メタデータは、暗号鍵を適切に管理するために、その暗号鍵に関連付けられている情報である。例えば、鍵の保護方法や有効期間などである。「暗号鍵管理システム設計指針（基本編）」では、メタデータの典型的な要素として a)～w)の 23 種類が記載されている。詳細は以下の通りである。

なお、全ての暗号鍵に対してメタデータが必要となるわけではなく、またメタデータが必要な場合であっても全ての適用可能なメタデータ要素と関連付ける必要は必ずしもないことに留意されたい。

a) 鍵ラベル（Key Label）：人間が読解可能なテキスト文字列で書かれた鍵の記述子のことである。例えば、“root CA Private Key 2022”や“Cryptrec Secret Key 2023”などである。
b) 鍵識別子（Key Identifier）：多数の鍵から特定の鍵を識別するための識別子のことである。CKMS は一般的にユニークな識別子を割り当てる。
c) 所有者識別子（Owner Identifier）：鍵を所有するエンティティの識別子のことである。
d) 鍵ライフサイクル状態（Key Lifecycle State）：「4.2 暗号鍵のライフサイクル」に記載した鍵の状態のことである。
e) 鍵フォーマット指定子（Key Format Specifier）：鍵のフォーマットの指定子のことである。例えば、RSA 公開鍵は法（modulus）と公開指数（public exponent）があり、その 2 つの値の格納順及び値の表記方法を指定する。Internet Engineering Task Force（IETF）は、(EC)DH、RSA、ECDSA などの鍵を格納するための方法を RFC 5208、RFC 5480、RFC 5958 で定義している。
f) 鍵生成に使用した製品（Product used to create the Key）：鍵生成に使用した製品情報のことである。
g) 鍵を使用する暗号アルゴリズム（Cryptographic Algorithm using the Key）：その鍵が使用する暗号アルゴリズムを指定する。例えば、ECDSA、RSA、AES、Camellia、HMAC-SHA256 などである。
h) スキーム又は暗号利用モード（Scheme or Modes of Operation）：その鍵が使用する暗号アルゴリズムを実行するための適用可能なスキーム又は暗号利用モードのことである。例えば、非対称アルゴリズムでは、有限体、binary field（標数 2 の体）又は楕円曲線（EC）上の離散対数問題の演算を指定する。対称アルゴリズムでは、ブロック暗号で使用される暗号利用モード（CBC、OFB、CCM、GCM 等）を指定する。詳細に関しては、［SP 800-38A］～［SP 800-38D］を参照されたい。
i) 鍵パラメタ（Parameters for the Key）：鍵のパラメタが存在する場合に指定する。例えば、ECDSA 鍵には（素数（p）、楕円曲線の係数（a, b）、生成元（G）、G の位数（n）、n のコファクタ（h））のドメインパラメタがある。
j) 鍵長（Length of the Key）：鍵の長さをビット（又はバイト）で指定する。例としては、RSA の法の 2048 ビットや楕円曲線暗号の鍵の 256 ビットである。
k) 鍵／アルゴリズム組のセキュリティ強度（Security Strength of the Key/Algorithm Pair）：「5.1.1 暗号アルゴリズムのセキュリティ強度」に記載したセキュリティ強度である。
l) 鍵タイプ（Key Type）：本節で説明している鍵タイプのことである。

m) 鍵に対する適切なアプリケーション（Appropriate Applications for the Key）：鍵を使用してよいアプリケーションのことである。例えば、TLS、SSH、デジタル署名である。

n) 鍵セキュリティポリシー識別子（Key Security Policy Identifier）：暗号鍵又は鍵タイプに適用できるセキュリティポリシーを特定する識別子のことである。このセキュリティポリシーは、生成から破壊までの鍵ライフサイクル全体にわたって、暗号鍵又は鍵タイプを保護するために使用するセキュリティコントロール一式である。

o) 鍵アクセスコントロールリスト（Key Access Control List（ACL））：暗号鍵及びメタデータの管理機能で制限された通りに、暗号鍵へのアクセスが可能なエンティティを特定するリストである（「7.1 鍵情報へのアクセスコントロール」を参照）。例えば、Microsoft Windows の ACL、Linux の ACL 、HSM を使った方法などがある。

p) 鍵使用カウント（Key Usage Count）：鍵が使用された回数のことである。

q) 親鍵（Parent Key）：メタデータに関連付けられた暗号鍵の導出元となった暗号鍵のことである。例えば、TLS 1.3 の通信に使われる暗号鍵である各種トラフィックシークレットは、マスタシークレットやハンドシェイクシークレットなどを親鍵として導出される。

r) 鍵機微性（Key Sensitivity）：暗号鍵の機微度又は重要度のことである。これは、リスクレベル（例えば、低、中、高）又は機密区分レベル（例えば、Confidential、Secret、Top Secret）に関係する。

s) 鍵保護（Key Protections）：暗号鍵に対する完全性、機密性及びソース認証（source authentication）の保護メカニズムのことである。公開鍵証明書は、CA のデジタル署名が完全性保護とソース認証の両方を提供する鍵保護の例である（［X.509］参照）。対称鍵及びそのハッシュ値を共に暗号化したものは機密性と完全性の保護の例である。なお、暗号鍵及びそのメタデータを外部エンティティから受信した場合は、それらが保護されているかどうかを事前に検証する必要があり、一般に、1 つの暗号機能（例：HMAC 又はデジタル署名）が完全性保護とソース認証の両方を提供するために使用される。

この保護メカニズムでは、いくつかの下位要素を持つことがある：

i. 完全性保護に使用されるメカニズム（例：ハッシュ値、MAC、又はデジタル署名）
ii. 機密性保護に使用されるメカニズム（例：鍵ラッピング、又は鍵配送）
iii. ソース認証に使用されるメカニズム（例：MAC、又はデジタル署名）
iv. 特定の非暗号学的な信頼プロセスによって実施される保護の表示

t) メタデータ保護（Metadata Protections）：関連付けられたメタデータの完全性、機密性及びソース認証を保護するために使用されるメカニズムのことである。一般には、鍵保護と同じメカニズムで保護するが、別のメカニズムで保護することもある。鍵保護と同様の下位要素を持つことがある。

u) 信頼関係保護（Trusted Association Protections）：暗号鍵とメタデータが正しく関連付けられていることを保証するために、その暗号鍵とメタデータとの信頼関係を保護するためのメカニズムのことである。上記の項目 s）で挙げられている保護によって、暗号鍵とメタデータがひとつの集約した項目として保護されている場合、信頼関係保護は暗黙的に提供されている。それ以外で信頼関係保護が必要な場合は、以下の項目が提供されるべきである：

i. 完全性保護に使用されるメカニズム（例：ハッシュ値、MAC、デジタル署名、又は信頼プロセス）
ii. ソース認証に使用されるメカニズム（例：暗号学的メカニズム又は非暗号学的な信頼プロセス）

v) 日時（Date Times）：暗号鍵の状態遷移のための日時のことである。例えば、鍵の使用開始日（活性化状態に遷移）や終了日（活性化状態から非活性化状態に遷移）、有効期限等に関するものがある。

w) 失効理由（Revocation Reason）：暗号鍵が失効した場合の失効理由のことである。例えば、暗号鍵の漏洩の疑い、暗号モジュール危殆化の疑い、鍵所有者の組織離脱、鍵の誤使用等がある。

4.2 鍵情報の選択

① CKMS が取り扱う全ての鍵タイプの利用用途及び生成手段、メタデータ、信頼関係、保護方針などの決定

項目	FR 番号	Framework Requirements の内容	SP800-130
D.01	FR6.1	CKMS 設計は、使用されているそれぞれの鍵タイプを明記及び定義しなければならない。	6.1 節
D.02	FR6.2	システムで使用されているそれぞれの鍵タイプに対して、CKMS 設計は、信頼関係のために選択される全てのメタデータ要素、メタデータ要素が作成され鍵との関連付けが満たされている状況、及び関連付けの手段（すなわち、暗号メカニズム又は信頼プロセス）を明記しなければならない。	6.2.1 節
D.03	FR6.13	それぞれの鍵タイプに対して、CKMS 設計は、暗号鍵及びメタデータ要素に関する以下の情報を明記しなければならない： a) 鍵タイプ b) 暗号鍵有効期間（cryptoperiod）（静的鍵（static key）に対して） c) 生成手段 i. 使用した乱数生成器（RNG） ii. 鍵生成の仕様（例えば、署名鍵については［FIPS 186］、Diffie-Hellman 鍵確立鍵（key establishment key）については［SP800-56A］） d) それぞれのメタデータ要素に対して、以下を含める i. メタデータのソース ii. メタデータの検証方法 e) 鍵確立（key establishment）の手段	6.2.2 節

		i. 鍵配送スキーム（使用されている場合） ii. 鍵合意スキーム（使用されている場合） iii. プロトコル名（名称があるプロトコルが使用されている場合） f) 暴露に対する保護（例えば、鍵の機密性、物理セキュリティ） g) 改ざんに対する保護（例えば、MAC 又はデジタル署名） h) 鍵を使用し得るアプリケーション（例えば、TLS、EFS、S/MIME、IPSec、PKINIT、SSH、等） i) 鍵の使用が許可されないアプリケーション j) 鍵保証（key assurances） i. 対称鍵保証（Symmetric key assurances）（例えば、フォーマットチェック） • 誰が保証を得るか • 保証が得られる状況 • どのように保証を得るか ii. 非対称鍵保証（Asymmetric key assurances）（例えば、所有と有効性の保証） • 誰が保証を得るか • 保証が得られる状況 • どのように保証を得るか iii. ドメインパラメタ有効性チェック • 誰が有効性チェックを実行するか • チェックが実行される状況 • どのようにドメインパラメタの有効性の保証を得るか	
D.04	FR6.14	CKMS 設計は、CKMS によって生成、保管、伝送、処理、及びその他管理される全ての鍵タイプ及びメタデータについて、全てのシンタックス、セマンティクス、及びフォーマットを明記しなければならない。	6.2.2 節

解説・考慮点

CKMS の設計にあたって、項目 D.01～D.04 は、鍵情報の選択にあたっての要求事項を明確化することを求めたものである。D.01 は利用する鍵タイプの一覧、D.02 はメタデータに関しての要求事項、D.03 は鍵情報の利用条件や取り扱い方法、D.04 は書式方法を対象にしている。

本節の要求事項で、「CKMS が取り扱う全ての鍵タイプの利用用途及び生成手段、メタデータ、信頼関係、保護方針などを決めなければならない」と求めているが、ここでの「全ての鍵」の意図は、CKMS 設計で「明示的に選択や管理する必要がある全ての鍵（タイプ）」に対してのことを指している。つまり、CKMS 設計で明示的に選択や管理しておらず、プロトコルや製品仕様により内部処理として自動的に生成・使用される暗号鍵は基本的には含まない。但し、このことは、

利用する製品やアプリケーション、システムが自動的に生成・使用される暗号鍵を「ブラックボックスとして使っている」という認識を持つことが重要である。したがって、このような処理を行う部分（多くは暗号モジュール）については信頼できる製品を使うことが望ましい。例えば、暗号モジュール認証を取得した製品などである。もし信頼性に確信が持てない暗号モジュールを使用している場合などは、可能であれば、内部の処理を調査し、暗号鍵の信頼性を確認することが望ましい。

具体的に、D.01 では、CKMS 設計者が明示的に選択して管理する全ての鍵タイプを洗い出し、どのような鍵タイプを利用しているのかを明確化することを求めている。その際、使用するソフトウェアの仕様書、設計書や規格などを参考にすべきである。市販品（COTS）デバイスやオープンソースソフトウェア（OSS）を使用している場合は、公開されている仕様書等を参照して記載すべき鍵タイプなどの情報を得ることができることもある。なお、システム内に他者が実装したサブモジュールや既存システム内に前任者等が設計したサブモジュールが含まれている場合には、CKMS 設計者が明示的に選択して管理する鍵情報として、それらのモジュールで使われる鍵情報も対象に含まれる場合がある。このようなケースでは、必要に応じて、例えば SBOM（Software Bill Of Materials : ソフトウェア部品表）などを利用して、実態調査を行うことも想定されたい。

暗号鍵の各種管理機能を実行するためには暗号鍵とメタデータが正しく関連付けられている必要がある。そのため、D.02 と D.03 では、D.01 で洗い出した個々の鍵タイプに対して、対象となる暗号鍵の完全性を確保するために必要となるメタデータの洗い出しと、その暗号鍵とメタデータが正しく関連付けられていることを保証するための方法を明確にすることが求められている。

D.02、D.03 における信頼関係や鍵の保護方法や利用手段の決定においては、システムで使用する OS やハードウェア環境により選択可能な方式が異なることに留意されたい。例えば暗号鍵の保管では、OS の標準的なファイル保護機能やプロセス保護機能（Microsoft Windows の ACL、Linux のパーミッションや ACL など）を利用する方法、強固な OS の機能（SELinux[19]など、Linux LSM[20]の保護）を利用する方法、TEE[21]に保管する方法、IC カードや TPM[22]に保管する方法、HSM[23]に保管する方法などがある。なお、これらの機能や方法が提供できるセキュリティ強度には違いがあるので、利用用途や想定される脅威等を踏まえて必要なセキュリティ強度を提供する機能や方法を選択することが重要である。

また、最近では、クラウド鍵管理 SaaS など外部の暗号鍵管理システムの保護方法などを利用することも考えられる。さらに、暗号鍵やメタデータの保護や信頼関係の保護では、前述する安全な暗号鍵の保管が可能な保護方法による信頼プロセスを利用する方法、又は MAC やデジタル

---

19 SELinux（Security-Enhanced Linux）: NSA により開発されたユーザ管理を詳細に制御できるようにした Linux のセキュリティ・アーキテクチャ

20 LSM（Linux Security Module）: Linux カーネルにセキュリティ機能を拡張するためのフレームワーク

21 TEE（Trusted Execution Environment）: チップ上に、通常の OS から独立した、信頼できる隔離実行可能な領域があり、その領域内で処理を完結させることができる環境のこと。その領域内に鍵の保存が可能である。Intel SGX、Arm TrustZone、AMD SEV、RISC-V Keystone などがある

22 TPM（Trusted Platform Module）: 暗号鍵を安全に格納できるセキュリティチップ。TCG（Trusted Computing Group）で規定された仕様には TPM1.2 及び TPM 2.0 が存在する

23 HSM（Hardware Security Module）: 耐タンパ性があり安全に暗号鍵管理や暗号処理をするモジュール

署名などの暗号メカニズムを使う方法を用いて実現する。

4.3 節との関係では、D.02、D.03 で信頼関係や鍵の保護方法や利用手段をどのように設定するかを高レベルの概要で整理し明らかにしておくことで、具体的な保護方法や利用手段について、2.3 節⑰及び 4.3 節に記載する内容との整合性を満たすことが重要である。ここでの「高レベルの概要」の意味は、4.3 節で決める事項を検討する際に本概要で定めたことと矛盾していないことが確認できる程度に具体化した情報、ということである。

加えて、D.03 で設定する情報や利用する各手段については、2 章で定める内容と矛盾なく、整合的であるようにする必要がある。例えば、鍵生成手段であれば 2.3 節⑧と、鍵確立手段であれば 2.5 節と、保管手段であれば 2.4 節①②と、鍵保証であれば 2.3 節⑩⑪とそれぞれ整合性が取れていることを確認することが重要である。

また、D.02 では、暗号鍵とメタデータの関連付けの手段として、両者に適切な暗号検証機能を適用し関連性が正しいことを検証する暗号メカニズム、又は物理的なセキュリティ手段により両者の関連性が正しいことを確認する信頼プロセスのいずれを利用しているか（または両方を利用しているか）を明らかにする。ここで、暗号メカニズムを選択した場合は、4.3 節の「暗号学的プロセスを利用するケース」に該当し、D.05～D.07 を記載することとなる。信頼プロセスを選択した場合は、「信頼プロセスを利用するケース」に該当し、D.08～D.10 を記載することになることに留意されたい。

D.04 は、主に自動処理する際に、全ての鍵タイプ及びメタデータが誤りなく適切に使われるようにするため、鍵タイプやメタデータの利用形態を統一化しておくための情報となる。これには、仕様書として明確化しておくほか、利用する標準規格や API などの情報を記載するなどのやり方がある。

これらの情報を記載しておくことにより、例えばある暗号鍵に対して危殆化が発生し、またその疑いが生じた場合においても、その危殆化の範囲をいち早く調査可能となることが期待される。また、監査等の効率も向上することが期待される。

《トイモデルと記載例》

本節のトイモデルでは、図 4-1 の通り、関係者のみがアクセス可能な施設内 Web サーバシステムとする。また、関係者のみが利用するシステムであることから、サーバ証明書を発行する認証局としては施設内にプライベート CA を独自に構築する形をとっている。ただし、トイモデルとしての説明を簡単にするため、ここでは CKMS の対象範囲を Web サーバに絞って定めるものとし、プライベート CA における暗号鍵管理については別途検討するものとする[24]。

[24] プライベート CA での暗号鍵管理においては、設置場所や運用環境、目的などに依存して、要求されるセキュリティレベルは大きく異なる。認証局のセキュリティとして最も高いレベルは、多くの民間認証局（パブリック CA）などが使っている WebTrust CA の基準（WebTrust Principles and criteria）に準拠する水準であるが、プライベート CA ではそこまでの水準を求められるケースはかなり少ない。なお、WebTrust CA の基準に近いセキュリティを確保することが求められるようなケースでは、CA 運用上どのような点に注意する必要があるかを検討する際に WebTrust のガイダンスが参考になる。
https://www.cpacanada.ca/en/business-and-accounting-resources/audit-and-assurance/overview-of-webtrust-services/principles-and-criteria

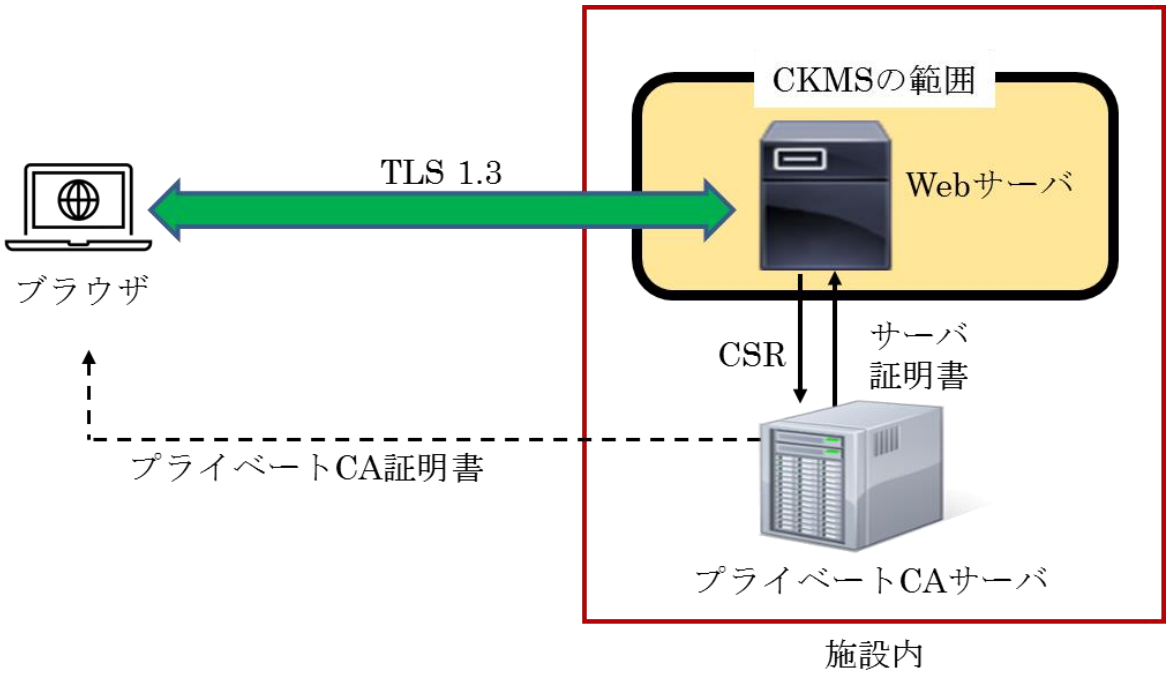


図 4-1　施設内 Web サーバシステム

Web サーバは、OS に Linux、Web サーバの機能を実現するソフトウェアに Apache HTTP Server を使用する。セキュリティプロトコルとして TLS 1.3 だけを許可し、TLS 1.3 を実現するソフトウェアには OpenSSL を使用する。

TLS における暗号鍵を適切に設定するために「TLS 暗号設定ガイドライン」などを参照し、証明書の署名方式は ECDSA NIST P-256 を使用し、鍵交換（鍵合意）の ECDH は X25519、X448、P-256、P-386、P-521 を使用する。なお、この設定は「TLS 暗号設定サーバ設定編」の Apache に関係する部分も参考にしている。

D.01 に明記する鍵タイプは、解説・考慮点に記載したように、CKMS 設計者が明示的に選択して管理する鍵である。本モデルでは、Web サイト正当性のための Web サーバの署名用のプライベート鍵と公開鍵が記載対象となる。また、TLS 1.3 の鍵交換（鍵合意）の ECDH 関係の鍵も記載対象となる。なお、OpenSSL 内部で生成する TLS セッションで利用される鍵は管理対象外となる。また、Web サーバの署名用のプライベート鍵と公開鍵は、PKCS#8（RFC 5958）形式のパスワード付き暗号ファイルで保存する。この暗号化には AES128 ビットの CBC モードを利用する。

ここで使用する証明書はサイトの正当性のためなので、証明書の key usage は電子署名（Digital Signature）を含まなければならない。なお、証明書はここで指定した用途以外では使用できないことに留意されたい。

以上のトイモデルにおける記載例は、以下の「施設内 Web サーバシステムにおける記載例」のようになる。

施設内 Web サーバシステムにおける記載例

D.01	[Web サーバが利用する鍵タイプ] ● 署名アルゴリズム：ECDSA（P-256） ● 署名プライベート鍵（Private Signature Key） ● 署名公開鍵（Public Signature Key） [鍵交換（鍵合意）で利用される鍵タイプ] ● 鍵交換（鍵合意）アルゴリズム：ECDH（X25519, X448, P-256, P-386, P-521） ● 鍵合意一時的プライベート鍵（Private Ephemeral Key Agreement Key） ● 鍵合意一時的公開鍵（Public Ephemeral Key Agreement Key）
D.02	● 署名プライベート鍵は、PKCS#8（RFC 5958）形式のパスワード付き暗号ファイルで保存する。ファイル暗号化には AES128-CBC を利用する。このファイルは Linux の root 権限で保存し、root 権限の無いユーザは Linux の ACL でアクセスすることができないようにしている ● 署名公開鍵の完全性保護のために、証明書に記載されている署名公開鍵とメタデータ（subjectAltName, subject key id など）は暗号メカニズム（デジタル署名）により関連付けられる
D.03	[署名プライベート鍵と署名公開鍵] a) 鍵タイプ ● 署名プライベート鍵（Private Signature Key） ● 署名公開鍵（Public Signature Key） b) 暗号鍵有効期間 ● 証明書の有効期間（署名プライベート鍵、署名公開鍵の有効期間） ● 日本時間 2023 年 5 月 8 日 00:00:00 ～ 日本時間 2024 年 6 月 7 日 23:59:59 c) 生成手段 i) 使用した乱数生成器（RNG）：Linux の乱数生成の疑似デバイスである /dev/random を使用する ii) 鍵生成の仕様：FIPS186-4 d) それぞれのメタデータ要素に対して、以下を含める i) メタデータのソース メタデータ要素（common name, 組織名、有効期間、subjectAltName, subject key id など）のソースは、証明書を作成したときの Web サーバの FQDN、IP アドレス、使用する鍵やアルゴリズムなどの各種設定情報により決定する ii) メタデータの検証方法 証明書に記述されているメタデータの内容検証のために、プライベート CA から始まる証明書パスの有効性検証、及び FQDN と IP アドレスの一致確認を実施する e) 鍵確立（key establishment）の手段：対象外 f) 暴露に対する保護（例えば、鍵の機密性、物理セキュリティ） 署名プライベート鍵は PKCS#8（RFC 5958）形式のパスワード付き暗号化状態で

	保存している。このファイルは Linux の root を所有者にして保存し、ルート以外のユーザは Linux の ACL でアクセスすることができないようにしている g) 改ざんに対する保護（例えば、MAC 又はデジタル署名） 証明書はプライベート CA によるデジタル署名により保護する h) 鍵を使用し得るアプリケーション Web Server（Apache HTTP Server）、TLS 1.3 i) 鍵の使用が許可されないアプリケーション h)で指定したアプリケーション以外は使用を許可しない j) 鍵保証（key assurances） i) 対称鍵保証（Symmetric key assurances） 該当無し ii) 非対称鍵保証（Asymmetric key assurances） 証明書発行者（Issuer）の署名により署名プライベート鍵を保証する iii) ドメインパラメタ有効性チェック 証明書の ECDSA 署名で使用している楕円曲線 P-256 はパラメタを OID で示している。ドメインパラメタの値自体を保管、伝送しないので、有効性は保証される [鍵合意一時的プライベート鍵と公開鍵] a) 鍵タイプ ● 鍵合意一時的プライベート鍵（Private Ephemeral Key Agreement Key） ● 鍵合意一時的公開鍵（Public Ephemeral Key Agreement Key） b) 暗号鍵有効期間 一時的 c) 生成手段 i) 使用した乱数生成器（RNG）：Linux の乱数生成の疑似デバイスである /dev/random を使用 ii) 鍵生成の仕様：SP800-56A Rev.3 d) それぞれのメタデータ要素に対して、以下を含める i) メタデータのソース TLS 1.3 のプロトコル ii) メタデータの検証方法 TLS 1.3 のプロトコルで検証 e) 鍵確立（key establishment）の手段 i) 鍵配送スキーム：使用していない ii) 鍵合意スキーム：ECDH（X25519, X448, P-256, P-386, P-521） iii) プロトコル名：TLS 1.3 f) 暴露に対する保護（例えば、鍵の機密性、物理セキュリティ） 鍵は、OpenSSL が動作するプロセスのメモリ上で一時的に使用され即破棄される g) 改ざんに対する保護（例えば、MAC 又はデジタル署名）

	鍵合意一時的プライベート鍵は OpenSSL が動作するプロセスのメモリ上だけで使用されることにより保護される。鍵合意一時的公開鍵も受信後はメモリ上で処理されることにより保護される h) 鍵を使用し得るアプリケーション Web Server（Apache HTTP Server）、TLS 1.3 i) 鍵の使用が許可されないアプリケーション h)で指定したアプリケーション以外は使用を許可しない j) 鍵保証（key assurances） i) 対称鍵保証（Symmetric key assurances） 該当無し ii) 非対称鍵保証（Asymmetric key assurances） ECDH の場合、当該楕円曲線上の点であり、特異な点でないことを確認する iii) ドメインパラメタ有効性チェック 鍵合意は、TLS 1.3 で規定されているドメインパラメタを使用している。規定された番号で合意し、ドメインパラメタの値自体を伝送しないので、有効性は保証される
D.04	● 署名のプライベート鍵と公開鍵の保管方法は PKCS#8 形式のパスワード付き暗号化状態で保存する ● 証明書は X.509 を使用する。伝送は、TLS 1.3 を使用する

4.3 鍵情報の保護方針

① メタデータ要素内に含まれている情報の保護方法の決定

● 暗号学的プロセスを利用する場合

項目	FR 番号	Framework Requirements の内容	SP800-130
D.05	FR6.3	メタデータ要素の鍵保護（Key Protections）で使用されるそれぞれの暗号メカニズムに対して、CKMS 設計は、以下を明記しなければならない： i. 暗号アルゴリズム ii. 鍵パラメタ iii. 鍵識別子 iv. 保護値（protection value）：この要素は、完全性保護、機密性保護、又はソース認証（source authentication）の保護値（protection value）を含む。例えば、適切に実装された MAC 又はデジタル署名技術は、完全性保護やソース認証（source authentication）を提供し得る。 v. 保護が適用された時期	6.2.1 節

		vi. 保護が検証された時期	
D.06	FR6.5	メタデータ要素のメタデータ保護（Metadata Protections）で使用されるそれぞれの暗号メカニズムに対して、CKMS 設計は、以下を明記しなければならない： i. 暗号アルゴリズム ii. 鍵パラメタ iii. 鍵識別子 iv. 保護値（protection value）（例：MAC、デジタル署名） v. 保護が適用された時期 vi. 保護が検証された時期 一般に、特に鍵とメタデータがひとまとめにされる場合、鍵とメタデータに対して同じメカニズムが使用される。	6.2.1 節
D.07	FR6.7	メタデータ要素の信頼関係保護で使用されるそれぞれの暗号メカニズムに対して、CKMS 設計は、以下を明記しなければならない： i. 暗号アルゴリズム ii. 鍵パラメタ iii. 鍵識別子 iv. 保護値（protection value）（例：MAC、デジタル署名） v. 保護が適用された時期 vi. 保護が検証された時期	6.2.1 節

- 信頼プロセスを利用する場合

項目	FR 番号	Framework Requirements の内容	SP800-130
D.08	FR6.4	メタデータ要素の鍵保護（Key Protections）で使用される暗号学的ではないそれぞれの信頼プロセスに対して、CKMS 設計は、以下を明記しなければならない： i. 他のプロセスと区別するために使用されるプロセス識別子 ii. プロセスの説明又はプロセスの説明へのポインタ	6.2.1 節
D.09	FR6.6	メタデータ要素のメタデータ保護（Metadata Protections）で使用される暗号学的ではないそれぞれの信頼プロセスに対して、CKMS 設計は、以下を明記しなければならない： i. このプロセスを他のプロセスから区別するために使用される識別子 ii. プロセスの説明又はプロセスの説明へのポインタ	6.2.1 節
D.10	FR6.8	メタデータ要素の信頼関係保護で使用される暗号学的ではないそれぞれの信頼プロセスに対して、CKMS 設計は、以下を明記しなければならない：	6.2.1 節

		i. このプロセスを他のプロセスから区別するために使用される識別子 ii. プロセスの説明又はプロセスの説明へのポインタ	

解説・考慮点

メタデータ要素内の暗号鍵、メタデータ及びそれらの信頼関係に関して、暗号学的プロセス又は信頼プロセスのいずれかにより保護しなければならない。
CKMS の設計にあたって、項目 D.05～D.07 は、暗号学的プロセスを利用する場合の保護方法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。D.05 は暗号鍵の保護、D.06 はメタデータの保護、D.07 は信頼関係の保護を対象にしている。
項目 D.08～D.10 は、信頼プロセスを利用する場合の保護方法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。D.05～D.07 と同様、D.08 は暗号鍵の保護、D.09 はメタデータの保護、D.10 は信頼関係の保護を対象にしている。
なお、D.05～D.07、D.08～D.10 のいずれかが対象である。

CKMS 設計者が管理する必要がある全ての暗号鍵とメタデータは、鍵タイプや使用用途により適切な保護が必要であり、例えば、プライベート鍵は復号や署名するエンティティ以外には知られることが無いように保管することが必要である。また、鍵タイプによってはメタデータとの組み合わせによりその暗号鍵の完全性を確保する場合もあるため、とりわけそのような鍵タイプの暗号鍵ではメタデータの管理も安全な鍵情報の管理を実施するために重要となる。そのため、D.02 で暗号鍵とメタデータの関連付けの手段を利用するとしたメタデータについては、どのような関連付けの手段を利用し、両者の関連性が正しいことを保証するかについての具体的な方法を示すことが求められる。

本節では、暗号鍵やメタデータ、信頼関係の保護方針を明確にすることで、鍵情報が安全に管理されていることを確認することを意図している。

具体的には、保護手段として暗号メカニズムを使用している場合は、そこで利用している具体的な暗号アルゴリズムや鍵パラメタなど、D.05～D.07 の項目を明確化する必要である。D.05 は暗号鍵に対する、D.06 は関連付けられたメタデータに対する、完全性、機密性及びソース認証（source authentication）の保護が対象であり、D.07 は、暗号鍵とメタデータが正しく関連付けられていることを保証するために、その暗号鍵とメタデータとの信頼関係の保護が対象である。なお、これらは同じメカニズムで保護される場合もあれば、別のメカニズムで保護することもある。さらには、暗号鍵とメタデータが同時に関連付けて保護され、信頼関係保護が暗黙的に提供されている場合もある。

保護手段として信頼プロセスを使用している場合は、そこで利用している具体的なプロセスについて D.08～D.10 の項目を明確化することが必要である。

なお、以上の内容は、D.02、D.03 での記載内容と整合していなければならない。

《トイモデルと記載例》

本節のトイモデルは 4.2 節と同じである。このトイモデルでは、D.02 において、「信頼プロセスはプライベート鍵の保護のために root 権限でプライベート鍵ファイルを保存し、OS のユーザ管理機能で root 以外のユーザはファイルにアクセスできなくし、暗号学的プロセスはデジタル署名を利用して公開鍵やメタデータの完全性を保護する」こととしている。

その方針に従い、本トイモデルが利用している、具体的な暗号メカニズムを使用した保護手段についての情報を D.05～D.07 に、具体的な信頼プロセスを使用した保護手段についての情報を D.08～D.010 にそれぞれ記載する。

以上のトイモデルにおける記載例は、以下の「施設内 Web サーバシステムにおける記載例」のようになる。

施設内 Web サーバシステムにおける記載例

D.05	署名公開鍵の保護 i. 暗号アルゴリズム ECDSA ii. 鍵パラメタ P-256 iii. 鍵識別子 公開鍵のハッシュ値 iv. 保護値（protection value） デジタル署名 v. 保護が適用された時期 日本時間 2023 年 5 月 8 日 00:00:00 vi. 保護が検証された時期 日本時間 2023 年 5 月 8 日 00:00:00
D.06	署名の鍵ペアのメタデータの保護 i. 暗号アルゴリズム ECDSA ii. 鍵パラメタ P-256 iii. 鍵識別子 公開鍵のハッシュ値 iv. 保護値（protection value） デジタル署名 v. 保護が適用された時期 日本時間 2023 年 5 月 8 日 00:00:00 vi. 保護が検証された時期 日本時間 2023 年 5 月 8 日 00:00:00

D.07	署名の鍵ペアとメタデータの関係性の保護 i. 暗号アルゴリズム ECDSA ii. 鍵パラメタ P-256 iii. 鍵識別子 公開鍵のハッシュ値 iv. 保護値（protection value） デジタル署名 v. 保護が適用された時期 日本時間 2023 年 5 月 8 日 00:00:00 vi. 保護が検証された時期 日本時間 2023 年 5 月 8 日 00:00:00

D.08	Web サーバでの署名プライベート鍵のファイルは、root 権限でのみアクセス可能。このアクセス権限は Linux のパーミッションを利用して実現している
D.09	Web サーバでの署名処理の入力として利用されるメタデータの管理に関するファイルは、root 又は Web サーバ管理ユーザ権限でのみアクセス可能。このアクセス権限は Linux のパーミッションを利用して実現している
D.10	D.09 での保護方法により、メタデータ要素の信頼関係も保護している

資料５－１

CRYPTREC暗号リストの改定について

今次のCRYPTREC暗号リスト改定（旧“全面改定”）に至る主な経緯

1. リストの全面改定を実施、今後のリスト改定方針の大枠を決定（2012年度暗号技術検討会）

・2003年に策定した「電子政府推奨暗号リスト」を、「電子政府推奨暗号リスト」、「推奨候補暗号リスト」、「運用監視暗号リスト」の３リストで構成する「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）」に再構成。

・「CRYPTREC暗号リスト」の次回の全面改定は10年後を目途に実施する等、今後のリスト改定方針の大枠を決定。

2. 2023年目途のリスト改定方針を決定（2020年度暗号技術検討会）

・2023年目途のリスト改定に当たり、３リスト構成を維持する方針、技術分類の構成を維持する方針を決定。

・上記改定に当たり、暗号技術の公募は行わない方針を決定。

3. 2023年目途のリスト改定等に適用する新たな基準を決定（2021年度暗号技術検討会）

・「推奨候補暗号リスト」から「電子政府推奨暗号リスト」へ昇格させる暗号アルゴリズムを選定する際の、暗号アルゴリズムの利用実績による選定基準（別紙１）を決定。

※ 上記基準決定後の「CRYPTREC 暗号リスト」移行ルールは別紙２のとおり。

4. 2023年のリスト改定に係る具体的改定内容を検討（2022年度暗号技術活用委員会）

・ 2022年度に暗号技術活用委員会が実施した暗号アルゴリズム利用実績調査の結果を踏まえ、暗号技術活用委員会において、「推奨候補暗号リスト」から「電子政府推奨暗号リスト」への昇格候補として暗号技術検討会に推薦する暗号アルゴリズム（別紙３）を決定。

5. 2023年のリスト改定に係る意見募集の公示案を決定（2022年度第1回暗号技術検討会）

・暗号技術活用委員会において決定された暗号技術検討会に推薦する暗号アルゴリズム（別紙３）を踏まえて、意見募集において公示するCRYPTREC暗号リストの改定案を決定した。

「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)」(案)に対する意見募集に寄せられたご意見並びにそれらに対するデジタル庁、総務省及び経済産業省の考え方（暗号技術検討会事務局案）（資料5-2）及びCRYPTREC暗号リスト(改定版)(暗号技術検討会事務局案)(資料5-3)をご審議いただきたい。

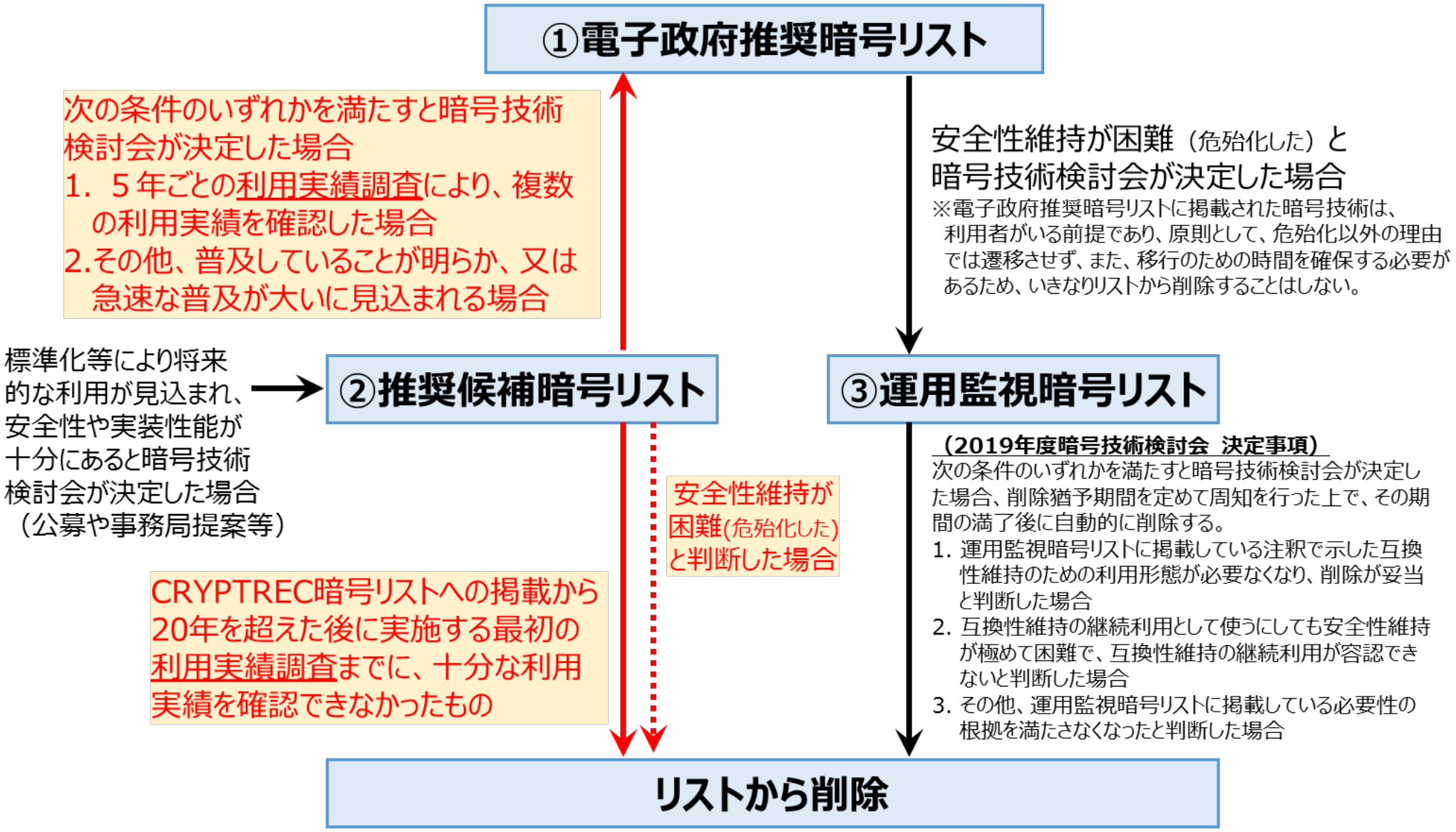
暗号アルゴリズムの利用実績による選定基準

別紙１

考慮項目		選定目安
採用実績	以下のいずれかを満たす場合、昇格の検討対象に含める。なお、採用実績は、 ● 5年ごとに実施予定の大規模アンケート調査による「**利用実績調査**」 ● 必要に応じて、事務局が（大規模アンケート調査によらずに）情報収集する「**利用実態確認**」 により確認するものとする。	
	① **利用実績調査**の結果、電子政府推奨暗号リストに掲載されている（同一カテゴリの）暗号技術の採用実績と遜色がないことが確認された場合	**電子政府推奨暗号リスト掲載の（同一カテゴリの）暗号技術の採用実績と同等以上の採用実績がある推奨候補暗号リスト掲載の暗号技術を昇格検討対象とする。**
	② **利用実績調査**又は**利用実態確認**の結果、**電子政府システムや重要インフラ等、日本の基幹システム**においてすでに利用されていることが確認された場合	**必要に応じて、利用実績調査に代わって、各府省庁等への照会を実施し、照会結果（クローズドな利用を含め）を基に昇格検討対象を選定する。**
	利用実績調査又は**利用実態確認**の結果、③～⑤のいずれかが確認された場合： ③ **利用者が多い主要な汎用製品群の複数**に搭載されるなど、明らかに採用が進展していると判断された場合 ④ **利用者が多い主要なオープンソースソフトウェアの複数**に搭載されるなど、明らかに採用が進展していると判断された場合 ⑤ **利用者が多い主要なサービスやプロトコルの複数**で利用されるなど明らかに採用が進展していると判断された場合	**「複数」「利用者が多い（主要な）」というキーワードの両方を十分に満たし、明らかな採用促進が確認された場合には、必要に応じて、昇格検討対象とする。** **※「複数」の意味は、必要条件として「2個以上が必要」ということであって、「2個以上あればよい」という十分条件としての意味ではないことに留意**
標準化実績	以下を満たす場合、昇格の検討対象に含める。	
	⑥ **利用実績調査**の結果、電子政府推奨暗号リストに掲載されている（同一カテゴリの）暗号技術の採用実績と遜色がないことが確認された場合	**電子政府推奨暗号リスト掲載の（同一カテゴリの）暗号技術の採用実績と同等以上の採用実績がある推奨候補暗号リスト掲載の暗号技術は昇格検討対象とする。**

CRYPTREC 暗号リスト移行ルール

別紙２



暗号技術検討会に推薦する暗号アルゴリズム

別紙３

暗号アルゴリズムの技術分類	推薦する暗号アルゴリズム
公開鍵暗号（署名）	EdDSA
ハッシュ関数	SHA-512/256 SHA3-256 SHA3-384 SHA3-512 SHAKE128 SHAKE256
暗号利用モード（秘匿モード）	XTS
認証暗号	ChaCha20-Poly1305
エンティティ認証	ISO/IEC 9798-4

資料５－２

「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）」（案）に対する意見募集に寄せられたご意見並びにそれらに対するデジタル庁、総務省及び経済産業省の考え方（暗号技術検討会事務局案）

「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）」（案）

に対する意見の募集で寄せられたご意見について

○ 意見募集期間：令和5年3月9日～令和5年3月23日

○ 提出意見総数： 4件

（1） 個人 4件

（2） 法人・団体 0件

項目	頂いたご意見	ご意見に対する考え方
【意見1】入力データについて		
全般	（意見）アルファベットの大文字と小文字を区別して認識するタイプの暗号を導入するのは、どうでしょうか。 （理由）より、セキュリティレベルが上ると思うから、です。 【個人1－1】	CRYPTREC 暗号リストに掲載している暗号技術では、入力データや鍵等はビット列として取り扱います。
【意見2】表の体裁について		
電子政府推奨暗号リスト	（意見）表の左上の枠を加えるのは、どうでしょうか。 （理由）枠が抜けているため、です。 【個人1－2】	ご指摘を踏まえ、表の改ページ箇所を修正いたしました。

項目	頂いたご意見	ご意見に対する考え方
【意見3】DSA について		
電子政府推奨暗号リスト	（意見）DSA に関しては「電子政府推奨暗号リスト」ではなく「運用監視暗号リスト」にリストするべきではないか？ （理由）NIST FIPS186-5（ドラフト）では以下の扱いとなっているため。 デジタル署名アルゴリズム (DSA) この標準の以前のバージョンでは、DSA が指定されていました。この規格は、DSA を承認しなくなりました。 デジタル署名の生成。 DSA は、事前に生成された署名を検証するために使用できます。 この規格の実装日。 DSA の仕様については、FIPS 186-4 [20] を参照してください。 ―― 4 The Digital Signature Algorithm (DSA) Prior versions of this standard specified the DSA. This standard no longer approves DSA for digital signature generation. DSA may be used to verify signatures generated prior to the implementation date of this standard. See FIPS 186-4 [20] for the specifications for DSA. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5-draft.pdf 【個人2】	頂きました御意見は、今後の検討課題とさせて頂きます。

項目	頂いたご意見	ご意見に対する考え方
【意見４】SHA-1 について		
運用監視暗号リスト	SHA-2 への移行が十分進んだことを考慮し、SHA-1 は廃止してよいと思う。 【個人３－１】	頂きました御意見は、今後の検討課題とさせて頂きます。
【意見５】3-key Triple DES について		
運用監視暗号リスト	AES への移行が十分に進んだ点や、現在脆弱性が指摘されている点も加味すると、3DES は廃止するべきだと思う。 【個人３－２】	頂きました御意見は、今後の検討課題とさせて頂きます。
【意見６】CBC について		
電子政府推奨暗号リスト	以下、「「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）」の改定案」に意見を行う。 ＞1 頁 ＞電子政府推奨暗号リスト 「暗号利用モード」の「秘匿モード」に「CBC」が含まれているが、CBC は、かなりの数、かなり多くの場合（TLS1.2 を含んでいる場合も多い）で危険という報告が行われているので、電子政府推奨暗号リストへの掲載は停止すべきと考える。 （条件によっては脆弱性が発揮されない場合もあると思われるが、そもそも CBC を利用しない方が安全と思われる。TLS などにおいては（TLS1.2 以降において）基本として全く CBC を利用しない方が望ましいと思われる。CBC の掲載は問題あるものと思われる。） 意見は以上である。 【個人４】	暗号利用モード（秘匿モード）CBC の暗号技術自体の安全性は CRYPTREC により確認されており、電子政府推奨暗号リストへの掲載を維持することが適当と考えます。 なお、暗号技術の利用時に安全性を確保する方法については、CRYPTREC で作成している「TLS 暗号設定ガイドライン」等において示しています。

資料５－３

CRYPTREC LS-0001-2022

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）（暗号技術検討会事務局案）

令和●年●月●日

デジタル庁・総務省・経済産業省

電子政府推奨暗号リスト

暗号技術検討会[1]及び関連委員会（以下、「CRYPTREC」という。）により安全性及び実装性能が確認された暗号技術[2]について、市場における利用実績が十分であるか今後の普及が見込まれると判断され、当該技術の利用を推奨するもののリスト。なお、利用する鍵長について、「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」[5]の規定に合致しない鍵長を用いた場合には、電子政府推奨暗号リストの暗号技術を利用しているとは見なされないことに留意すること。

技術分類		暗号技術
公開鍵暗号	署名	DSA
		ECDSA
		EdDSA
		RSA-PSS(注1)
		RSASSA-PKCS1-v1_5(注1)
	守秘	RSA-OAEP(注1)
	鍵共有	DH
		ECDH
共通鍵暗号	64ビットブロック暗号(注2)	該当なし
	128ビットブロック暗号	AES
		Camellia
	ストリーム暗号	KCipher-2
ハッシュ関数		SHA-256
		SHA-384
		SHA-512
		SHA-512/256
		SHA3-256
		SHA3-384
		SHA3-512
		SHAKE128(注12)
		SHAKE256(注12)
（次ページに続く）		

[1]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。

[2] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合、CRYPTREC暗号リストに掲載されたいずれかの暗号技術と組み合わせること。

[5] 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, https://www.cryptrec.go.jp/list.html

技術分類		暗号技術
暗号利用モード	秘匿モード	CBC
		CFB
		CTR
		OFB
		XTS(注17)
	認証付き秘匿モード(注13)	CCM
		GCM(注4)
メッセージ認証コード		CMAC
		HMAC
認証暗号		ChaCha20-Poly1305
エンティティ認証		ISO/IEC 9798-2
		ISO/IEC 9798-3
		ISO/IEC 9798-4

（注1） 「政府機関の情報システムにおいて使用されている暗号アルゴリズムSHA-1及びRSA1024に係る移行指針」（平成20年4月情報セキュリティ政策会議決定、平成24年10月情報セキュリティ対策推進会議改定）を踏まえて利用すること。
https://www.nisc.go.jp/pdf/policy/general/angou_ikoushishin.pdf
（平成25年3月1日現在）

（注2） CRYPTREC暗号リストにおいて、64ビットブロック暗号により、同一の鍵を用いて暗号化する場合、2^{20}ブロックまで、同一の鍵を用いてCMACでメッセージ認証コードを生成する場合、2^{21}ブロックまでとする。

（注4） 初期化ベクトル長は96ビットを推奨する。

（注12） ハッシュ長は256ビット以上とすること。

（注13） CRYPTREC暗号リスト掲載のブロック暗号を、認証付き秘匿モードと組み合わせて、「認証暗号」として使うことができる。

（注17） ブロック暗号には、CRYPTREC暗号リスト掲載128ビットブロック暗号を使う。利用用途はストレージデバイスの暗号化に限り、実装方法はNIST SP800-38Eに従うこと。

推奨候補暗号リスト

CRYPTRECにより安全性及び実装性能が確認され、今後、電子政府推奨暗号リストに掲載される可能性のある暗号技術[3]のリスト。なお、本リストに記載されている暗号技術を利用する際は、「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」[6]の規定に合致する鍵長を用いることが求められることに留意すること。

技術分類		暗号技術
公開鍵暗号	署名	該当なし
	守秘	該当なし
	鍵共有	PSEC-KEM（注5）
共通鍵暗号	64ビットブロック暗号（注6）	CIPHERUNICORN-E
		Hierocrypt-L1
		MISTY1
	128ビットブロック暗号	CIPHERUNICORN-A
		CLEFIA
		Hierocrypt-3
	ストリーム暗号	Enocoro-128v2
		MUGI
		MULTI-S01（注7）
ハッシュ関数		該当なし
暗号利用モード	秘匿モード	該当なし
	認証付き秘匿モード（注14）	該当なし
メッセージ認証コード		PC-MAC-AES
認証暗号		該当なし
エンティティ認証		該当なし

（注5） KEM (Key Encapsulating Mechanism) – DEM (Data Encapsulating Mechanism) 構成における利用を前提とする。

（注6） CRYPTREC暗号リストにおいて、64ビットブロック暗号により、同一の鍵を用いて暗号化する場合、2^{20}ブロックまで、同一の鍵を用いてCMACでメッセージ認証コードを生成する場合、2^{21}ブロックまでとする。

（注7） 平文サイズは64ビットの倍数に限る。

（注14） CRYPTREC暗号リスト掲載のブロック暗号を、認証付き秘匿モードと組み合わせて、「認証暗号」として使うことができる。

[3] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合、CRYPTREC暗号リストに掲載されたいずれかの暗号技術と組み合わせること。

[6] 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, https://www.cryptrec.go.jp/list.html

運用監視暗号リスト

実際に解読されるリスクが高まるなど、推奨すべき状態ではなくなったとCRYPTRECにより確認された暗号技術[4]のうち、互換性維持のために継続利用を容認するもののリスト。互換性維持[7]以外の目的での利用は推奨しない。なお、本リストに記載されている暗号技術を利用する際は、「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」[8]の規定に合致する鍵長を用いることが求められることに留意すること。

技術分類		暗号技術
公開鍵暗号	署名	該当なし
	守秘	RSAES-PKCS1-v1_5（注8）（注9）
	鍵共有	該当なし
共通鍵暗号	64ビットブロック暗号（注15）	3-key Triple DES
	128ビットブロック暗号	該当なし
	ストリーム暗号	該当なし
ハッシュ関数		RIPEMD-160
		SHA-1（注8）
暗号利用モード	秘匿モード	該当なし
	認証付き秘匿モード（注16）	該当なし
メッセージ認証コード		CBC-MAC（注11）
認証暗号		該当なし
エンティティ認証		該当なし

（注8） 「政府機関の情報システムにおいて使用されている暗号アルゴリズムSHA-1及びRSA1024に係る移行指針」（平成20年4月 情報セキュリティ政策会議決定、平成24年10月情報セキュリティ対策推進会議改定）を踏まえて利用すること。
https://www.nisc.go.jp/pdf/policy/general/angou_ikoushishin.pdf
（平成25年3月1日現在）

（注9） TLS 1.0, 1.1, 1.2で利用実績があることから当面の利用を認める。

（注11） 安全性の観点から、メッセージ長を固定して利用すべきである。

（注15） CRYPTREC暗号リストにおいて、64ビットブロック暗号により、同一の鍵を用いて暗号化する場合、2^{20}ブロックまで、同一の鍵を用いてCMACでメッセージ認証コードを生成する場合、2^{21}ブロックまでとする。

（注16） CRYPTREC暗号リスト掲載のブロック暗号を、認証付き秘匿モードと組み合わせて、「認証暗号」として使うことができる。

[4] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合、CRYPTREC暗号リストに掲載されたいずれかの暗号技術と組み合わせること。
[7] 既に稼働中のシステムやアプリケーション等との間での相互運用を継続すること
[8] 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, https://www.cryptrec.go.jp/list.html

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）

平成２５年３月１日
デジタル庁・総務省・経済産業省
（最終更新：令和5年3月8日）

電子政府推奨暗号リスト

暗号技術検討会[1]及び関連委員会（以下、「CRYPTREC」という。）により安全性及び実装性能が確認された暗号技術[2]について、市場における利用実績が十分であるか今後の普及が見込まれると判断され、当該技術の利用を推奨するもののリスト。なお、利用する鍵長について、「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」[5]の規定に合致しない鍵長を用いた場合には、電子政府推奨暗号リストの暗号技術を利用しているとは見なされないことに留意すること。

技術分類		暗号技術
公開鍵暗号	署名	DSA
		ECDSA
		RSA-PSS(注1)
		RSASSA-PKCS1-v1_5(注1)
	守秘	RSA-OAEP(注1)
	鍵共有	DH
		ECDH
共通鍵暗号	64ビットブロック暗号(注2)	該当なし
	128ビットブロック暗号	AES
		Camellia
	ストリーム暗号	KCipher-2
ハッシュ関数		SHA-256
		SHA-384
		SHA-512
暗号利用モード	秘匿モード	CBC
		CFB
		CTR
		OFB
	認証付き秘匿モード(注13)	CCM
		GCM(注4)
メッセージ認証コード		CMAC
		HMAC
認証暗号		該当なし
エンティティ認証		ISO/IEC 9798-2
		ISO/IEC 9798-3

[1]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。

[2] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合、CRYPTREC暗号リストに掲載されたいずれかの暗号技術と組み合わせること。

[5] 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, https://www.cryptrec.go.jp/list.html

（注1） 「政府機関の情報システムにおいて使用されている暗号アルゴリズムSHA-1及びRSA1024に係る移行指針」（平成20年４月情報セキュリティ政策会議決定、平成24年10月情報セキュリティ対策推進会議改定）を踏まえて利用すること。
https://www.nisc.go.jp/active/general/pdf/angou_ikoushishin.pdf
（平成25年３月１日現在）

（注2） CRYPTREC暗号リストにおいて、64ビットブロック暗号により、同一の鍵を用いて暗号化する場合、2^{20}ブロックまで、同一の鍵を用いてCMACでメッセージ認証コードを生成する場合、2^{21}ブロックまでとする。

（注4） 初期化ベクトル長は96ビットを推奨する。

（注13） CRYPTREC暗号リスト掲載のブロック暗号を、認証付き秘匿モードと組み合わせて、「認証暗号」として使うことができる。

推奨候補暗号リスト

CRYPTRECにより安全性及び実装性能が確認され、今後、電子政府推奨暗号リストに掲載される可能性のある暗号技術[3]のリスト。なお、本リストに記載されている暗号技術を利用する際は、「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」[6]の規定に合致する鍵長を用いることが求められることに留意すること。

技術分類		暗号技術
公開鍵暗号	署名	EdDSA
	守秘	該当なし
	鍵共有	PSEC-KEM(注5)
共通鍵暗号	64ビットブロック暗号(注6)	CIPHERUNICORN-E
		Hierocrypt-L1
		MISTY1
	128ビットブロック暗号	CIPHERUNICORN-A
		CLEFIA
		Hierocrypt-3
	ストリーム暗号	Enocoro-128v2
		MUGI
		MULTI-S01(注7)
ハッシュ関数		SHA-512/256
		SHA3-256
		SHA3-384
		SHA3-512
		SHAKE128(注12)
		SHAKE256(注12)
暗号利用モード	秘匿モード	XTS(注17)
	認証付き秘匿モード(注14)	該当なし
メッセージ認証コード		PC-MAC-AES
認証暗号		ChaCha20-Poly1305
エンティティ認証		ISO/IEC 9798-4

（注5） KEM (Key Encapsulating Mechanism) – DEM (Data Encapsulating Mechanism) 構成における利用を前提とする。

（注6） CRYPTREC暗号リストにおいて、64ビットブロック暗号により、同一の鍵を用いて暗号化する場合、2^{20}ブロックまで、同一の鍵を用いてCMACでメッセージ認証コードを生成する場合、2^{21}ブロックまでとする。

（注7） 平文サイズは64ビットの倍数に限る。

（注12） ハッシュ長は256ビット以上とすること。

（注14） CRYPTREC暗号リスト掲載のブロック暗号を、認証付き秘匿モードと組み合わせて、「認証暗号」として使うことができる。

（注17） ブロック暗号には、CRYPTREC暗号リスト掲載128ビットブロック暗号を使う。利用用途はストレージデバイスの暗号化に限り、実装方法はNIST SP800-38Eに従うこと。

[3] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合、CRYPTREC暗号リストに掲載されたいずれかの暗号技術と組み合わせること。

[6] 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, https://www.cryptrec.go.jp/list.html

運用監視暗号リスト

実際に解読されるリスクが高まるなど、推奨すべき状態ではなくなったとCRYPTRECにより確認された暗号技術[4]のうち、互換性維持のために継続利用を容認するもののリスト。互換性維持[7]以外の目的での利用は推奨しない。なお、本リストに記載されている暗号技術を利用する際は、「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」[8]の規定に合致する鍵長を用いることが求められることに留意すること。

技術分類		暗号技術
公開鍵暗号	署名	該当なし
	守秘	RSAES-PKCS1-v1_5（注8）（注9）
	鍵共有	該当なし
共通鍵暗号	64ビットブロック暗号（注15）	3-key Triple DES
	128ビットブロック暗号	該当なし
	ストリーム暗号	該当なし
ハッシュ関数		RIPEMD-160
		SHA-1（注8）
暗号利用モード	秘匿モード	該当なし
	認証付き秘匿モード（注16）	該当なし
メッセージ認証コード		CBC-MAC（注11）
認証暗号		該当なし
エンティティ認証		該当なし

（注8） 「政府機関の情報システムにおいて使用されている暗号アルゴリズムSHA-1及びRSA1024に係る移行指針」（平成20年4月 情報セキュリティ政策会議決定、平成24年10月情報セキュリティ対策推進会議改定）を踏まえて利用すること。
https://www.nisc.go.jp/active/general/pdf/angou_ikoushishin.pdf
（平成25年3月1日現在）

（注9） TLS 1.0, 1.1, 1.2で利用実績があることから当面の利用を認める。

（注11） 安全性の観点から、メッセージ長を固定して利用すべきである。

（注15） CRYPTREC暗号リストにおいて、64ビットブロック暗号により、同一の鍵を用いて暗号化する場合、2^{20}ブロックまで、同一の鍵を用いてCMACでメッセージ認証コードを生成する場合、2^{21}ブロックまでとする。

（注16） CRYPTREC暗号リスト掲載のブロック暗号を、認証付き秘匿モードと組み合わせて、「認証暗号」として使うことができる。

[4] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合、CRYPTREC暗号リストに掲載されたいずれかの暗号技術と組み合わせること。
[7] 既に稼働中のシステムやアプリケーション等との間での相互運用を継続すること
[8] 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, https://www.cryptrec.go.jp/list.html

変更履歴情報

変更日付	変更箇所	変更前の記述	変更後の記述
平成27年3月27日	（注10）	128-bit RC4は、SSL（TLS1.0以上）に限定して利用すること。	互換性維持のために継続利用をこれまで容認してきたが、今後は極力利用すべきでない。SSL/TLSでの利用を含め、電子政府推奨暗号リストに記載された暗号技術への移行を速やかに検討すること。
平成28年3月29日	推奨候補暗号リスト（技術分類：ハッシュ関数）	該当なし	SHA-512/256 SHA3-256 SHA3-384 SHA3-512 SHAKE256（注12）
	（注12）	[新規追加]	ハッシュ長は256ビット以上とすること。
平成29年3月30日	推奨候補暗号リスト（技術分類：ハッシュ関数）	SHA-512/256 SHA3-256 SHA3-384 SHA3-512 SHAKE256（注12）	SHA-512/256 SHA3-256 SHA3-384 SHA3-512 SHAKE128（注12） SHAKE256（注12）
平成30年3月29日	（注2） （注6）	より長いブロック長の暗号が利用できるのであれば、128ビットブロック暗号を選択することが望ましい。	CRYPTREC暗号リストにおいて、64ビットブロック暗号により、同一の鍵を用いて暗号化する場合、2^{20}ブロックまで、同一の鍵を用いてCMACでメッセージ認証コードを生成する場合、2^{21}ブロックまでとする。
	（注15）	[新規追加]	
	電子政府推奨暗号リスト（技術分類：共通鍵暗号）	3-key Triple DES（注3）	該当なし
	（注3）	3-key Triple DESは、以下の条件を考慮し、当面の利用を認める。 1） NIST SP 800-67として規定されていること。 2） デファクトスタンダードとしての位置を保っていること。	[削除]
	運用監視暗号リスト（技術分類：共通鍵暗号）	該当なし	3-Key Triple DES（注15）
	電子政府推奨暗号リスト	[技術分類の新設]	技術分類：認証暗号 暗号技術：該当なし
	推奨候補暗号リスト		技術分類：認証暗号 暗号技術：ChaCha20-Poly1305
	運用監視暗号リスト		技術分類：認証暗号 暗号技術：該当なし

	（注13） （注14） （注16）	[新規追加]	CRYPTREC暗号リスト掲載のブロック暗号を、認証付き秘匿モードと組み合わせて、「認証暗号」として使うことができる。
	電子政府推奨暗号リスト（見出し）	名称	暗号技術
	推奨候補暗号リスト（見出し）		
	運用監視暗号リスト（見出し）		
令和２年12月21日	推奨候補暗号リスト（技術分類：暗号利用モード　秘匿モード）	該当なし	XTS（注17）
	（注17）	[新規追加]	ブロック暗号には、CRYPTREC暗号リスト掲載128ビットブロック暗号を使う。利用用途はストレージデバイスの暗号化に限り、実装方法はNIST SP800-38Eに従うこと。
令和３年4月1日	運用監視暗号リスト（技術分類：共通鍵暗号）	128-bit RC4（注10）	該当なし
	（注10）	互換性維持のために継続利用をこれまで容認してきたが、今後は極力利用すべきでない。TLSでの利用を含め、電子政府推奨暗号リストに記載された暗号技術への移行を速やかに検討すること。	[削除]
令和4年3月30日	文書クレジット	総務省・経済産業省	デジタル庁・総務省・経済産業省
	推奨候補暗号リスト（技術分類：公開鍵暗号　署名）	該当なし	EdDSA
	電子政府推奨暗号リスト（本文）	暗号技術検討会及び関連委員会（以下、「CRYPTREC」という。）により安全性及び実装性能が確認された暗号技術について、市場における利用実績が十分であるか今後の普及が見込まれると判断され、当該技術の利用を推奨するもののリスト。	暗号技術検討会及び関連委員会（以下、「CRYPTREC」という。）により安全性及び実装性能が確認された暗号技術について、市場における利用実績が十分であるか今後の普及が見込まれると判断され、当該技術の利用を推奨するもののリスト。なお、利用する鍵長について、「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」の規定に合致しない鍵長を用いた場合には、電子政府推奨暗号リストの暗号技術を利用しているとは見なされないことに留意すること。

	電子政府推奨暗号リスト（本文）	「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」	「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」[5]
	電子政府推奨暗号リスト（脚注）	該当なし	[5] 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, https://www.cryptrec.go.jp/list.html
	推奨候補暗号リスト（本文）	CRYPTRECにより安全性及び実装性能が確認され、今後、電子政府推奨暗号リストに掲載される可能性のある暗号技術のリスト。	CRYPTRECにより安全性及び実装性能が確認され、今後、電子政府推奨暗号リストに掲載される可能性のある暗号技術のリスト。なお、本リストに記載されている暗号技術を利用する際は、「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」の規定に合致する鍵長を用いることが求められることに留意すること。
	推奨候補暗号リスト（本文）	「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」	「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」[6]
	推奨候補暗号リスト（脚注）	該当なし	[6] 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, https://www.cryptrec.go.jp/list.html
	運用監視暗号リスト（本文）	実際に解読されるリスクが高まるなど、推奨すべき状態ではなくなったとCRYPTRECにより確認された暗号技術のうち、互換性維持のために継続利用を容認するもののリスト。互換性維持以外の目的での利用は推奨しない。	実際に解読されるリスクが高まるなど、推奨すべき状態ではなくなったとCRYPTRECにより確認された暗号技術のうち、互換性維持のために継続利用を容認するもののリスト。互換性維持以外の目的での利用は推奨しない。なお、本リストに記載されている暗号技術を利用する際は、「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」の規定に合致する鍵長を用いることが求められることに留意すること。
	運用監視暗号リスト（本文）	互換性維持	互換性維持[7]
	運用監視暗号リスト（脚注）	該当なし	[7] 既に稼働中のシステムやアプリケーション等との間での相互運用を継続すること
	運用監視暗号リスト（本文）	「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」	「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」[8]
	運用監視暗号リスト（脚注）	該当なし	[8] 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, https://www.cryptrec.go.jp/list.html
令和５年３月８日	推奨候補暗号リスト（技術分類：共通鍵暗号）	SC2000	[削除] （提案会社からの取下げ申請を承認したことによる）

資料６

2023 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

下記の通り、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。

- 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。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当該ワーキンググループで検討し、更新を行う。
- 2021年度に承認された「軽量暗号ガイドライン更新方針」に従って、「CRYPTREC暗号技術ガイドライン(軽量暗号)」2023 年度版の案を完成させる。

(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、2回の開催を予定する。

回	開催日	議案
第1回	2023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 ● 各暗号技術調査ワーキンググループの活動計画(案)の審議 ● 軽量暗号ガイドライン更新のための調査・評価に関する審議
第2回	2024月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 ● 各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、各ガイドライン(案)に関する審議 ● 軽量暗号ガイドライン更新のための調査・評価結果に関する審議

以上

資料７

2023 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） 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・2022 年度に引き続いて暗号鍵管理ガイダンス WG を設置し、2022 年度発行版では記載を見送った部分の拡充を行う。2022 年度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

2022 年度の活用委員会での議論を踏まえて、暗号利活用に向けた有用なガイダンステーマを一つ又は二つ選定し、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。

[2023 年度完成を目指すガイドライン／ガイダンス]

- TLS 暗号設定ガイドラインのアップデート

[おおむね 2024 年度完成を目指すガイドライン／ガイダンスの候補]

同種のガイドライン／ガイダンスを作成している団体／組織などとの連携（リエゾン、joint WG）も含めて体制を検討し、新たなガイドライン／ガイダンスを作成する（以下は、候補例）。

- クラウドにおける鍵管理ガイダンス（例えば、日本クラウドセキュリティアライアンス（CSA））
- 暗号化消去ガイダンス（例えば、データ適正消去実行証明協議会（ADEC））
- 認証についてのガイダンス（例えば、FIDO アライアンス、OpenID ファウンデーション）

3. 活動スケジュール

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。

回	開催日	議案（予定）
第1回	2023年7月初旬	■ 委員長互選 ■ 2023年度活用委員会活動計画の確認 ■ 暗号鍵管理ガイダンスWG活動計画の審議 ■ 「TLS暗号設定ガイドライン」についての見直し検討 ■ 「暗号利活用のための新たなガイダンス」についての検討
第2回	2022年3月上旬	■ 「TLS暗号設定ガイドライン」についての見直しとりまとめ ■ 暗号鍵管理ガイダンスWG報告 ■ 「暗号利活用のための新たなガイダンス」についての中間とりまとめ ■ 2023年度暗号技術活用委員会活動報告案について

以上

暗号技術検討会
２０２２年度　報告書（案）

２０２３年３月

１．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においても、これまで取り組んできた暗号アルゴリズムのセキュリティ確保を引き続き推進することに加えて、暗号アルゴリズムを利用したプロトコルのセキュリティ確保のための活動拡大や、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貢献が求められている。

2022年度の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された暗号技術調査WG（耐量子計算機暗号）及び暗号技術調査WG（高機能暗号）において、それぞれ耐量子計算機暗号及び高機能暗号に関するガイドラインを作成した。また、軽量暗号ガイドラインについては、ガイドライン更新のための基となる調査のため、軽量暗号技術に対する安全性評価及び実装性能評価を実施し、標準化動向についても併せて調査した。また、暗号技術調査WG（耐量子計算機暗号）では、「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、CRYPTREC暗号リストの改定に向けた利用実績に関する評価、暗号利活用のために作成すべきガイダンス候補の検討を行った。また、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、「暗号鍵管理ガイダンス」の構成を見直しつつ、2022年度版の作成を行った。なお、「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」は2022年度開催しなかった。これらの2022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 2022」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。

2023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。

暗号技術検討会において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月に改定が行われ、CRYPTREC暗号リストとして発表され、2023年4月に再改定を行っている。その後においても、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、2021年9月に発足したデジタル庁、総務省及び経済産業省は、継続的に暗号技術検討会を開催している。

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：松本勉横浜国立大学教授）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。

2022年度のCRYPTRECにおいては、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）及び暗号技術調査WG（高機能暗号）において、それぞれ耐量子計算機暗号及び高機能暗号に関するガイドラインを作成した。また、軽量暗号ガイドラインについては、ガイドライン更新のための基となる調査のため、NIST Lightweight Cryptography Projectのファイナリストを対象とした安全性評価及び実装性能評価を実施した。なお、安全性評価に関しては、ISO/IEC標準規格として29192シリーズで規格化された軽量メッセージ認証コードの1つであるTsudik's keymodeも追加対象とした。また、標準化動向についても併せて調査した。また、暗号技術調査WG（耐量子計算機暗号）では、「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、CRYPTREC暗号リストの改定に向けた利用実績に関する評価、暗号利活用のために作成すべきガイダンス候補の検討を行った。また、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、「暗号鍵管理ガイダンス」の構成を見直しつつ、2022年度版の作成を行った。

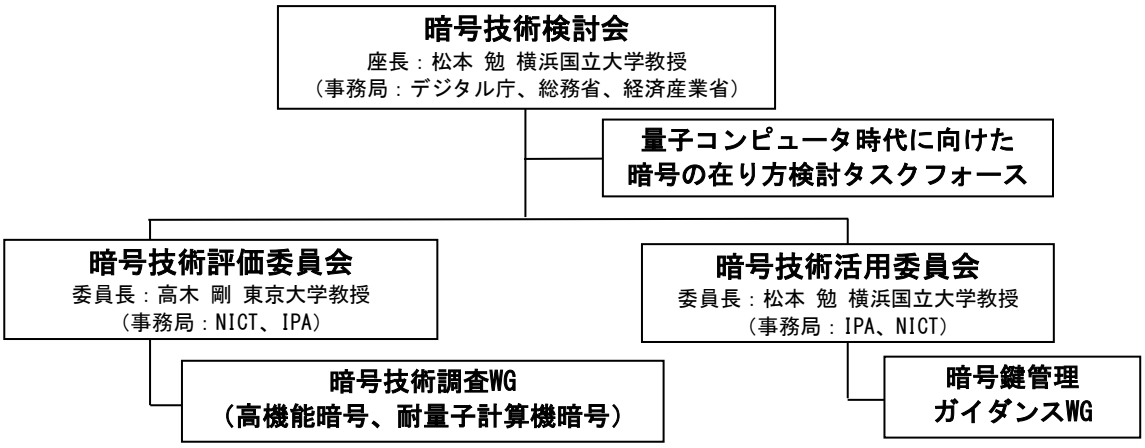


図2.2-1　2022年度CRYPTREC体制図

２．３．暗号技術検討会の開催実績

2022年度の暗号技術検討会は、CRYPTREC 暗号リストの改定に係る承認、意見募集の実施方法及びその後の対応に係る承認等を行うために第1回を開催し、暗号技術評価委員会、暗号技術活用委員会の活動報告、CRYPTREC暗号リストの改定に係る審議、暗号技術検討会2022年度報告書に係る承認等を行うために第2回を開催した。

【第 1 回】2023年2月27日（月）～2023年3月8日（水）（書類審議）
（主な議題）
・公募提案暗号の自主取下げ要望に対する取り扱いルールの策定並びに「ECDSA、 ECDH 及び SC2000」の取扱いについて
・CRYPTREC 暗号リストの改定について
・意見募集の実施方法及びその後の対応について
（概要）
・CRYPTREC暗号リストの改定について、原案のとおり承認された。
・意見募集の実施方法及びその後の対応について、原案のとおり承認された。

【第 2 回】2023年3月30日（木）9:00～11:00
（主な議題）
・2022年度暗号技術評価委員会 活動報告について【報告】
・CRYPTREC暗号技術ガイドライン（耐量子計算機暗号）及び CRYPTREC暗号技術ガイドライン（高機能暗号）について【承認】
・2022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について【承認】
・「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）」（案）に対する意見募集に寄せられたご意見に対するデジタル庁、総務省及び経済産業省の考え方（暗号技術検討会事務局案）並びにCRYPTREC暗号リストの改定版（暗号技術検討会事務局案）について【承認】
・2023年度暗号技術評価委員会活動計画（案）について【承認】
・2023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 2022年度 報告書（案）について【承認】
（概要）
・2022年度暗号技術評価委員会について事務局より活動報告を行った。
・CRYPTREC暗号技術ガイドライン（耐量子計算機暗号）及び CRYPTREC暗号技術ガイドライン（高機能暗号）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。
・2022年度暗号技術活用委員会について事務局より活動報告を行った。
・暗号鍵設定ガイダンス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。
・「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）」（案）に対する意見募集に寄せられたご意見に対するデジタル庁、総務省及び経済産業省の考え方（暗号

技術検討会事務局案）並びにCRYPTREC暗号リストの改定版（暗号技術検討会事務局案）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。

・2023年度暗号技術評価委員会活動計画（案）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。

・2023年度暗号技術活用委員会活動計画（案）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。

・暗号技術検討会 2022年度 報告書（案）について事務局より説明が行われ、議論結果を追記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ホームページへの公表
- 推奨候補暗号リストへの新規暗号（事務局選出）の追加
- 新世代暗号に係る調査

また、次期CRYPTREC暗号リストとは別文書として、耐量子計算機暗号、高機能暗号、及び、軽量暗号に関するガイドラインを作成する。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を作成するため、2021-2022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。当該ガイドラインを作成した。
- 高機能暗号に関するガイドラインを作成するため、2021-2022年度に、高機能暗号に関するワーキンググループを設置し、当該ガイドラインを作成した。
- 軽量暗号に関するガイドラインについては、2016年度に作成した「CRYPTREC暗号技術ガイドライン（軽量暗号)」の更新のため、2022年度は、掲載されている暗号方式に関わる安全性解析について、2017年度以降の技術動向調査を行う。2023年度中を目途に現ガイドラインを更新する。

これらの課題について2022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 2022（暗号技術評価委員会報告）に掲載する。

３．１．３．自主取下げに係る電子メールによる審議と結果

ECDSA、ECDH及びSC2000の応募暗号について取り下げの申請があったため、暗号技術評価委員会として以下の対応を行った。

表2：取り下げへの対応

	理由
ECDSA及びECDH	取り扱いを応募暗号技術からCRYPTRECが選出した暗号技術に変更し、現状通り、電子政府推奨暗号リストに記載しておくことは妥当であると判断す

	る。仕様書の参照先についても変更無しとする。
SC2000	応募社の判断を尊重し、取り下げを認める。推奨候補暗号リストから当該暗号技術を削除することは妥当であると判断する。

３．１．４．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ても安全性を保てると期待される暗号（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。

そして、2021年度、2022年度の2年間で耐量子計算機暗号ガイドラインを作成した。

2021年度のPQC WGの活動ではガイドライン作成の準備として、耐量子計算機暗号に関する研究動向調査を行い、ガイドライン執筆方針を以下のように定めた。

- 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分類とし、さらに耐量子計算機暗号の活用方法をガイドラインのみに記載する。これらの項目に関する情報を調査した。
- ガイドライン及び調査報告書の目次案
  i. 導入
  ii. PQC の活用方法（ガイドラインにのみ記載）
  iii. 格子に基づく暗号技術
  iv. 符号に基づく暗号技術
  v. 多変数多項式に基づく暗号技術
  vi. 同種写像に基づく暗号技術
  vii. ハッシュ関数に基づく署名技術

  - iii 章以降の構成（A 章の場合）
    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. 主要な暗号方式
    A.3.1. 暗号方式1（例：CRYSTALS-KYBER, Classic McEliece）
    A.3.2. 暗号方式2
    A.3.3. 暗号方式3
    ・・・
    A.4. まとめ

そして、2022年度暗号技術評価委員会において、2022年度のPQC WGの活動として下記2点について実施する活動計画が承認された。

- 耐量子計算機暗号のガイドラインの執筆
- 「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

それらの成果（３．１．４．１～３．１．４．２節）は2022年度第2回暗号技術評価委員会にて報告され、了承された。

３．１．４．１．耐量子計算機暗号に関するガイドラインの作成方針

耐量子計算機暗号ガイドライン及び調査報告書

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し、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する。ただし、暗号理論に精通していない利用者のために、耐量子計算機暗号の活用方法を耐量子計算機暗号ガイドラインでは記載し、調査報告書には記載しない。

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。

ガイドラインの章立て

1　はじめに
　1.1　耐量子計算機暗号(PQC) の必要性について
　1.2　PQC の研究及び標準化等に関する動向
　1.3　本調査における代表的なPQC の5 種類の分類を調査対象として選択した理由
2　PQC の活用方法
　2.1　暗号の利用形態
　2.2　各利用形態における課題
　2.3　各利用形態における対策
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　代表的な格子に基づく暗号方式
　3.3　格子に基づく主要な暗号方式: CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON
　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

３．１．４．２．「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。2022年度において、対応方針の説明文をより一般の読者に読みやすくなるよう、以下のとおり修正した。

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2022年6月・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3.2-1及び図3.2-2）。

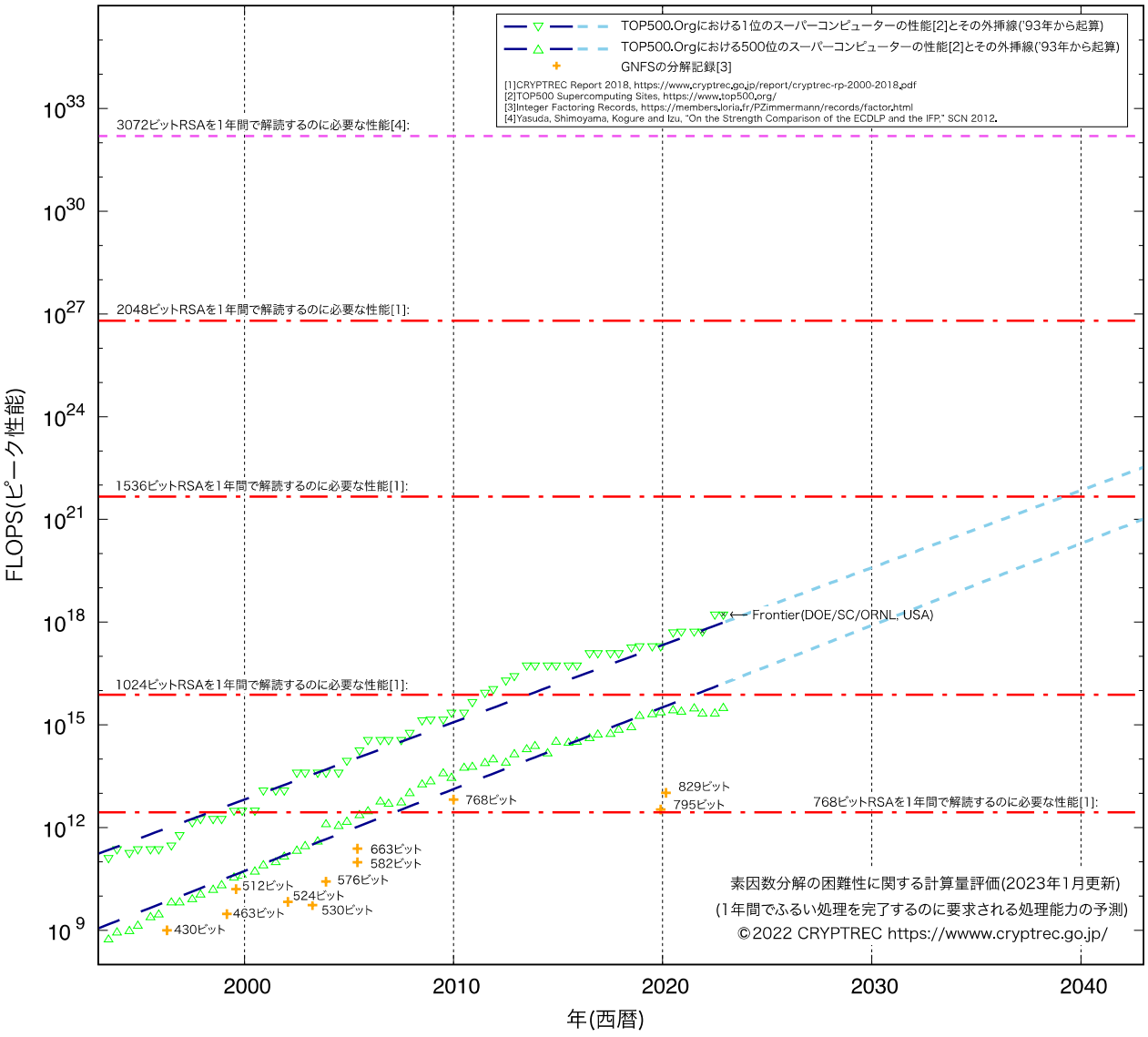


図3.2-1：素因数分解の困難性に関する計算量評価（2023年1月更新）[1]

[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。

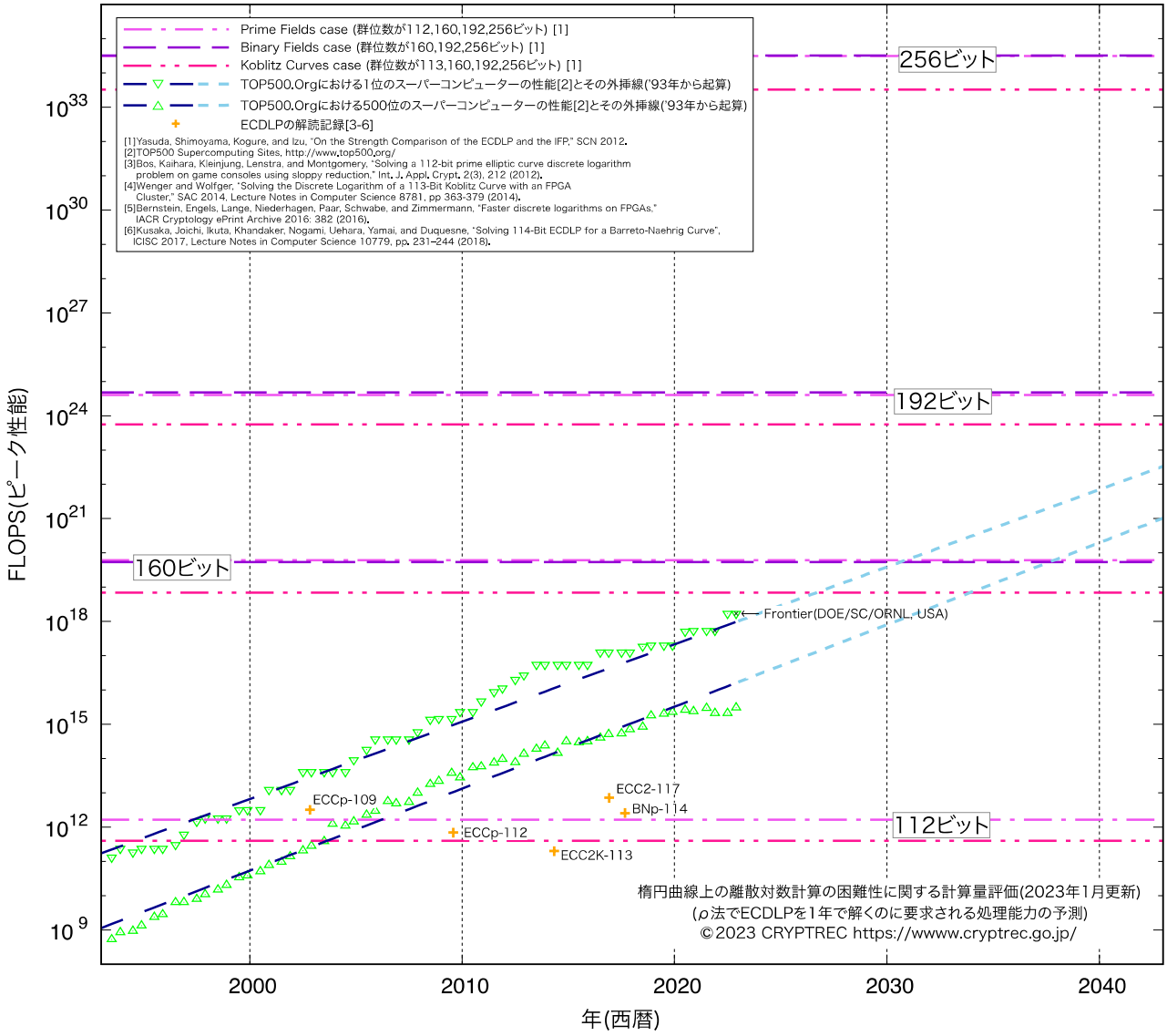


図3. 2-2：楕円曲線上の離散対数計算の困難性に関する計算量評価（2023年1月更新）[2]

[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。

３．１．５．暗号技術調査ワーキンググループ（高機能暗号）

公開鍵暗号は、アプリケーションが多様となりその活用が広まっている。その中で、従来の公開鍵暗号よりも機能が向上した高機能暗号を利用してアプリケーションに適用することが有効と考えられている。そこで、2020年度第2回暗号技術検討会において高機能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（高機能暗号）（以下「高機能暗号WG」という）を設置することが承認された。

そして、2021年度、2022年度の2年間で高機能暗号ガイドラインを作成した。

2021年度の高機能暗号WGの活動ではガイドライン作成の準備として、高機能暗号のスコープの明確化、技術に関する調査、アプリケーションに関する調査を行い、ガイドライン執筆方針を以下のように定めた。

- 高機能暗号のスコープ
  高機能暗号を「従来の暗号技術に対して、機能が追加・向上されるなどの優位性を主張する暗号、および、従来の暗号技術では困難であった事象を解決できるなどの新規機能を有することを主張する暗号技術」とした。
- 高機能暗号技術に関する現状調査
  ガイドラインに記載する高機能暗号を 3 分類､17 項目とし､それぞれの項目について“技術”、“活用事例”、“標準化”を調査した。
- 高機能暗号のアプリケーションに関する調査
  日本電気株式会社、三菱電機株式会社に対して 2022 年度にヒアリングを実施することとし、日本電気株式会社にはプライバシ保護、三菱電機株式会社についてはＤＢ関連のデータ保護のアプリケーションの事例を紹介していただくこととした。

そして、2022年度暗号技術評価委員会において、2022年度の高機能暗号WGの活動として下記2点について実施する活動計画が承認された。

- 2021 年度に作成した目次案に沿ったガイドラインの執筆
- 高機能暗号のアプリケーションに関するヒアリング調査および調査内容のガイドラインへの反映

それらの成果（３．１．５．１～３．１．５．２節）は2022年度第2回暗号技術評価委員会にて報告され、了承された。

３．１．５．１．2021年度に作成した目次案に沿ったガイドラインの執筆

ガイドラインの作成

高機能暗号ガイドラインは、高機能暗号を導入することを考えている技術開発者や、コンソーシアム・標準化団体に関与する技術者などを読者として想定し、暗号理論に精通していない方々を対象として執筆した。

ガイドラインに記載する暗号方式の選定基準及び候補について

主要な高機能暗号方式として、対象となる暗号方式は暗号技術調査ワーキンググループ（高機能暗号）によって承認されたものである。

ガイドラインの章立て

３．１．５．２．高機能暗号のアプリケーションに関するヒアリング調査および調査内容のガイドラインへの反映

2021年度の高機能暗号WGにおいて、高機能暗号によって既存技術より効率的になる分野、既存技術でカバーできていない分野などで、高機能暗号の活用が期待される分野を整理する。この活動の一環として、より深くアプリケーション、応用例を知るためにエンドユーザのヒアリングを検討した。そして、2022年度第1回、第2回の高機能暗号WGにおいて、以下のヒアリングを実施した。

① 日本電気株式会社：秘密分散を利用した医療データ活用

② 三菱電機株式会社：検索可能暗号＆属性ベース暗号

ヒアリングは、発表、質疑形式で行った。

このヒアリング内容は、本ガイドラインの“２．４．４章活用事例から見た高機能暗号の利用方法”に掲載したが、ヒアリング先である企業、団体、個人の宣伝とはならないように、できるだけ、企業、団体、個人名などを削除できるようにし、ヒアリング先に了解を得た。

３．１．６．「CRYPTREC暗号技術ガイドライン（軽量暗号）」更新に関わる活動

３．１．６．１．背景

2019年度に設置された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にて、「CRYPTRECにおいて、軽量暗号はCRYPTREC暗号リストには組み込まず、別途ガイドラインという形で取り扱う」ことが決定され、2020年度第2回暗号技術検討会にて、2016年度に作成した「CRYPTREC暗号技術ガイドライン（軽量暗号）」（以下、「2016年度版ガイドライン」という）を2023年度中を目処に更新することが承認された。2021年度第2回暗号技術評価委員会においてその更新方針が承認された。

当該更新方針に従い、今年度は、NIST軽量暗号プロジェクト（NIST Lightweight Cryptography Project ． 以下、「NIST LWC」という）のファイナリスト10方式（ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJAMBU, Xoodyak）を対象とした安全性評価及び実装性能評価を外部評価により実施した。なお、安全性評価に関しては、これら10方式に加え、ISO/IEC標準規格として 29192シリーズで規格化されている軽量メッセージ認証コードの1つであるTsudik's keymodeも対象としていた。また、軽量暗号に関わる NIST公開文書やISO/IECなどの標準化動向に関わる調査を外部評価により実施した。その外部評価実施内容を報告する。

なお、2023年2月7日に、NIST LWC 最終選考結果が発表され ASCON が選ばれた。

３．１．６．２．実施概要

安全性評価、実装性能評価、標準化動向調査、それぞれについて外部評価により以下のとおり実施した。

- 安全性評価：NIST LWCファイナリストに選定された10方式とISO/IEC標準規格として承認されたTsudik's keymode の安全性に関する調査及び評価を実施した。
- 実装性能評価：NIST LWCファイナリストに選定された10方式の実装性能（ハードウェア及びソフトウェア）に関する調査及び評価を実施した。
- 標準化動向調査：軽量暗号を取り巻く標準化動向(CAESAR プロジェクト、ISO/IECの軽量暗号関連カテゴリ、NIST LWCなど)の調査を実施した。

３．１．６．３．調査結果概要

[安全性評価結果概要]

NIST LWCファイナリストに選定された10方式とISO/IEC標準として規格化されたTsudik's keymodeの安全性評価について、2022年9月現在における調査結果を次のとおり表にまとめた。

安全性を脅かす攻撃が存在しない方式	特定の場合を除き、 安全性を脅かす方式が存在しない方式
ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, Xoodyak	TinyJAMBU, Tsudik's keymode

TinyJAMBUでは、関連鍵設定の場合に現実的な計算量での偽造攻撃が実行可能である。本攻撃が成立するような関連鍵の使用を避けることでTinyJAMBUの安全性を確保できることを確認した。

Tsudik's keymodeでは、使用するハッシュ関数が伸長攻撃と呼ばれる攻撃を許す場合に偽造攻撃が実行可能となるという既知の脆弱性が存在する。伸長攻撃が実行不可能なハッシュ関数を使用することでTsudik's keymodeの安全性を確保できることを確認した。

[実装性能評価結果概要]

それぞれの方式に関する現時点での実装性能を確認した。

- ハードウェア実装：回路面積とスループット性能に着目して評価した。例えば、Xilinx 社 の Artix-7 上では、TinyJAMBU の回路面積が小さいこと、SPARKLE は比較的面積コストが大きくなることなどを確認した。
- ソフトウェア実装：レイテンシ、コードサイズ、RAMサイズなどに着目し評価した。例えば、低リソースプラットフォーム (Arm Cortex-M0)上では、Elephant が最もレイテンシが高く、TinyJAMBU、Xoodyak、ASCON、SPARKLE が低レイテンシであることが分かった。コートサイズは、ASCON が最も大きく、他の候補暗号方式には大きな差は見られなかった。RAM の使用量は、コンパイル時の静的なメモリサイズのレポートから、どのアルゴリズムも約 1kByte程度であった。

[標準化動向調査結果概要]

CAESAR プロジェクト、ISO/IECの軽量暗号関連カテゴリ、NIST LWC などの状況を調査し、まとめた。調査結果により、軽量暗号をとりまく現状を確認した。例えば、2016年度版ガイドラインに掲載されている SIMON と SPECK については、ISO/IECの軽量暗号のカテゴリで議論されていたが、結果として軽量暗号としてはISO/IEC標準規格として承認されず、自動認識・データキャプチャ技術に関する仕様で利用可能な軽量暗号方式として規格化されていることを確認した。

また、評価指標に関して、安全性評価については、提案方式の設計根拠が十分に提示されない場合に第3者による評価が十分に行えないと判断され、評価対象から外されるなどの事例があった。実装性能評価については、従来論文ごとに異なる環境や測定シナリオで示されることが多くあったが、近年は AES-GCM や SHA-256 など広く世界で利用されているアルゴリズムとの比較などにより統一的な測定フレームワークを用いて実施することが一般化されてきていることが分かった。

３．１．６．４．外部評価報告書に対する暗号技術評価委員会の見解

実施した外部評価報告書は、今年度に目的としていた調査対象の暗号方式に対して、安全性・実装性能・標準化動向の調査として十分な内容を含んでいると考えられることから、本報告書を CRYPTREC の技術調査報告書とすることが了承された。

３．１．６．５．来年度の計画

2021 年度第 2 回暗号技術評価委員会において承認された更新方針に従い、暗号技術評価委員会事務局によりガイドラインの更新案を編集し、ドラフト版について外部有識者にガイドラインとして掲載内容の適切性や情報の過不足などについてレビュー頂き、完成版を 2023 年度暗号技術評価委員会で審議する予定である。

３．１．７．暗号技術評価委員会の開催実績

2021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3.2-5のとおりである。

表3.2-5　暗号技術評価委員会の開催状況

回	開催日	議案
第1回	2022年7月26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 ■ 暗号技術調査ワーキンググループ（耐量子計算機暗号）の活動計画案の審議 ■ 暗号技術調査ワーキンググループ（高機能暗号）の活動計画案の審議 ■ 外部評価（軽量暗号に関するガイドライン係る技術動向調査）実施についての審議

		■ 監視状況報告
第2回	2023月2月27日	■ 自主取下げに係る電子メールによる審議内容と結果の報告 ■ 暗号技術調査ワーキンググループ（耐量子計算機暗号）の活動内容の報告 ■ 暗号技術調査ワーキンググループ（高機能暗号）の活動内容の報告 ■ 軽量暗号ガイドラインに係る技術動向調査結果の報告 ■ 監視状況報告 ■ CRYPTREC Report 2022作成について ■ CRYPTRECシンポジウム開催について

３．２．暗号技術活用委員会

３．２．１．活動の概要

2022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 2022暗号技術活用委員会報告[3]を参照されたい。

(1) 利用実績に関する評価

2022年度はCRYPTREC暗号リストの改定が予定されており、その際、推奨候補暗号リストから電子政府推奨暗号リストへの昇格にあたって利用実績に基づいた選定が行われることが決まっている。

そこで、IPAが実施する「暗号アルゴリズムの利用実績に関する調査」による調査結果に基づき、2021年度に承認された利用実績による選定基準の下で利用実績に関する評価を行う。

(2) 暗号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を目的として、2021年度に取りまとめた作業の進め方に基づき、暗号鍵管理ガイダンスWGにて暗号鍵管理ガイダンスを作成する。

(3) 暗号利活用のために作成すべきガイダンス候補の検討

暗号利活用のために作成すべきガイダンス候補を検討し、今後の執筆に向けた準備を行う。

３．２．２．2022年度の活動内容

３．２．２．１．利用実績に関する評価

IPAが実施した暗号アルゴリズム利用実績調査の結果、及び2021年度に承認された利用実績に基づく選定基準（選定ルール）（下表）に基づき、現在の推奨候補暗号リストに掲載のアルゴリズムのうち、電子政府推奨暗号リスト掲載への推薦候補案について検討・選定し、暗号技術検討会に推薦した。

【検討方針】

IPAが実施した暗号アルゴリズム利用実績調査[4]では、現在の推奨候補暗号リストに掲載のアルゴリズムのうち、EdDSAのみアンケートによる利用実績調査の対象外であった。

このため、「EdDSA」以外の「推奨候補暗号リスト」に掲載の暗号アルゴリズムについては「利用実績調査（考慮項目①～⑥）」結果に基づいて判定し、「EdDSA」については「利用実態確認（考慮項目②～⑤）」結果に基づいて判定することとした。

---

[3] CRYPTREC Report 2022 暗号技術活用委員会報告, https://www.cryptrec.go.jp/promo_cmte.html

[4] IPA、「暗号アルゴリズムの利用実績に関する調査報告書」の公開、https://www.ipa.go.jp/security/fy24/reports/cryptrec/crypto-algorithm/crypt_usageper_report.html

表　利用実績に基づく選定基準（選定ルール）

考慮項目		選定目安
採用実績	以下のいずれかを満たす場合、昇格の検討対象に含める。なお、採用実績は、 ● 5 年ごとに実施予定の大規模アンケート調査による「**利用実績調査**」 ● 必要に応じて、事務局が（大規模アンケート調査によらずに）情報収集する「**利用実態確認**」 により確認するものとする。	
	① **利用実績調査**の結果、電子政府推奨暗号リストに掲載されている（同一カテゴリの）暗号技術の採用実績と遜色がないことが確認された場合	**電子政府推奨暗号リスト掲載の（同一カテゴリの）暗号技術の採用実績を同等以上の採用実績がある推奨候補暗号リスト掲載の暗号技術を昇格検討対象とする。**
	② **利用実績調査**又は**利用実態確認**の結果、**電子政府システムや重要インフラ等、日本の基幹システム**においてすでに利用されていることが確認された場合	**必要に応じて、利用実績調査に代わって、各府省庁等への照会を実施し、照会結果（クローズドな利用を含め）を基に昇格検討対象を選定する。**
	利用実績調査又は**利用実態確認**の結果、③～⑤のいずれかが確認された場合： ③ **利用者が多い主要な汎用製品群の複数**に搭載されるなど、明らかに採用が進展していると判断された場合 ④ **利用者が多い主要なオープンソースソフトウェアの複数**に搭載されるなど、明らかに採用が進展していると判断された場合 ⑤ **利用者が多い主要なサービスやプロトコルの複数**で利用されるなど、明らかに採用が進展していると判断された場合	**「複数」「利用者が多い（主要な）」というキーワードの両方を十分に満たし、明らかな採用促進が確認された場合には、必要に応じて、昇格検討対象とする。** **※「複数」の意味は、必要条件として「2個以上が必要」ということであって、「2個以上あればよい」という十分条件としての意味ではないことに留意**
標準化実績	以下を満たす場合、昇格の検討対象に含める。	
	⑥ **利用実績調査**の結果、電子政府推奨暗号リストに掲載されている（同一カテゴリの）暗号技術の採用実績と遜色がないことが確認された場合	**電子政府推奨暗号リスト掲載の（同一カテゴリの）暗号技術の採用実績を同等以上の採用実績がある推奨候補暗号リスト掲載の暗号技術は昇格検討対象とする。**

【電子政府推奨暗号リスト掲載への推薦候補案】

- **エンティティ認証、ハッシュ関数、署名を除いた技術分類について：**

技術分類		推薦候補	推薦しない候補	理由
公開鍵暗号	鍵共有	該当なし	PSEC-KEM	●他の鍵共有と比較して優位な利用実績があるとは認められない
共通鍵暗号	64ビットブロック暗号	該当なし	CIPHERUNICORN-E Hierocrypt-L1 MISTY1	●他の64ビットブロック暗号と比較して優位な利用実績があるとは認められない
	128ビットブロック暗号	該当なし	CIPHERUNICORN-A CLEFIA Hierocrypt-3 SC2000	●他の128ビットブロック暗号と比較して優位な利用実績があるとは認められない
	ストリーム暗号	該当なし	Enocoro-128v2 MUGI MULTI-S01	●他のストリーム暗号と比較して優位な利用実績があるとは認められない
認証暗号		ChaCha20-Poly1305	該当なし	●考慮項目①②④について、利用実績があると認められる
暗号利用モード	秘匿モード	XTS	該当なし	●考慮項目①②④について、他の秘匿モードと比較して利用実績があると認められる
メッセージ認証コード		該当なし	PC-MAC-AES	●他のメッセージ認証コードと比較して優位な利用実績があるとは認められない

- **エンティティ認証について：**

技術分類	推薦候補	推薦しない候補	理由
エンティティ認証	ISO/IEC 9798-4	該当なし	●考慮項目①②において、他のエンティティ認証と比較して利用実績があると認められる

- **ハッシュ関数について：**

技術分類	推薦候補	推薦しない候補	理由
ハッシュ関数	SHA-512/256	該当なし	●考慮項目④において、他のハ

	SHA3-256 SHA3-384 SHA3-512 SHAKE128 SHAKE256		ッシュ関数と比較して利用実績があると認められる

- **EdDSA について：**

技術分類		推薦候補	推薦しない候補	理由
公開鍵暗号	署名	EdDSA	該当なし	●考慮項目④において、他の署名と比較して利用実績があると認められる

３．２．２．２．自主取下げ申請への対応

富士通株式会社から取下げ申請があったSC2000について、審議の結果、暗号技術検討会でCRYPTREC暗号リストからの取下げルールを整備することを条件に、申請を了承することとした。

３．２．２．３．暗号鍵管理ガイダンスの作成

暗号鍵管理検討の初めとして、暗号鍵管理が必要なシステムの設計者向けに、暗号鍵管理の設計で明記する事項や考慮する点などを解説することを目的としたガイダンスである。本ガイダンスの位置づけと想定読者は以下の通りとする。

位置づけ

- 暗号鍵管理機能を持つシステム設計者のガイダンスを作成する。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）」を詳しく解説することを中心に作成する
- 暗号鍵管理で必要となる項目について、シンプルなモデルを例示し説明する
- シンプルなモデルを用いた説明においては、鍵管理における要求や思想が理解できるような記載を行う
- 暗号鍵管理における特に注意すべきリスクを説明する

想定読者

- 暗号鍵管理機能を持つシステム設計者

2022年度版暗号鍵ガイダンスの章構成は以下のとおりである。具体的には、本ガイダンスは「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として書かれている。

1. はじめに
   イントロダクションとして、暗号鍵管理の重要性、及び本ガイダンスの位置づけについて説明している。
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
   暗号鍵管理システム設計指針（基本編）での「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」における検討項目についての解説・考慮点を記載している。具体的には、CKMSにおいてどのように暗号鍵が管理されるかを対象にしており、暗号鍵の生成から廃棄までのライフサイクル全期間にわたって暗号鍵を管理するのに必要となる機能や運用方法を取り決める検討項目を取り扱っている。また、簡単なモデル（トイモデル）としてS/MIMEをモデルに取り上げ、記載例を示した。
3. 暗号アルゴリズムの選択
   暗号鍵管理システム設計指針（基本編）での「暗号アルゴリズムの選択」における検討項目についての解説・考慮点を記載している。具体的には、暗号アルゴリズムや鍵長を選択に関する重要なポイントの解説、特にCRYPTREC暗号リスト（電子政府推奨暗号リスト）、暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準や暗号鍵設定ガイダンスを参考に選択することを推奨している。また、Webブラウザをクライアントとするクライアントーサーバシステムをモデルにした記載例を示した。
4. 暗号アルゴリズム運用に必要な鍵情報の管理
   暗号鍵管理システム設計指針（基本編）での「暗号アルゴリズム運用に必要な鍵情報の管理」における検討項目についての解説・考慮点を記載している。ここでは、3章で決定した暗号アルゴリズムを運用するときに必要な鍵情報の管理の説明、具体的には管理すべき全ての鍵を明確にし、その鍵とメタデータの保護方法についての解説・考慮点を記載している。また、関係者がアクセス可能なWebサーバの鍵設定・管理をモデルにして記載例を示した。

３．２．２．４．暗号利活用のために作成すべきガイダンス候補の検討

2023年度以降に作成すべきガイドライン／ガイダンスの候補について、以下の視点を踏まえ、検討を行った。今回の議論を踏まえ、どのガイドライン／ガイダンスを作成するかを2023年度活動計画に反映することを決定した。

検討にあたっての視点

- どのようなガイドライン／ガイダンスが求められているか
- CRYPTREC がメインで作るのがよいか、それとも他の組織（IPA、業界団体など）がメインで／共同で作るのがよいか
- 暗号技術の切り口メインで有用なガイドライン／ガイダンスになるか（暗号以外の部分がメインになったりしないか）

候補に挙がったガイドライン／ガイダンスのテーマ

1	認証についてのガイダンス（特に二要素認証）
2	身元（本人）確認のためのガイダンス（例えばeKYC）
3	電子メールに関するガイドライン／ガイダンス
4	クラウドにおける鍵管理ガイダンス
5	組込機器の開発における、暗号プロトコル（例：認証プロトコル）のパラメータ選定基準
6	経営層も含めた人達を対象にした、暗号技術の啓発ドキュメント
7	暗号の使い方に関するガイドライン（ガイダンス）
8	PKIガイドライン（ガイダンス）
9	暗号化消去
10	DNSの暗号に関わるガイドライン（ガイダンス）
11	暗号資産
12	eシール
13	APIに関するガイドライン／ガイダンス
14	高機能暗号の標準化
15	耐量子計算機暗号のガイダンス
16	耐量子計算機暗号への移行に関するガイダンス
17	FIDOなどの普及促進を促すガイダンス
18	リモート署名などの普及促進を促すガイダンス
19	暗号化消去などの普及促進を促すガイダンス
20	TLS暗号設定ガイドラインのアップデート
21	運用ガイドラインやガイダンスに求められるニーズ／課題の整理

３．２．３．暗号技術活用委員会の開催状況

2022年度の暗号技術活用委員会での審議概要は表の通りである。

表　暗号技術活用委員会の開催状況

回	開催日	議案
メール	2022年7月7日 ～ 7月15日	● 2022 年度暗号鍵管理ガイダンス WG 活動計画について

第一回	2022年8月4日	● 2022 年度暗号技術活用委員会活動計画について ● 暗号アルゴリズム利用実績調査の中間報告について ● 2022 年度暗号鍵管理ガイダンス WG 活動計画について ● 暗号鍵管理ガイダンス WG 進捗報告について ● 運用ガイドライン／ガイダンス候補について
第二回	2022年12月20日	● 暗号アルゴリズム利用実績調査の最終報告について ● 電子政府推奨暗号リスト掲載への推薦候補案について ● 暗号鍵管理ガイダンス WG 進捗報告について ● 運用ガイドライン／ガイダンス候補について
第三回	2023年3月14日	● 暗号鍵管理ガイダンス WG 活動報告 ● 運用ガイドライン／ガイダンス候補について ● 2022 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、暗号アルゴリズムの安全性確保やその利活用に係る議論のみならず、鍵管理の安全な運用に向けた取組など、暗号をとりまく環境変化に応じた新たなニーズへの対応などに取り組むこととしている。2023年度もCRYPTREC暗号リストの改定を必要に応じて行うべく、その改定に向けた検討を進めていく。

暗号技術評価委員会においては、耐量子計算機暗号ガイドラインが完成したところであるが、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から、引き続き耐量子計算機暗号に関する最新動向を把握する必要がある。また、現在、軽量暗号に関するガイドラインの更新に向けた検討中であり、次年度完成させる予定である。暗号技術活用委員会においては、2022年度版暗号鍵管理ガイダンスを完成させたところであるが、「暗号鍵管理システム設計指針（基本編）」に記載がありながら今回解説・考慮点の記載を見送った部分の拡充を行う。また、TLS暗号設定ガイドラインのアップデートを実施するとともに、2022年度活用委員会での議論を踏まえ、暗号利活用に向けた新たな有用なガイダンス作成に着手する予定である。

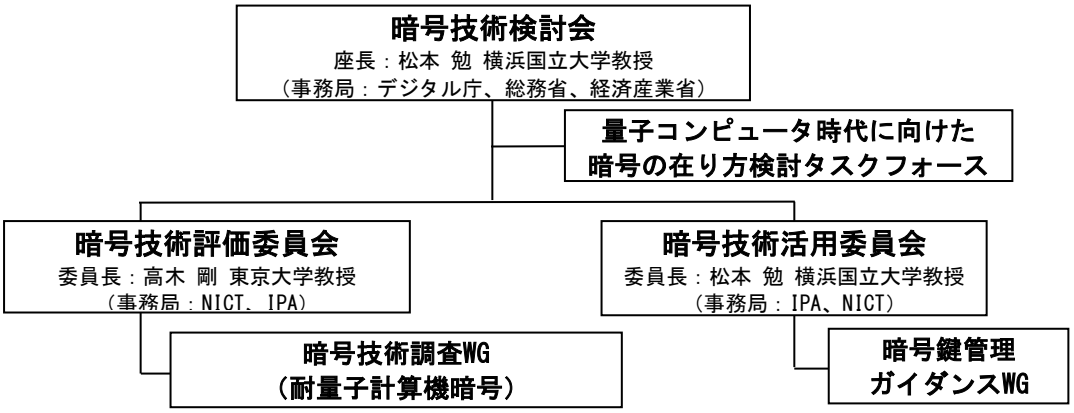


図4-1 2023年度CRYPTRECの体制図（予定）