資料１

2025年度 第１回 暗号技術検討会

令和８年３月25日(水) 9:00～11:00
経済産業省本館17階　第２特別会議室
（ ハ イ ブ リ ッ ド 開 催 ）

＜議事次第＞

１．開会
２．議事
(1) 2025年度暗号技術評価委員会 活動報告【報告】
(2) CRYPTREC暗号リストにおける仕様書参照先の変更【承認】
(3) 外部評価：耐量子計算機暗号ML-KEMの安全性・実装性能に関する評価及び調査（案）【承認】
(4) 外部評価：耐量子計算機暗号への移行に関する技術動向調査（案）【承認】
(5) 2025年度暗号技術活用委員会 活動報告【報告】
(6) 政府機関等における耐量子計算機暗号（PQC）への移行【報告】
(7) CRYPTREC暗号リストの改定 ～耐量子計算機暗号（PQC）対応～【審議】
(8) 「耐量子計算機暗号（PQC）タスクフォース」の設置【審議】
(9) 2026年度暗号技術評価委員会 活動計画(案)【承認】
(10) 2026年度暗号技術活用委員会 活動計画(案)【承認】
(11) 暗号技術検討会 2025年度 報告書（案）【承認】
(12) 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　2025年度 暗号技術評価委員会 活動報告
資料３－２　CRYPTREC暗号リストにおける仕様書参照先の変更（案）
資料３－３　耐量子計算機暗号ML-KEMの安全性・実装性能に関する評価及び調査（案）
資料３－４　耐量子計算機暗号への移行に関する技術動向調査（案）
資料４　　2025年度 暗号技術活用委員会 活動報告
資料５　　政府機関等における耐量子計算機暗号（PQC）への移行について
資料６－１　耐量子計算機暗号（PQC）に対応したCRYPTREC暗号リストの在り方について
資料６－２　CRYPTREC暗号リスト（案）
資料７　　「耐量子計算機暗号（PQC）リスト検討タスクフォース」開催要綱（案）
資料８　　2026年度 暗号技術評価委員会 活動計画（案）
資料９　　2026年度 暗号技術活用委員会 活動計画（案）
資料10　　暗号技術検討会 2025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

2026.3.25現在

構成員

阿部　正幸	日本電信電話株式会社　社会情報研究所　フェロー
石井　義則	一般社団法人情報通信ネットワーク産業協会　常務理事
上原哲太郎	立命館大学　情報理工学部　情報理工学科　教授
國廣　　昇	筑波大学　システム情報系　教授
黒田　真弓	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
島岡　政基	セコム株式会社　IS研究所　デジタルプラットフォームディビジョン 主幹研究員
高木　　剛	東京大学　大学院情報理工学系研究科　数理情報学専攻　教授
田村　裕子	日本銀行　金融研究所　情報技術研究センター　企画役
本間　尚文	東北大学　電気通信研究所　教授
松井　　充	国立研究開発法人情報通信研究機構　主席研究員
松浦　幹太	東京大学　生産技術研究所　教授
松本　　勉	国立研究開発法人産業技術総合研究所 フェロー 横浜国立大学　先端科学高等研究院　上席特別教授
松本　　泰	特定非営利活動法人日本ネットワークセキュリティ協会　フェロー
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部門　研究グループ長
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部門　研究部門長

（五十音順、敬称略）

オブザーバ

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

別紙２

暗号技術検討会の公開について

１　会議の公開について

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

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

２　検討会の資料の公開について

（１）検討会の資料については、原則公開とする。

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

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

３　議事概要の公開について

（１）議事概要については、原則公開とする。

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

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

資料３－１

2025 年度暗号技術評価委員会活動報告（案）

１．活動目的

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

２．暗号技術評価委員会の構成（敬称略）

委員長	高木　剛	（東京大学）
委員	青木　和麻呂	（文教大学）
委員	伊藤　忠彦	（セコム株式会社）
委員	岩田　哲	（名古屋大学）
委員	上原　哲太郎	（立命館大学）
委員	大東　俊博	（東海大学）
委員	國廣　昇	（筑波大学）
委員	四方　順司	（横浜国立大学）
委員	千田　浩司	（群馬大学）
委員	花岡　悟一郎	（産業技術総合研究所）
委員	藤崎　英一郎	（北陸先端科学技術大学院大学）
委員	本間　尚文	（東北大学）
委員	松本　勉	（産業技術総合研究所・横浜国立大学）
委員	山村　明弘	（秋田大学）

３．活動概要

2025 年度暗号技術評価委員会活動計画に沿って以下の内容を行った。

（１）暗号技術の安全性及び実装に係る監視及び評価

以下の通り、暗号技術の安全性及び実装に係る監視・評価を実施した。

①　CRYPTREC暗号リストの監視

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

- 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。今年度実施の監視状況報告の詳細は、CRYPTREC Report 2025で報告する。
- ECDHの仕様書参照先が更新されていることを確認した。このため、仕様書の更新箇所を確認・調査し、参照先の変更に問題ないことを確認した上で、当該参照先を更新した。

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

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

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削除に至るような事態は生じていない。

③ CRYPTREC注意喚起レポートの発行

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

CRYPTREC注意喚起レポートの発行に至るような事態は生じていない。

④ 推奨候補暗号リストへの新規暗号（事務局選出）の追加に係る検討

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

CRYPTREC暗号リストへの掲載に向けたPQCの技術的検討として、耐量子計算機暗号ML-KEMの安全性・実装性能に関する調査・評価を外部評価により実施し、本報告書の内容を踏まえ、暗号技術評価委員会としての見解をまとめた。

⑤ 新技術等に関する調査及び評価

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

（ア）NISTのPQC標準化において第4ラウンドが進行中であり、2024年度の暗号技術調査ワーキンググループ（耐量子計算機暗号）の委員からも、2025年度以降もワーキンググループを設置する意見が出ていることから、引き続き、暗号技術調査ワーキンググループ（耐量子計算機暗号）を設置して、耐量子計算機暗号に関する最新動向を把握する。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。

● 耐量子計算機暗号に関する最新動向を2025年度から2026年度にかけて調査・把握し、ガイドライン及び調査報告書を作成することが確認された。 ● ガイドライン及び調査報告書の執筆に関し、2026年9月30日までの情報を可能な限り調査して掲載するなど、基本的な方針について合意された。 ● 「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」の予測図を更新し、2025年度においても大きな進展がないことを確認した。

（イ）CRYPTREC暗号リスト掲載に向けたPQCの技術的検討に資するため、以下の活動を行なった。

- 耐量子計算機暗号ML-KEMの安全性・実装性能に関する調査及び評価を外部評価により実施した。
- 耐量子計算機暗号ML-DSAの安全性・実装性能に関する調査及び評価を外部評価により開始した。

（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。

PQCへの移行方針の技術的検討に資するため、PQCへの移行に関する技術動向調査を外部評価により実施した。

４．スケジュール及び各委員会での活動内容

（１）第 1 回暗号技術評価委員会（2025 年 7 月 1 日：オンライン）の活動内容

① 2025 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）活動計画（案）について審議し、原案のまま承認された。

② 耐量子計算機暗号への対応方針と 2025 年度外部評価（案）について審議し、原案のまま承認された。

③ CRYPTREC 暗号リストにおける仕様書参照先の変更（案）について審議し、ECDH での使用が許可された MAC の一種である KMAC に関する表現の一部修正に関し、メール審議で対応することが確認された。

④ 監視状況に関する報告が行われ、CRYPTREC 暗号リストに掲載されている暗号技術の安全性に問題がないことが確認された。

（２）メール審議（2025 年 7 月 2 日～16 日）

- ECDH の仕様書参照先の変更等について審議し、KMAC に関連する表現の一部修正に関して承認された。

（３）メール審議（2025 年 12 月 2 日～19 日）

- 外部評価スケジュールの変更と ML-DSA 外部評価の実施（案）について審議し、原案のまま承認された。

（４）第 2 回暗号技術評価委員会（2026 年 3 月 3 日：オンライン）の活動内容

① 2025 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）の活動報告が行われた。

② 外部スケジュールの再変更と SLH-DSA 外部評価の実施（案）について審議し、原案のまま承認された。

③ 2025 年度外部評価「耐量子計算機暗号 ML-KEM の安全性・実装性能に関する調査及

び評価」について審議し、暗号技術評価委員会としての見解については一部表現を修正することを前提に承認された。あわせて、CRYPTREC の技術調査報告書として公開することについても承認された。

④ 2025 年度外部評価「耐量子計算機暗号への移行に関する技術動向調査」について審議し、外部評価報告書については軽微な修正を行うことを前提に、CRYPTREC の技術調査報告書として公開することが承認された。

⑤ 監視状況に関する報告が行われ、CRYPTREC 暗号リストに掲載されている暗号技術の安全性に問題がないことが確認された。

⑥ 2025 年度暗号技術評価委員会活動報告案の確認が行われた。

⑦ 2026 年度暗号技術評価委員会活動計画案の確認が行われた。

⑧ CRYPTREC Report 2025 の目次案の確認が行われた。

以上

資料３－２

CRYPTREC 暗号リストにおける仕様書参照先の変更

１．経緯と目的

2024 年度第 1 回暗号技術評価委員会にて、CRYPTREC 暗号の仕様書一覧[1]に掲載されている暗号方式のうち、Web ページが存在しなくなっている、またはアップデートされている事例が 3 件あることを報告した。それらのうち、公開鍵暗号（署名）ECDSA および 128 ビットブロック暗号 AES に関して仕様書参照先の変更が承認されたものの、公開鍵暗号（鍵共有）ECDH の新しい仕様書において変更項目が非常に多いことから、時間をかけてその変更項目の調査を行い、2025 年度第 1 回暗号技術評価委員会にて改めて報告することが了承された。

そこで、公開鍵暗号（鍵共有）ECDH の新しい仕様書における主な変更項目について報告するとともに、ECDH の仕様書参照先の変更および CRYPTREC 暗号リストにおける注釈の追加に関して事務局案を提示する。

２．現在の仕様書参照先

以下の表のとおり。

技術分類		暗号名称	仕様書
公開鍵暗号	鍵共有	ECDH	SEC 1: Elliptic Curve Cryptography (September 20, 2000 Version 1.0) https://www.secg.org/SEC1-Ver-1.0.pdf または NIST SP 800-56A Revision 2 (May 2013) において、C(2e, 0s, ECC CDH)として規定されたもの https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf

３．更新された仕様書参照先

2 つある仕様書参照先のうち、NIST SP 800-56A Revision 2 (May 2013) が以下のとおり更新されていることを確認した。

NIST SP 800-56A Revision 3 (April 2018)

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf

４．旧バージョンとの主な差異（抜粋）

（１）許可されている暗号強度

旧：80, 112, 128, 192, 256 ビット

新：112, 128, 192, 256 ビット　（80 ビット強度が削除）

[1] https://www.cryptrec.go.jp/method.html

（２）許可されている MAC の種類

旧：HMAC(use of approved hash function), CMAC(use of approved block cipher)

新：HMAC(use of approved hash function), AES-CMAC, KMAC[2]

（CMAC が AES に限定、KMAC が追加）

５．仕様書参照先の変更と CRYPTREC 暗号リストにおける注釈の追加

（１）仕様書参照先の変更【実施済】

仕様書参照先を旧バージョンから新バージョンに変更する。また、KMAC が CRYPTREC 暗号リスト掲載の MAC ではないため、「使用する MAC は HMAC または AES-CMAC に限る。」という注釈を追記する。変更後の仕様書参照先は、次のとおり。変更箇所を赤字で示す。

技術分類		暗号名称	仕様書
公開鍵暗号	鍵共有	ECDH	SEC 1: Elliptic Curve Cryptography (September 20, 2000 Version 1.0) https://www.secg.org/SEC1-Ver-1.0.pdf または NIST SP 800-56A Revision 3 (April 2018) において、C(2e, 0s, ECC CDH)として規定されたもの(*4) https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf

(*4) 使用する MAC は HMAC または AES-CMAC に限る。

（２）CRYPTREC 暗号リストにおける注釈の追加（案）【審議事項】

KMAC が CRYPTREC 暗号リスト掲載の MAC ではないため、「使用する MAC は HMAC または AES-CMAC に限る。」という注釈を追記する。変更後の CRYPTREC 暗号リスト（抜粋）は、次のとおり。変更箇所を赤字で示す。

技術分類		暗号名称
公開鍵暗号	鍵共有	ECDH (注 20)

（注 20）使用する MAC は HMAC または AES-CMAC に限る。

以上

[2] NIST SP 800-185: SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash, and ParallelHash. (https://csrc.nist.gov/pubs/sp/800/185/final)

資料３－３

外部評価：ML-KEM の安全性・実装性能に関する評価及び調査（案）

１　背景

（１）2022 年 7 月 5 日に NIST から耐量子計算機暗号（PQC）の標準化方式として、公開鍵暗号 1 方式と電子署名 3 方式が発表された。これら 4 方式のうち、格子に基づく公開鍵暗号方式 ML-KEM は FIPS 203 として、格子に基づく署名方式 ML-DSA は FIPS 204 として、ハッシュ関数に基づく署名方式 SLH-DSA は FIPS 205 として、それぞれ 2024 年 8 月 13 日に標準化された。

（２）2024 年度暗号技術検討会において、PQC への対応について議論が行われ、CRYPTREC 暗号リストへの掲載に向けた PQC の技術的検討と、PQC への移行方針の検討を両輪として並行に進めていくべきであるとの合意が得られた。これに伴い、ML-KEM、ML-DSA、SLH-DSA の安全性・実装性能の評価を先行して実施することで合意された。

（３）2025 年度第 1 回暗号技術評価委員会において、ML-KEM の安全性・実装性能に関する評価及び調査を外部評価により実施することが承認された。

２　実施概要

２．１　ML-KEM の安全性に関する調査及び評価

ML-KEM の安全性評価に関する 1 件目の外部評価を安田雅哉 様（立教大学）に依頼した。選出理由と依頼内容は次のとおり。

（１）選出理由

ML-KEM の安全性を支える数学問題とその数学問題の求解アルゴリズムにおける計算量見積に関して広い知見をお持ちである。当該分野に関する数多くの実績があるとともに、CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）の第 3 章「格子に基づく暗号技術」の執筆における主担当者としての実績がある。

（２）依頼内容

耐量子計算機暗号 ML-KEM の安全性について、公開されている解析手法や評価結果の有無を調査し、存在する場合はその影響範囲などについてまとめるなど、安全性評価を実施した上で報告書を作成する。

２．２　ML-KEM の安全性・実装性能に関する調査及び評価

ML-KEM の安全性評価に関する 2 件目の外部評価に加え、実装性能評価に関する外部評価を勝又秀一 様（PQShield）[1]に依頼した。選出理由と依頼内容は次のとおり。

---

[1] PQShield 所属の他研究者との共同執筆であり、勝又様には主執筆者かつ調整窓口として依頼した。

（１）選出理由

勝又様は格子暗号の安全性評価やその応用先である暗号プロトコルの安全性評価に関する広い知見をお持ちであり、当該分野に関する数多くの実績がある。また、共同執筆者に関しては、実装性能評価に関する広い知見をお持ちであり、当該分野に関する数多くの実績がある。

（２）依頼内容

耐量子計算機暗号 ML-KEM の安全性・実装性能について、公開されている解析手法や評価結果の有無を調査し、存在する場合はその影響範囲などについてまとめるなど、安全性と実装性能を評価した上で報告書を作成する。

３　外部評価報告書の概要【報告事項】

３．１　ML-KEM の安全性に関する調査及び評価

（１）安全性証明（別紙 1 第 3 章、別紙 2 第 4 章）

- 古典ランダムオラクルモデルにおいて、使用される 2 つのハッシュ関数がランダムオラクルと仮定した場合、タイトな IND-CCA2 安全性を持つことが確認された。
- 量子ランダムオラクルモデルにおいて、IND-CCA2 安全性はノンタイトであり漸近的な保証を与えるにとどまることが確認された。しかし、耐量子性の観点で十分に高い信頼性を有する結果とみなされている。

（２）Module-LWE 問題に対する計算量見積（別紙 1 第 4-5 章、別紙 2 第 5 章）

①　BKZ シミュレーション[2]と dimension-for-free 技術に基づく計算量見積

Primal 攻撃の計算量(攻撃に必要なゲートコストとメモリ量)を見積もった結果[3]、攻撃に必要なゲートコストは、ML-KEM の全てのパラメータセットにおいて NIST 安全性水準を上回っており、十分な安全性を有することが確認された（表 1)。

表１．BKZ シミュレーションと dimension-for-free 技術に基づく primal 攻撃の計算量見積

	ML-KEM-512	ML-KEM-768	ML-KEM-1024
NIST 安全性レベル	レベル 1 (AES-128 相当)	レベル 3 (AES-192 相当)	レベル 5 (AES-256 相当)
要求されるゲートコスト（ビット）	143	207	272
攻撃に必要なゲートコスト（ビット）	151.5	215.1	287.3
攻撃に必要なメモリ量（ビット）	93.8	138.5	189.7

②　幾何級数仮定（GSA）と MATZOV 計算量モデルに基づく計算量見積

Primal 攻撃、dual 攻撃および hybrid 攻撃の計算量を見積った結果、攻撃に必要な

[2] BKZ は Block-Korkine-Zolotarev の略であり、格子基底簡約を行う BKZ アルゴリズムのシミュレーターを指す。

[3] 既存研究において、dual 攻撃が primal 攻撃と比較して計算コストがかかると予想されているため、ここでは Primal 攻撃の計算量見積のみ提供されている。

ゲートコストは、ML-KEM の全てのパラメータセットにおいて NIST 安全性レベルの水準を下回っていることが確認された（表 2）。しかし、これらの見積では、計算量に影響しうる重要な性質[4]が考慮されておらず、計算量見積が過小評価されている可能性がある。これらの性質を考慮した場合には、攻撃者に最も有利な状況を仮定しても、NIST 安全性水準を上回ることが議論されている。

表 2. 幾何級数仮定（GSA）と MATZOV 計算量モデルに基づく計算量見積

	ML-KEM-512	ML-KEM-768	ML-KEM-1024
NIST 安全性レベル	レベル 1 （AES-128 相当）	レベル 3 （AES-192 相当）	レベル 5 （AES-256 相当）
要求されるゲートコスト（ビット）	143	207	272
Primal 攻撃のゲートコスト（ビット）	140.2	201.0	270.7
Dual 攻撃のゲートコスト（ビット）	149.9	214.3	288.5
Hybrid 攻撃のゲートコスト（ビット）	139.7	196.4	262.3

（３）Module 構造を考慮した攻撃（別紙 1 第 4.5 節、別紙 2 第 5.4 節）

Module 構造を考慮した攻撃は、現時点において、Module 構造を考慮しない攻撃を上回るものではないということが確認された。

（４）暗号強度に関する考察

①　安田様の見解（別紙 1 第 6 章）

表 1 に示すとおり、攻撃に必要とされるゲートコストは、ML-KEM の全てのパラメータセットにおいて NIST 安全性水準を上回っており、十分な安全性を有する。さらに、最新の技術動向を踏まえても、現時点では表 1 の評価結果が覆る可能性は低い。

②　勝又様の見解（別紙 2 第 5.7 節）

調査対象とした全ての攻撃クラスにおいて、ML-KEM のいずれのパラメータセットに対しても、現実的な脅威とみなせる脆弱性は現在のところ発見されていない。さらに、具体的な計算量見積に基づく評価の結果、古典計算および量子計算の双方について攻撃者側に極めて有利な仮定を置いた場合であっても、NIST 安全性レベルに対して十分な安全性マージンを有していると考えられる。

３．２　ML-KEM の実装性能に関する調査及び評価

（１）サイドチャネル攻撃耐性（別紙 2 第 6 章）

秘密鍵の復元につながるサイドチャネル攻撃の可能性が確認された。これに対する対策として、マスキングおよびハイディングの適用が推奨される。

（２）ハードウェア実装性能（別紙 2 第 6.4 節）

回路面積の最適化実装[XL21]、計算時間の最適化実装[DMG23]およびサイドチャネル攻

[4] 例えば、篩処理の Progressive 化による格子基底の理想的な挙動からのずれ、BDGL 型篩処理で発生するオーバーヘッド等がある。

撃対策が施された実装[Kam+22]について紹介された（表 3）。

表 3. ML-KEM（Kyber）の FPGA 実装比較

参考文献	パラメータ	計算時間（μs）			回路面積		FPGA
		KeyGen	Encap	Decap	LUT(x1000)	FF(x1000)	
[XL21]	512	23.4	31.5	41.4	7.4	4.6	Artix-7
	768	39.2	49.2	62.4	7.4	4.6	
	1024	58.3	70.3	86.4	7.4	4.6	
[DMG23]	512	10.0	14.7	20.5	9.5	8.5	Artix-7
	768	12.0	17.0	22.2	10.5	9.8	
	1024	16.2	21.7	26.4	11.6	11.1	
[Kam+22] (M+H)	512	-	88.1	137.7	163.6	-	Virtex-7

※ M+H はマスキング対策とハイディング対策の両方を施した実装を表す。

（３）ソフトウェア実装性能（別紙 2 第 7 章）

① 計算時間

OpenSSL 3.6.0 を使用して測定した結果、ML-KEM は ECDH と同等以上の性能を発揮することが確認された（表 4）。

表 4. ECDH と ML-KEM における計算時間の比較（ミリ秒）

鍵交換アルゴリズム		安全性レベル	KeyGen	Encap	Decap
古典	EC X25519	1†	0.027	0.058	0.029
	EC P-256	1†	0.008	0.058	0.047
	EC P-384	3†	0.088	0.327	0.229
	EC P-521	5†	0.098	0.341	0.226
量子	ML-KEM-512	1	0.020	0.014	0.023
	ML-KEM-768	3	0.031	0.020	0.032
	ML-KEM-1024	5	0.047	0.028	0.043
ハイブリッド	X25519 + ML-KEM-768	1† + 3	0.061	0.076	0.060
	P-256 + ML-KEM-768	1† + 3	0.044	0.076	0.076
	P-384 + ML-KEM-1024	3† + 5	0.143	0.344	0.256

※ †は EC の古典安全性レベルを表しており、耐量子安全性は考慮していない。

② 帯域幅

ML-KEM と ECDH の帯域幅（具体的には、鍵長と暗号文長）を比較した結果、最大で 25 倍の差が生じていることが確認された（表 5）。なお、帯域幅の増加は対処可能であり、インターネット上での ML-KEM の使用を妨げるものではないことが実証された。

（４）実装性能に関する考察（別紙 2 Executive Summary）

ML-KEM の計算時間は従来方式と比べて高速である一方、鍵長や暗号文長が増加するため、メモリ制約が厳しいデバイス等においては実装上の課題が生じる可能性があるが、それ以外の用途においては問題なく利用できる。

サイドチャネル攻撃に対して厳密に保護されていることを前提として、政府および重要インフラシステムへの広範な展開にも適していると考えられる。

表 5. ECDH と ML-KEM における帯域幅（鍵長と暗号文長）の比較（バイト）

鍵交換アルゴリズム		安全性レベル	カプセル化鍵	暗号文
古典	EC X25519	1†	32	32
	EC P-256	1†	65	65
	EC P-384	3†	97	97
	EC P-521	5†	123	123
量子	ML-KEM-512	1	800	768
	ML-KEM-768	3	1184	1088
	ML-KEM-1024	5	1568	1568
ハイブリッド	X25519 + ML-KEM-768	1† + 3	1216	1120
	P-256 + ML-KEM-768	1† + 3	1249	1153
	P-384 + ML-KEM-1024	3† + 5	1665	1617

※ †は古典的な安全性レベルを表しており、耐量子安全性は考慮していない。

4　審議事項

（1）2025 年度外部評価報告書（別紙）に基づき、ML-KEM の安全性・実装性能に関する暗号技術評価委員会の見解を次のとおりとしてよろしいかご審議いただきたい。

- ML-KEM は、全てのパラメータセット（ML-KEM-512/768/1024）において、米国 NIST が規定する安全性レベル 1/3/5 を満たしている。
- ML-KEM は、従来方式と比較して高速である一方、鍵長および暗号文長が増加するが、メモリ制約が厳しいデバイスを除き、実用上問題なく利用できる。
- ML-KEM は、サイドチャネル攻撃対策が不十分な場合に脆弱性が生じる可能性があるものの、適切な対策の実装を前提とすれば、電子政府システムを含む多様なシステムへの広範な展開が可能である。

さらに、CRYPTREC Report 2025 暗号技術評価委員会報告として公開してよろしいかも併せてご審議いただきたい。

（2）2025 年度外部評価報告書（別紙）は、ML-KEM の安全性・実装性能に関する技術動向調査として十分な内容を含んでいると考えられる。このため、本報告書を CRYPTREC の技術調査報告書として公開してよろしいかご審議いただきたい。

以上

耐量子計算機暗号 ML-KEM の
安全性に関する調査及び評価

安田 雅哉
（立教大学理学部）

2026 年 3 月 4 日

第 1 章

調査結果・評価結果の概要（エグゼクティブサマリー）

FIPS 標準化された ML-KEM [64] は，加群格子上の LWE である Module-LWE 問題に基づく KEM で，通常（構造化なし）の LWE 問題に基づく方式に比べて効率的である．

■ML-KEM の構成概要　ML-KEM の基礎環は $R = \mathbb{Z}[X]/(X^n+1)$ $(n = 256)$ で，素数 $q = 3329$ を法とする剰余環 $R_q = R/qR$ を用いる．また，安全性レベルに応じて，3 種類の階数 $k \in \{2, 3, 4\}$ を選択する．ML-KEM では，秘密鍵 $\boldsymbol{s} = (s_1, \ldots, s_k) \in R_q^k$ とノイズ $\boldsymbol{e} = (e_1, \ldots, e_k) \in R_q^k$ の成分多項式 $s_i, e_i \in R_q$ のすべての $\mathbb{Z}_q$ 係数は，中心二項分布 CBD_η $(\eta \in \{2, 3\})$ からサンプルされる．本章では，すべてのベクトルは列ベクトルとする．このとき，すべての成分が R_q 上一様ランダムに選ばれた行列 $\boldsymbol{A} \in R_q^{k \times k}$ に対して，組 $(\boldsymbol{A}, \boldsymbol{t})$ を ML-KEM の公開鍵とする．ただし，$\boldsymbol{t} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} \in R_q^k$ とする．また，各係数が $\{0, 1\}$ に属する平文 $m \in R_q$ に対し，すべての $\mathbb{Z}_q$ 係数を CBD_η からサンプルした $\boldsymbol{y}, \boldsymbol{e}_1 \in R_q^k$ と $e_2 \in R_q$ を選び，公開鍵 $(\boldsymbol{A}, \boldsymbol{t})$ を用いて，

$$c = (\boldsymbol{u}, v) = \left(\boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{e}_1, \boldsymbol{t}^\top \boldsymbol{y} + e_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot m\right) \in R_q^k \times R_q \tag{1.1}$$

を m の暗号文とする．さらに，暗号文に対して，$v - \boldsymbol{s}^\top \boldsymbol{u} \in R_q$ の各 $\mathbb{Z}_q$ 係数をノイズ補正することで復号できる．このように構成した公開鍵暗号方式を，藤崎-岡本変換により KEM（ML-KEM）に変換する．ML-KEM では，R_q における乗算を高速化するために，数論変換が多用される．

■ML-KEM の証明可能安全性　ML-KEM の秘密鍵 $\boldsymbol{s} \in R_q^k$ に対して，R_q^k 上で一様にサンプルされた $\boldsymbol{a} \in R_q^k$ と $t = \boldsymbol{a}^\top \boldsymbol{s} + e \in R_q$ の組 $(\boldsymbol{a}, t) \in R_q^k \times R_q$ を Module-LWE サンプルという．ただし，$e \in R_q$ の各 $\mathbb{Z}_q$ 係数は CBD_η からサンプルされる．Module-LWE サンプルの分布が $R_q^k \times R_q$ 上の一様ランダム分布と識別困難な Module-LWE 仮定の下で，ML-KEM の基盤である公開鍵暗号方式は IND-CPA 安全である．したがって，藤崎-岡本変換で得られる ML-KEM は IND-CCA 安全である．その安全性証明は，ランダムオラクルモデル（ROM）においてはタイトである一方，量子ランダムオラクルモデル（QROM）においてはノンタイトである（文献 [13, §4] を参照）．しかし，いくつかの自然な仮定の下では，QROM においてもタイトな安全性帰着がある．

表 1.1　ML-KEM における 3 種類のパラメータとゲートコストによる攻撃計算量の見積もり

ML-KEM パラメータ（k：階数パラメータ）	512（$k=2$）	768（$k=3$）	1024（$k=4$）
攻撃可能な BKZ の最小ブロックサイズ β	413	637	894
攻撃に必要なゲートコスト（ビット）	151.5	215.1	287.3
NIST 標準化の安全性レベル [63]	レベル 1	レベル 3	レベル 5
要求される古典ゲート数（ビット）	143	207	272

■ML-KEM の安全性を支える Module-LWE 問題に対する攻撃計算量　現時点で，ML-KEM の安全性を支える Module-LWE 問題に対する最良の攻撃法は，$\mathbb{Z}_q$ 上の LWE 問題に帰着した上で，BKZ 基底簡約などの $\mathbb{Z}$ 格子上のアルゴリズムを適用するものである．暗号文 (1.1) の形から，攻撃に利用できる Module-LWE サンプル数は最大 $k+1$ 個であり，ML-KEM に対しては primal 攻撃と dual 攻撃が有効となる．表 1.1 に，ML-KEM パラメータと，それらに対する攻撃計算量の見積もりをまとめる．具体的には，ML-KEM パラメータに対し primal 攻撃と BKZ 基底簡約の組み合わせが有効で，攻撃者に有利な観点で，BKZ の progressive 化とシミュレーション [29]，dimension-for-free [32] など最新技術の効果を考慮する．また，BKZ の内部 SVP オラクルで呼び出す篩アルゴリズムの最内部にある繰り返し関数に対して，文献 [6] の解析に基づくゲートコストを表 1.1 に示す（詳細は文献 [13, Table 4] を参照）．表 1.1 から，各 ML-KEM パラメータに対して，攻撃に必要なゲートコストは耐量子計算機暗号の NIST 標準化 [63] の安全性レベル 1, 3, 5 で要求される古典ゲート数を上回り，十分な安全性を持つと考えられる．

篩アルゴリズムの解析の精密化・改良による影響　篩アルゴリズムの多角的な解析の精密化と，将来予想されるアルゴリズム的改良を考慮すると，表 1.1 内のゲートコスト評価は $-16 \sim 14$ 程度変動する可能性がある [13, §5.3, Summary]．最悪の場合，NIST 標準化で要求される古典ゲート数を下回る可能性があるが，これはあくまで攻撃者に最も有利な条件下での評価に過ぎない．実際には，文献 [56, §4.1.1] で指摘されているように，篩アルゴリズムのメモリアクセスのコストを現実的に反映した条件下では，NIST 標準化で要求される古典ゲート数は維持されると考えられる．（文献 [79] によるメモリアクセスのコスト削減は実用的なもので，[13] の予想の範囲内と考えられる．）

最新の dual-sieve 攻撃による影響　文献 [23] で新しい dual-sieve 攻撃が提案され，NIST 標準化で要求される古典ゲート数を下回ると主張している．しかし，その解析は理想的な理論モデルに基づき，オーバーヘッドが隠れている．また，LWE チャレンジで検証されている primal 攻撃に比べ，dual 攻撃の実用性の解析は進んでおらず，文献 [35] の指摘のように，dual 攻撃の成功確率は実際よりかなり高く見積もられている．したがって，文献 [23] の攻撃計算量は実際よりかなり小さく見積もられている可能性が高く，表 1.1 内のゲートコスト評価には影響しないと考えられる．

代数構造を利用した格子アルゴリズムの影響　加群格子上の BKZ 基底簡約は，$\mathbb{Z}$ 格子上のアルゴリズムと同程度の品質の基底を出力するか不明で，実用的な動作のための実装基盤も現時点では整備されていない．また，イデアル格子上の SVP に対する量子アルゴリズムは，Module-LWE への適用には障壁があり，文献 [56, Appendix C] の指摘のように，ML-KEM に対する実用的な攻撃に繋がる可能性は低い．以上から，現時点で，代数構造を利用したアルゴリズムは，代数構造を利用しないアルゴリズムよりも影響が大きいとは言えない．

第 2 章

ML-KEM の構成に関する解説

本章では，FIPS 標準として制定された耐量子計算機暗号の鍵カプセル化メカニズム（KEM）である ML-KEM [64] の構成について解説する．ML-KEM の安全性は Module-LWE 問題の計算困難性に基づく．具体的には，ML-KEM は，$\mathbb{Z}_q$ 上の LWE 問題に基づく Regev [67] の公開鍵暗号方式をひな形とし，それを Module-LWE 問題に一般化した公開鍵暗号方式を藤崎-岡本変換により KEM 変換した暗号方式である．

2.1 LWE 問題と Regev による公開鍵暗号方式

本節では，ML-KEM のひな形である LWE 問題に基づく Regev [67] による公開鍵暗号方式について解説する．そのために，$\mathbb{Z}_q$ 上の LWE 問題から述べる．

2.1.1 $\mathbb{Z}_q$ 上の LWE 問題

LWE（Learning with Errors）問題は機械学習理論から派生した計算問題で，奇素数 q による整数剰余類環 $\mathbb{Z}_q$ 上の秘密ベクトル $\boldsymbol{s} \in \mathbb{Z}_q^n$ 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 $\boldsymbol{s}$ 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 $\boldsymbol{s} = (s_1, s_2, s_3, s_4) \in \mathbb{Z}_{17}^4$ に関する連立線形近似方程式

$$
\begin{cases}
14s_1 + 15s_2 + 5s_3 + 2s_4 \approx 8 \pmod{17} \\
13s_1 + 14s_2 + 14s_3 + 6s_4 \approx 16 \pmod{17} \\
6s_1 + 10s_2 + 13s_3 + s_4 \approx 12 \pmod{17} \\
10s_1 + 4s_2 + 12s_3 + 16s_4 \approx 12 \pmod{17} \\
9s_1 + 5s_2 + 9s_3 + 6s_4 \approx 9 \pmod{17} \\
3s_1 + 6s_2 + 4s_3 + 5s_4 \approx 16 \pmod{17} \\
\qquad\qquad \vdots \qquad\qquad\qquad\quad \vdots \\
6s_1 + 7s_2 + 16s_3 + 2s_4 \approx 3 \pmod{17}
\end{cases}
$$

が与えられたとする（この数値例は文献 [68] から引用）．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．LWE 問題は，この連立線形近似方程式の解

$\boldsymbol{s} \in \mathbb{Z}_q^n$ を求める計算問題である．ちなみに，この数値例では $\boldsymbol{s} = (0, 13, 9, 11) \in \mathbb{Z}_{17}^4$ 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，ガウス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．

定式化された $\mathbb{Z}_q$ 上の LWE 問題 [67] は，以下である．

定義 2.1 (LWE 問題)**.** n を正の整数とし，q を奇素数とする．また，χ を $\mathbb{Z}_q$ 上のノイズ分布とする（例えば，χ として平均 0，標準偏差 $\sigma > 0$ の $\mathbb{Z}$ 上の離散ガウス分布 $D_{\mathbb{Z},\sigma}$ をとる）．$\mathbb{Z}_q^n$ 上一様ランダムに選ばれた秘密ベクトル $\boldsymbol{s} \in \mathbb{Z}_q^n$ を固定する．また，各成分が $\mathbb{Z}_q$ 上一様ランダムに選ばれた $\boldsymbol{a} \in \mathbb{Z}_q^n$ とノイズ分布 χ からサンプルされた $e \in \mathbb{Z}_q$ に対して，

$$(\boldsymbol{a}, t) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

の組を出力する確率分布を $L_{\boldsymbol{s},\chi}$ とする．ただし，

$$t = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e \bmod q \in \mathbb{Z}_q$$

とする（2 つのベクトル $\boldsymbol{v}$ と $\boldsymbol{w}$ の内積を $\langle \boldsymbol{v}, \boldsymbol{w} \rangle$ で表す）．このとき，次の 2 つの問題を考える．

- **判定問題**：与えられた複数の組 $(\boldsymbol{a}_i, t_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ $(i = 1, 2, \ldots, m)$ が，LWE における確率分布 $L_{\boldsymbol{s},\chi}$ からサンプルされた元か，$\mathbb{Z}_q^n \times \mathbb{Z}_q$ 上一様ランダムに生成された元かを決定せよ．
- **探索問題**：LWE における確率分布 $L_{\boldsymbol{s},\chi}$ からサンプルされた複数の組 $(\boldsymbol{a}_i, t_i)$ $(i = 1, 2, \ldots, m)$ から秘密ベクトル $\boldsymbol{s} \in \mathbb{Z}_q^n$ を復元せよ．

注意 2.1. 上記の LWE 問題について，探索問題の解である秘密ベクトル $\boldsymbol{s} \in \mathbb{Z}_q^n$ を得ることができれば，明らかに判定問題を解くことができる．逆に，探索問題は判定問題に帰着可能で [67, Lemma 4.2]，判定問題を解くオラクルを用いて探索問題を解くことができる．また，秘密ベクトル $\boldsymbol{s}$ を $\mathbb{Z}_q^n$ 一様ランダムに選んだ場合と，離散ガウス分布から選んだ場合の LWE 問題の計算困難性は等しい（詳細は [55] を参照）．

一般に，上記の 2 つの問題において，LWE における確率分布 $L_{\boldsymbol{s},\chi}$ は任意個の組 $(\boldsymbol{a}, t)$ をサンプルするオラクルとしてみなす．具体的には，ある固定したサンプル数 $m > 0$ に対して，LWE における確率分布 $L_{\boldsymbol{s},\chi}$ からサンプルされた異なる m 個の組

$$\begin{cases} (\boldsymbol{a}_1, t_1), & t_1 = \langle \boldsymbol{a}_1, \boldsymbol{s} \rangle + e_1 \bmod q \\ (\boldsymbol{a}_2, t_2), & t_2 = \langle \boldsymbol{a}_2, \boldsymbol{s} \rangle + e_2 \bmod q \\ \quad \vdots & \\ (\boldsymbol{a}_m, t_m), & t_m = \langle \boldsymbol{a}_m, \boldsymbol{s} \rangle + e_m \bmod q \end{cases}$$

から, LWE 問題を解くことを考える．特に, 求解に要する計算時間が最も短くなるような m を攻撃者が選べることを想定する．第 i 行ベクトルを $\boldsymbol{a}_i$ とする $m \times n$ 行列を $\boldsymbol{A}$ とし, $\boldsymbol{t} = (t_1, t_2, \ldots, t_m)$ とおく．このとき, 上記の m 個の LWE サンプルの組は

$$\big(\boldsymbol{A}, \boldsymbol{t}\big) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m \tag{2.1}$$

と簡潔に表せて，関係式

$$\boldsymbol{t} \equiv \boldsymbol{s}\boldsymbol{A}^{\top} + \boldsymbol{e} \pmod{q} \tag{2.2}$$

を満たす．ただし，$\boldsymbol{e} = (e_1, e_2, \ldots, e_m) \in \mathbb{Z}_q^m$ をノイズベクトルとする（各 e_i は χ からサンプルされた元であることに注意）．

2.1.2 Regev による公開鍵暗号方式

Regev による公開鍵方式 [67] の構成には, 以下の 4 つのパラメータが必要である．

- n: LWE 次元
- m: LWE サンプルの個数（$m \geq 1.1 \cdot n \log q$ となる最小の整数を選ぶ）
- q: 剰余パラメータ（$n^2 \leq q \leq 2n^2$ を満たす素数を選ぶ）
- $\alpha > 0$: 離散ガウス分布の標準偏差を定めるノイズパラメータ（$\alpha = 1/(\sqrt{n} \cdot \log^2 n)$）

以下に，具体的な公開鍵暗号方式の構成を示す．

■鍵生成　一様ランダムに秘密鍵ベクトル $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$ を選ぶ．次に，平均 0, 標準偏差 $\sigma = \alpha q$ の $\mathbb{Z}$ 上の離散ガウス分布 $\chi = D_{\mathbb{Z},\sigma}$ を用いて，秘密鍵ベクトル $\boldsymbol{s}$ による LWE 分布 $L_{\boldsymbol{s},\chi}$ から生成した m 個のサンプル

$$\{(\boldsymbol{a}_i, t_i)\}_{i=1}^{m}, \quad t_i = \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + e_i \bmod q \in \mathbb{Z}_q \quad (i = 1, 2, \ldots, m) \tag{2.3}$$

を公開鍵とする．ただし，各 e_i は χ からサンプリングされた元とする．

■暗号化　集合 $\{1, 2, \ldots, m\}$ の中から，一様ランダムに選んだ部分集合を S とする．このとき，上記の公開鍵を用いて，平文 $\mu \in \{0, 1\}$ の暗号文を次で定める．

$$c = (\boldsymbol{u}, v) = \left(\sum_{i \in S} \boldsymbol{a}_i,\ \mu \cdot \left\lfloor \frac{q}{2} \right\rceil + \sum_{i \in S} t_i \right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \tag{2.4}$$

■復号　式 (2.4) の形の暗号文 $c = \big(\boldsymbol{u}, v\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ に対し，秘密鍵ベクトル $\boldsymbol{s}$ を用いて

$$\big[v - \langle \boldsymbol{u}, \boldsymbol{s} \rangle \big]_q$$

を計算し, その値が十分 0 に近い場合は 0 を出力し, それ以外の場合は 1 を出力する．ただし, $[z]_q$ は元 $z \in \mathbb{Z}_q$ を $\big[-\frac{q}{2}, \frac{q}{2}\big)$ に収めた値とする．具体的には，法 q による整数 z の値が $0 \leq z < \frac{q}{2}$ であれば $[z]_q = z$ とし，$\frac{q}{2} \leq z < q$ であれば $[z]_q = z - q$ と定める．

復号の正当性

復号について，式 (2.4) の暗号文 $c = (\boldsymbol{u}, v)$ に対して，$\sigma \ll q$ であれば

$$v - \langle \boldsymbol{u}, \boldsymbol{s} \rangle = \sum_{i \in S} e_i + \mu \cdot \left\lfloor \frac{q}{2} \right\rceil \approx \mu \cdot \left\lfloor \frac{q}{2} \right\rceil \in \mathbb{Z}_q$$

が成り立つ．これより，$\mu = 0$ の場合は $[v - \langle \boldsymbol{u}, \boldsymbol{s} \rangle]_q \approx 0$, $\mu = 1$ の場合は $[v - \langle \boldsymbol{u}, \boldsymbol{s} \rangle]_q \approx \pm \frac{q}{2}$ が成り立つ．具体的には，

$$\left| \sum_{i \in S} e_i \right| \lesssim \sigma m < \frac{q}{4} \Longleftrightarrow 4\sigma m < q$$

であれば，高い確率で復号に成功する．

注意 2.2. 上記の暗号方式の構成において，式 (2.3) の公開鍵を，式 (2.1) のように $(\boldsymbol{A}, \boldsymbol{t})$ と行列表示する．ただし，公開鍵 $(\boldsymbol{A}, \boldsymbol{t})$ は LWE 関係式 (2.2) を満たす．このとき，式 (2.4) の形の暗号文は

$$c = (\boldsymbol{u}, v) = \left(\boldsymbol{y}\boldsymbol{A},\ \mu \cdot \left\lfloor \frac{q}{2} \right\rceil + \langle \boldsymbol{y}, \boldsymbol{t} \rangle \right)$$

と表すことができる．ただし，$\{1, 2, \ldots, m\}$ の部分集合 S に対して，$\boldsymbol{y} = (y_1, \ldots, y_m)$ の各成分 $y_i \in \{0, 1\}$ は

$$y_i = \begin{cases} 1 & (i \in S) \\ 0 & (i \notin S) \end{cases}$$

と定める．

2.2 ML-KEM の構成

ML-KEM は CRYSTALS-Kyber [13] に基づく加群格子上の KEM 方式で，その安全性は加群格子上の LWE 問題（Module-LWE 問題）の計算量困難性に基づく．具体的には，ML-KEM では，2 のべき数 $n = 2^8 = 256$ に対して，

$$R \coloneqq \mathbb{Z}[X]/(X^n + 1) \tag{2.5}$$

を基本環とし，素数 $q = 3329$ に対して

$$R_q \coloneqq R/qR = \mathbb{Z}_q[X]/(X^n + 1) \tag{2.6}$$

を R の剰余環とする．環 R_q の任意の元は $\mathbb{Z}_q$ を係数とする $n - 1$ 以下の次数の多項式

$$f(X) = f_0 + f_1 X + \cdots + f_{n-1} X^{n-1} \quad (f_i \in \mathbb{Z}_q)$$

と表せ，その係数ベクトル

$$\boldsymbol{f} = (f_0, f_1, \ldots, f_{n-1}) \in \mathbb{Z}_q^n$$

を対応させることで，$\mathbb{Z}_q$ 加群として R_q は $\mathbb{Z}_q^n$ と同型である．ML-KEM は，3 種類の（R_q 上の自由加群としての）階数パラメータ $k \in \{2, 3, 4\}$ に対し，$\mathbb{Z}_q$ 加群 $R_q^k \simeq (\mathbb{Z}_q^n)^k$ 上の LWE 問題を安全性の根拠とした KEM である．以下で，定式化した Module-LWE 問題を述べておく [21, 51]（$\mathbb{Z}_q$ 上の LWE 問題への帰着については，後述の 4.1 節を参照）．

定義 2.2 (Module-LWE 問題)**.** 秘密の元 $\boldsymbol{s}(X) = \big(s_1(X), \ldots, s_k(X)\big) \in R_q^k$ を固定する．また，一様ランダムに選ばれた $\boldsymbol{a}(X) = \big(a_1(X), \ldots, a_k(X)\big) \in R_q^k$ と R_q 上のノイズ分布 χ からサンプルされた $e(X) \in R_q$ に対して，

$$\begin{gathered} \big(\boldsymbol{a}(X), t(X)\big) \in R_q^k \times R_q, \\ t(X) = \big\langle \boldsymbol{a}(X), \boldsymbol{s}(X) \big\rangle + e(X) = \sum_{i=1}^{k} a_i(X) s_i(X) + e(X) \end{gathered} \tag{2.7}$$

の組を出力する確率分布を $L_{\boldsymbol{s}(X),\chi}$ とする（ML-KEM では，χ として中心二項分布サンプリングをとる）．ただし，$\big\langle \boldsymbol{a}(X), \boldsymbol{s}(X) \big\rangle \in R_q$ は，R_q を成分とする長さ k の 2 つのベクトル $\boldsymbol{a}(X), \boldsymbol{s}(X) \in R_q^k$ の内積とする．このとき，次の 2 つの問題を考える．ただし，サンプル数 m は，攻撃者を有利とする観点から適当に選べると仮定することが多い．

- **判定問題**：与えられた複数の組 $\big(\boldsymbol{a}_j(X), t_j(X)\big) \in R_q^k \times R_q$ $(j = 1, 2, \ldots, m)$ が，Module-LWE における確率分布 $L_{\boldsymbol{s}(X),\chi}$ からサンプルされた元か，$R_q^k \times R_q$ 上一様ランダムに生成された元かを決定せよ．
- **探索問題**：Module-LWE における確率分布 $L_{\boldsymbol{s}(X),\chi}$ からサンプルされた複数の組 $\big(\boldsymbol{a}_j(X), t_j(X)\big)$ $(j = 1, 2, \ldots, m)$ から秘密の元 $\boldsymbol{s}(X) \in R_q^k$ を復元せよ．

注意 2.3. 円分体 $K = R \otimes_{\mathbb{Z}} \mathbb{Q}$ のイデアル R の双対を $R^\vee$ とし，$R_q^\vee = R^\vee / qR^\vee$ とする．文献 [51] では，Module-LWE の秘密 $\boldsymbol{s}(X)$ は $(R_q^\vee)^k$ 上の一様サンプリングから選ばれる．ただし，式 (2.5) の形の R に対しては，$R^\vee = \frac{1}{n} R$ より，単純なスケーリングにより，秘密 $\boldsymbol{s}$ は $(R_q)^k$ 上の一様サンプリングから選ばれるとしてよい．通常の LWE 問題と同様で，$\boldsymbol{s}$ が $(R_q)^k$ 上一様ランダムに選ばれた場合と，$\boldsymbol{s}$ の成分多項式 $s_i(X)$ の $\mathbb{Z}_q$ 係数が $[-\eta, \eta]$ $(1 \leq \eta \ll q)$ 上一様ランダムに選ばれた場合の Module-LWE 問題の困難性は等しい [20]．ML-KEM では，$\boldsymbol{s}$ の各成分多項式 $s_i(X) \in R_q$ の $\mathbb{Z}_q$ 係数は中心二項分布からサンプリングされるが，この場合の Module-LWE 問題が $(R_q)^k$ 上一様ランダムに選ばれた場合と同程度の困難性をもつかどうかは証明されていない．さらに，注意 2.1 と同じように，Module-LWE 問題においても，探索問題が解ければ判定問題を解くことができる．また，探索問題は判定問題に多項式時間帰着可能なので，判定問題と探索問題は多項式時間帰着の意味で等価である（詳細は文献 [51, 57] を参照）．

ML-KEM では，R_q における乗算を高速化するために，Number-Theoretic Transform（NTT）とよばれる数論変換を利用する．以降では，ML-KEM の最も基本となる構成要素である NTT を説明したのちに，ML-KEM の構成について説明する．

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

NTT は，環 R_q の元 $f(X)$ を R_q と同型な環 T_q の元 $\widehat{f}$ に写し，T_q における乗算を利用して効率的に R_q の 2 つの元の乗算を行う手法である．これは複素数体 $\mathbb{C}$ 上の高速フーリエ変換による多項式乗算と同じアイデアで，NTT はその $\mathbb{Z}_q$ 上版とみなせる．上述したように，ML-KEM では

2 のべき数 $n = 2^8 = 256$ と素数 $q = 3329$ で定まる剰余環 R_q を用いる（ML-KEM の暗号パラメータについては，後述の 2.2.3 節を参照）．これらの暗号パラメータの組

$$(n, q) = \big(256, 3329\big)$$

において，$\mathbb{Z}_q^* \coloneqq \mathbb{Z}_q \setminus \{0\}$ は位数 $q - 1 = 3328 = 2^8 \cdot 13$ の巡回群で，$\mathbb{Z}_q^*$ は位数 $2^8 = 256 = n$ の巡回部分群 $\langle\zeta\rangle$ を唯一つ含む．具体的には，$\mathbb{Z}_q$ において

$$\zeta \coloneqq 17 \bmod q \in \mathbb{Z}_q$$

が 1 の原始 n 乗根で，ζ の奇数べきによる集合

$$\big\{\zeta, \zeta^3, \zeta^5, \ldots, \zeta^{n-1}\big\}$$

が $\mathbb{Z}_q$ に含まれる 1 の原始 n 乗根全体の集合である．ここで，$N = \frac{n}{2} = 128$ とおくと，各 $i = 0, 1, \ldots, N-1$ に対して，

$$\zeta^{(2i+1)N} \equiv -1 \pmod{q}$$

が成り立つ．ゆえに，多項式環 $\mathbb{Z}_q[X]$ において，$X^n + 1$ は次のように N 個の 2 次式の積に分解できる．

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

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

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

具体的には，この同型は

$$\begin{aligned} \mathsf{NTT} : R_q &\longrightarrow T_q, \\ f(X) &\longmapsto \widehat{f} \coloneqq \big(f \bmod (X^2 - \zeta^{2i+1})\big)_{i=0}^{N-1} \end{aligned} \tag{2.8}$$

で定まる．ここで，T_q を **NTT 空間**（NTT domain），$\widehat{f} = \mathsf{NTT}\big(f\big) \in T_q$ を $f(X) \in R_q$ の **NTT 表現**（NTT representation）とよぶ．

NTT 表現について

R_q の元 $f(X) = f_0 + f_1 X + \cdots + f_{n-1} X^{n-1}$ の偶数（even）と奇数（odd）の次数に関する多項式をそれぞれ

$$\begin{cases} f_e(Y) \coloneqq f_0 + f_2 Y + f_4 Y^2 + \cdots + f_{2N-2} Y^{N-1}, \\ f_o(Y) \coloneqq f_1 + f_3 Y + f_5 Y^2 + \cdots + f_{2N-1} Y^{N-1} \end{cases}$$

とおく（$N = \frac{n}{2}$ に注意）．この構成から，明らかに

$$f(X) = f_e(X^2) + f_o(X^2)X \tag{2.9}$$

が成り立つ．ここで，各 $i = 0, 1, \ldots, N-1$ に対して，

$$\begin{cases} \widehat{f}_{2i} \coloneqq f_e(\zeta^{2i+1}) = \displaystyle\sum_{j=0}^{N-1} f_{2j}\zeta^{(2i+1)j}, \\ \widehat{f}_{2i+1} \coloneqq f_o(\zeta^{2i+1}) = \displaystyle\sum_{j=0}^{N-1} f_{2j+1}\zeta^{(2i+1)j} \end{cases} \tag{2.10}$$

とおくと，式 (2.9) より

$$f(X) \equiv \widehat{f}_{2i} + \widehat{f}_{2i+1}X \pmod{\left(X^2 - \zeta^{2i+1}\right)} \tag{2.11}$$

が成り立つ（X^2 に ζ^{2i+1} を代入したと考えればよい）．これより，$f(X) \in R_q$ の NTT 表現は

$$\widehat{f} = \left(\widehat{f}_{2i} + \widehat{f}_{2i+1}X\right)_{i=0}^{N-1} \in T_q$$

とかける（式 (2.8) を参照）．

NTT 表現の行列表示

$\mathbb{Z}_q$ の元を成分とする $N \times N$ 行列を

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

とおく．R_q の元 $f(X) = f_0 + f_1X + \cdots + f_{n-1}X^{n-1}$ の偶数と奇数の次数に関するそれぞれの係数ベクトル $(f_0, f_2, \ldots, f_{2N-2}), (f_1, f_3, \ldots, f_{2N-1}) \in \mathbb{Z}_q^N$ に対して，式 (2.10) より

$$\begin{pmatrix} \widehat{f}_0 \\ \widehat{f}_2 \\ \widehat{f}_4 \\ \vdots \\ \widehat{f}_{2N-2} \end{pmatrix} = \begin{pmatrix} f_e(1) \\ f_e(\zeta^3) \\ f_e(\zeta^5) \\ \vdots \\ f_e(\zeta^{2N-1}) \end{pmatrix} = \boldsymbol{B} \begin{pmatrix} f_0 \\ f_2 \\ f_4 \\ \vdots \\ f_{2N-2} \end{pmatrix},$$

$$\begin{pmatrix} \widehat{f}_1 \\ \widehat{f}_3 \\ \widehat{f}_5 \\ \vdots \\ \widehat{f}_{2N-1} \end{pmatrix} = \begin{pmatrix} f_o(1) \\ f_o(\zeta^3) \\ f_o(\zeta^5) \\ \vdots \\ f_o(\zeta^{2N-1}) \end{pmatrix} = \boldsymbol{B} \begin{pmatrix} f_1 \\ f_3 \\ f_5 \\ \vdots \\ f_{2N-1} \end{pmatrix}$$

が成り立つ．つまり，$f \in R_q$ の偶数と奇数の次数の係数ベクトルはそれぞれ行列 $\boldsymbol{B}$ による線形変換（つまり，離散フーリエ変換）で $\widehat{f} \in T_q$ の偶数と奇数の添え字番号のベクトルに写る．行列 $\boldsymbol{B}$ の逆行列は

$$\boldsymbol{C} = \frac{1}{N} A(\zeta^{-1}) \in \mathbb{Z}_q^{N \times N}$$

で与えられるので，式 (2.8) の NTT 写像の逆写像 NTT^{-1} は，行列 $\boldsymbol{C}$ を用いて同様に計算できる（つまり，逆離散フーリエ変換から計算可能．具体的な NTT の計算アルゴリズムは，FIPS 仕様書 [64, Algorithms 9, 10] を参照）．

NTT 空間における乗算

R_q の 2 つの元 $f(X), g(X)$ に対して，その積を

$$h(X) = f(X) \cdot g(X) \in R_q$$

とおく．$h(X)$ の NTT 表現 $\widehat{h} \in T_q$ について，式 (2.11) から，各 $i = 0, 1, \ldots, N-1$ に対して

$$\begin{aligned} \widehat{h}_{2i} + \widehat{h}_{2i+1} X &\equiv h(X) = f(X) \cdot g(X) \\ &\equiv \left(\widehat{f}_{2i} + \widehat{f}_{2i+1} X\right) \left(\widehat{g}_{2i} + \widehat{g}_{2i+1} X\right) \pmod{\left(X^2 - \zeta^{2i+1}\right)} \end{aligned}$$

が成り立つ．ここで，2 つの NTT 表現

$$\widehat{f} = \left(\widehat{f}_{2i} + \widehat{f}_{2i+1} X\right)_{i=0}^{N-1}, \quad \widehat{g} = (\widehat{g}_{2i} + \widehat{g}_{2i+1} X)_{i=0}^{N-1} \in T_q$$

の積を

$$\begin{aligned} \widehat{f} \circ \widehat{g} &\coloneqq \left(\left(\widehat{f}_{2i} + \widehat{f}_{2i+1} X\right) \cdot \left(\widehat{g}_{2i} + \widehat{g}_{2i+1} X\right) \bmod \left(X^2 - \zeta^{2i+1}\right)\right)_{i=0}^{N-1} \\ &= \left(\widehat{f}_{2i} \widehat{g}_{2i} + \widehat{f}_{2i+1} \widehat{g}_{2i+1} \zeta^{2i+1} + \left(\widehat{f}_{2i} \widehat{g}_{2i+1} + \widehat{f}_{2i+1} \widehat{g}_{2i}\right) X\right)_{i=0}^{N-1} \in T_q \end{aligned} \tag{2.12}$$

と定める（法 $(X^2 - \zeta^{2i+1})$ において，$X^2 = \zeta^{2i+1}$ であることに注意）．このとき，

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

が成り立つ（つまり，式 (2.8) の NTT 写像は環の同型写像である）．特に，NTT 空間 T_q における乗算は，成分ごとの演算であるため，（R_q における乗算に比べて）効率的に計算可能である．具体的には，R_q における乗算には $O(n^2)$ 回の $\mathbb{Z}_q$ 上の乗算が必要であるのに対し，式 (2.12) から，NTT 空間における乗算には $4n = O(n)$ 回だけの $\mathbb{Z}_q$ 上の乗算を要する．また，式 (2.8) の NTT 写像の計算は高速数論変換を用いて $O(n \log n)$ で可能であるため，NTT 変換の計算時間を含めても R_q での乗算よりも高速となる．

2.2.2 ML-KEM の基本構成と処理概要

KEM は公開チャネル上で二者が安全に秘密情報を共有するためのアルゴリズム群である．安全に共有された秘密情報は共通鍵暗号の鍵生成の乱数シードなどに用いられ，暗号や認証などの安全なやり取りの中で重要な役割を果たす．（R_q 上の自由加群としての）階数 k の R_q 加群 $R_q^k \simeq (\mathbb{Z}_q^n)^k$ 上の LWE 問題に基づく ML-KEM は，次の 2 つのステップで構成される．

- 1 つ目は，Module-LWE 問題の計算困難性に基づく公開鍵暗号（K-PKE）を構成する．
- 2 つ目は，K-PKE を藤崎-岡本変換により KEM（ML-KEM）に変換する．

藤崎-岡本変換の性質により，公開鍵暗号方式から構成される KEM はより一般的な攻撃モデルにおいて安全であり，IND-CCA2 安全性を満たす（詳しくは次章を参照）．本項では，FIPS 文書 [64] に合わせて，すべてのベクトルは列ベクトルとする．

K-PKE の処理概要

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

■K-PKE 鍵生成アルゴリズム Algorithm 1 に，鍵生成アルゴリズム（[64, Algorithm 13], K-PKE.KeyGen）の主な処理をまとめる．具体的には，乱数 d を入力として，暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ と復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ を出力する．ステップ 2 において，NTT 表現の公開鍵行列の各成分 $\widehat{\boldsymbol{A}}[i,j]$ は，入力する乱数から擬似ランダムな T_q の元を出力する SampleNTT 関数 [64, Algorithm 7] を用いて生成する（具体的には，$\widehat{\boldsymbol{A}}[i,j] \leftarrow \mathsf{SampleNTT}(\rho \| i \| j)$ により生成）．ステップ 3, 4 において，各多項式 $\boldsymbol{s}[i]$ または $\boldsymbol{e}[i]$ のすべての十分小さい $\mathbb{Z}_q$ 係数は，SamplePolyCBD 関数 [64, Algorithm 8] を用いて生成する．具体的には，$\eta \in \{2,3\}$ に対する $\mathbb{Z}_q$ 上の中心二項分布 CBD_η（Centered Binomial Distribution）を

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

と定め，$\boldsymbol{s}[i]$ と $\boldsymbol{e}[i]$ の各 $\mathbb{Z}_q$ 係数は CBD_η からサンプルする．ただし，CBD_η の乱数シードとして，ステップ 1 で生成した σ を用いる．ステップ 8 において，[64] では $\left(\widehat{\boldsymbol{t}}, \rho\right)$ と $\widehat{\boldsymbol{s}}$ をそれぞれ符号化関数 ByteEncode [64, Algorithm 5] で符号化したものを暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ と復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ とする．この鍵生成アルゴリズムにおいて，ρ から NTT 表現の公開鍵行列 $\widehat{\boldsymbol{A}}$ が復元可能なので, 暗号鍵 $\mathtt{ek}_{\mathtt{PKE}}$ は NTT 表現の Module-LWE インスタンスの組

$$\left(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{t}}\right)$$

に対応する．特に，それらの NTT 逆変換を

$$\boldsymbol{t} = \mathsf{NTT}^{-1}\left(\widehat{\boldsymbol{t}}\right) \in R_q^k, \quad \boldsymbol{A} = \mathsf{NTT}^{-1}\left(\widehat{\boldsymbol{A}}\right) \in (R_q)^{k \times k}$$

Algorithm 1 K-PKE.KeyGen：K-PKE 鍵生成アルゴリズム（[64, Algorithm 13] の簡略版）

入力： 乱数 d

出力： 暗号鍵 $\mathsf{ek}_{\mathsf{PKE}}$ と復号鍵 $\mathsf{dk}_{\mathsf{PKE}}$

1: $(\rho, \sigma) \leftarrow \mathsf{G}(d \| k)$ ▷ ハッシュ関数 G を用いて擬似ランダムな乱数の組 (ρ, σ) を生成

2: $\widehat{\boldsymbol{A}} = \left(\widehat{\boldsymbol{A}}[i,j]\right)_{i,j=0}^{k-1} \in (T_q)^{k \times k}$ ▷ 乱数 ρ から NTT 表現の公開鍵行列を生成

3: $\boldsymbol{s} = \left(\boldsymbol{s}[i]\right)_{i=0}^{k-1} \in R_q^k$
▷ 各 $\boldsymbol{s}[i] \in R_q$ のすべての $\mathbb{Z}_q$ 係数は中心二項分布 CBD_η からサンプル（十分小さい）

4: $\boldsymbol{e} = \left(\boldsymbol{e}[i]\right)_{i=0}^{k-1} \in R_q^k$
▷ 各 $\boldsymbol{e}[i] \in R_q$ のすべての $\mathbb{Z}_q$ 係数は CBD_η からサンプル（十分小さい）

5: $\widehat{\boldsymbol{s}} = \left(\mathsf{NTT}(\boldsymbol{s}[i])\right)_{i=0}^{k-1} \in T_q^k$ ▷ 各 $\boldsymbol{s}[i]$ を NTT 変換

6: $\widehat{\boldsymbol{e}} = \left(\mathsf{NTT}(\boldsymbol{e}[i])\right)_{i=0}^{k-1} \in T_q^k$ ▷ 各 $\boldsymbol{e}[i]$ を NTT 変換

7: $\widehat{\boldsymbol{t}} = \widehat{\boldsymbol{A}} \circ \widehat{\boldsymbol{s}} + \widehat{\boldsymbol{e}} = \left(\sum_{j=0}^{k-1} \widehat{\boldsymbol{A}}[i,j] \circ \widehat{\boldsymbol{s}}[j] + \widehat{\boldsymbol{e}}[i]\right)_{i=0}^{k-1} \in T_q^k$
▷ NTT 空間上で，R_q^k 上の LWE 関係式 $\boldsymbol{t} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$ を生成（式 (2.13) を参照）

8: $\mathsf{ek}_{\mathsf{PKE}} = \left(\widehat{\boldsymbol{t}}, \rho\right)$, $\mathsf{dk}_{\mathsf{PKE}} = \widehat{\boldsymbol{s}}$
▷ $\widehat{\boldsymbol{A}}$ は ρ から復元できるので，$\mathsf{ek}_{\mathsf{PKE}}$ は Module-LWE インスタンスの組 $\left(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{t}}\right)$ に対応

9: **return** $\left(\mathsf{ek}_{\mathsf{PKE}}, \mathsf{dk}_{\mathsf{PKE}}\right)$

とおく．ただし，ベクトルまたは行列に対する NTT^{-1} は，各成分に対する NTT 逆変換とする．このとき，R_q^k 上の LWE 関係式

$$\begin{aligned} & \boldsymbol{t} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} \in R_q^k \\ \iff & \boldsymbol{t}[i] = \sum_{j=0}^{k-1} \boldsymbol{A}[i,j] \cdot \boldsymbol{s}[j] + \boldsymbol{e}[i] \in R_q \quad (i = 0, 1, \ldots, k-1) \end{aligned} \tag{2.13}$$

が成り立つ（式 (2.7) を参照）．ただし，$\boldsymbol{t}[i], \boldsymbol{e}[i], \boldsymbol{A}[i,j], \boldsymbol{s}[j]$ はそれぞれ R_q の元で，$\mathbb{Z}_q$ 係数の $n-1$ 次以下の多項式で表されることに注意する．一方，復号鍵 $\mathsf{dk}_{\mathsf{PKE}}$ は NTT 表現の LWE の秘密 $\widehat{\boldsymbol{s}}$ であるので，暗号鍵から復号鍵を見つけるのは $T_q^k \simeq R_q^k$ 上の探索 LWE 問題（つまり，探索 Module-LWE 問題）である．特に，適切な暗号パラメータ（後述の 2.2.3 節を参照）を利用した場合，その Module-LWE 問題を解くのは計算量的に非常に困難である．また，鍵生成アルゴリズムにおいて，NTT 空間上で公開鍵行列 $\widehat{\boldsymbol{A}}$ を直接生成すると共に，ステップ 7 で NTT 空間上で Module-LWE 関係式を生成することで，計算の高速化を図る．

■K-PKE 暗号化アルゴリズム Algorithm 2 に，暗号化アルゴリズム（[64, Algorithm 14], K-PKE.Encrypt）の主な処理をまとめる．具体的には，暗号化鍵 $\mathsf{ek}_{\mathsf{PKE}}$，平文 m と乱数 r を入力し，暗号文 c を出力する．ステップ 2, 3, 4 において，r をシードとした擬似乱数を引数とした SamplePolyCBD 関数で，すべての $\mathbb{Z}_q$ 係数が十分小さい多項式を生成する．ステップ 7 では，バイト列で表現された平文 m を ByteDecode 関数 [64, Algorithm 6] でビット列

Algorithm 2 K-PKE.Encrypt：K-PKE 暗号化アルゴリズム（[64, Algorithm 14] の簡略版）

入力： 暗号化鍵 $\mathtt{ek}_{\mathtt{PKE}} = \left(\widehat{\boldsymbol{t}}, \rho\right)$，平文 m と乱数 r

出力： 暗号文 c

1: ρ から NTT 表現の公開鍵行列 $\widehat{\boldsymbol{A}} \in (T_q)^{k \times k}$ を復元

2: $\boldsymbol{y} = \left(\boldsymbol{y}[i]\right)_{i=0}^{k-1} \in R_q^k$

▷ 各 $\boldsymbol{y}[i] \in R_q$ のすべての $\mathbb{Z}_q$ 係数は中心二項分布 CBD_η からサンプル（十分小さい）

3: $\boldsymbol{e}_1 = (\boldsymbol{e}_1[i])_{i=0}^{k-1} \in R_q^k$

▷ 各 $\boldsymbol{e}[i] \in R_q$ のすべての $\mathbb{Z}_q$ 係数は CBD_η からサンプルする（十分小さい）

4: $e_2 \in R_q$ ▷ すべての $\mathbb{Z}_q$ 係数は CBD_η からサンプルする（十分小さい）

5: $\widehat{\boldsymbol{y}} = (\mathsf{NTT}\left(\boldsymbol{y}[i]\right))_{i=0}^{k-1} \in T_q^k$ ▷ 各 $\boldsymbol{y}[i]$ を NTT 変換

6: $\boldsymbol{u} = \mathsf{NTT}^{-1}\left(\widehat{\boldsymbol{A}}^\top \circ \widehat{\boldsymbol{y}}\right) + \boldsymbol{e}_1 = \boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{e}_1 = \left(\sum_{j=0}^{k-1} \boldsymbol{A}[j,i]\boldsymbol{y}[j] + \boldsymbol{e}_1[i]\right)_{i=0}^{k-1} \in R_q^k$

▷ ただし，$\boldsymbol{A} = \mathsf{NTT}^{-1}(\widehat{\boldsymbol{A}}) = (\boldsymbol{A}[i,j])_{i,j=0}^{k-1} \in (R_q)^{k \times k}$ とする

7: $\mu = \mathsf{Decompress}\left(\mathsf{ByteDecode}(m)\right) \in R_q$

▷ 平文 m をビット列化した後に R_q の元に変換（式 (2.14) と (2.15) を参照）

8: $v = \mathsf{NTT}^{-1}\left(\widehat{\boldsymbol{t}}^\top \circ \widehat{\boldsymbol{y}}\right) + e_2 + \mu = \boldsymbol{t}^\top \boldsymbol{y} + e_2 + \mu \in R_q$

9: **return** $c = \left(\boldsymbol{u}, v\right) \in R_q^k \times R_q$ ▷ 式 (2.16) を参照

▷ FIPS 203 では，$\boldsymbol{u}, v$ を Compress 関数で圧縮したあと，ByteEncode 関数で符号化

$(m_0, m_1, \ldots, m_{n-1}) \in \{0,1\}^n$ に変換した後に，各ビット $m_i \in \{0,1\}$ を Decompress 関数で

$$\mu_i = \left\lceil \frac{q}{2} \cdot m_i \right\rfloor \in \mathbb{Z}_q \tag{2.14}$$

に変換する．ただし，$\lceil a \rfloor$ は有理数 $a \in \mathbb{Q}$ の最近似整数への丸め込みとする．また，各 μ_i を係数とする多項式を

$$\mu = \mu_0 + \mu_1 X + \cdots + \mu_{n-1} X^{n-1} \in R_q \tag{2.15}$$

とする．正確には，FIPS 203 [64] では，ステップ 9 において，$\boldsymbol{u}$ と v はそれぞれ Compress 関数で圧縮した後，ByteEncode 関数で符号化する．出力する暗号文は

$$c = (\boldsymbol{u}, v) = \left(\boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{e}_1, \boldsymbol{t}^\top \boldsymbol{y} + e_2 + \mu\right) \in R_q^k \times R_q \tag{2.16}$$

の形で，LWE に基づく Regev の公開鍵暗号方式の暗号文 (2.4) と同じである （注意 2.2 を参照）．また，ステップ 6 と 8 において，NTT 空間上で行列-ベクトル積 $\boldsymbol{A}^\top \boldsymbol{y}$ と内積 $\boldsymbol{t}^\top \boldsymbol{y} = \langle \boldsymbol{t}, \boldsymbol{y} \rangle$ を計算することで，計算の高速化を図る．

■K-PKE 復号アルゴリズム アルゴリズム 3 に，復号アルゴリズム（[64, Algorithm 15], K-PKE.Decrypt）の主な処理をまとめる．具体的には，復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ と暗号文 c を入力とし，復号文 m' を出力する．ステップ 2 において，多項式表現の R_q の元 $w = w_0 + w_1 X + \cdots + w_{n-1} X^{n-1}$

Algorithm 3 K-PKE.Decrypt：K-PKE 復号アルゴリズム（[64, Algorithm 15] の簡略版）

入力： 復号鍵 $\mathtt{dk}_{\mathtt{PKE}} = \widehat{\boldsymbol{s}}$ と暗号文 $c = (\boldsymbol{u}, v)$

出力： 復号文 m'

1: $w = v - \mathsf{NTT}^{-1}\left(\widehat{\boldsymbol{s}}^{\top} \circ \mathsf{NTT}(\boldsymbol{u})\right) = v - \boldsymbol{s}^{\top}\boldsymbol{u} \in R_q$ ▷ メインの復号処理

2: **return** $m' = \mathsf{ByteEncode}\left(\mathsf{Compress}(w)\right)$

に対して，各係数 $w_i \in \mathbb{Z}_q$ を Compress 関数で

$$z_i = \left\lceil \frac{2}{q} \cdot w_i \right\rfloor \bmod 2 \in \{0, 1\} \tag{2.17}$$

に変換する．また，ビット列 $(z_0, z_1, \ldots, z_{n-1})$ を ByteEncode 関数 [64, Algorithm 5] でバイト列に変換する．特に，ByteEncode 関数と ByteDecode 関数はお互いの逆関数である．

復号の正当性

式 (2.16) の暗号文 $c = (\boldsymbol{u}, v)$ に対して，R_q^k 上の LWE 関係式 (2.13) から，

$$\begin{aligned}
w &= v - \boldsymbol{s}^{\top}\boldsymbol{u} && \text{(Algorithm 3 のステップ 1 を参照)} \\
&= \left(\boldsymbol{t}^{\top}\boldsymbol{y} + e_2 + \mu\right) - \boldsymbol{s}^{\top}\left(\boldsymbol{A}^{\top}\boldsymbol{y} + \boldsymbol{e}_1\right) && \text{(式 (2.16) を利用)} \\
&= \boldsymbol{t}^{\top}\boldsymbol{y} + e_2 + \mu - (\boldsymbol{A}\boldsymbol{s})^{\top}\boldsymbol{y} - \boldsymbol{s}^{\top}\boldsymbol{e}_1 \\
&= \boldsymbol{t}^{\top}\boldsymbol{y} + e_2 + \mu - (\boldsymbol{t} - \boldsymbol{e})^{\top}\boldsymbol{y} - \boldsymbol{s}^{\top}\boldsymbol{e}_1 && \text{(式 (2.13) を利用)} \\
&= \mu + \underbrace{e_2 + \boldsymbol{e}^{\top}\boldsymbol{y} - \boldsymbol{s}^{\top}\boldsymbol{e}_1}_{\text{すべての } \mathbb{Z}_q \text{ 係数が十分小さい}} \in R_q
\end{aligned}$$

が成り立つ．ここで，$\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{e}_1, \boldsymbol{y} \in R_q^k$ の各成分 $\boldsymbol{s}[i], \boldsymbol{e}[i], \boldsymbol{e}_1[i], \boldsymbol{y}[i] \in R_q$ と $e_2 \in R_q$ のすべての $\mathbb{Z}_q$ 係数は十分小さいことに注意する（すべて $\mathbb{Z}_q$ 上の中心二項分布 CBD_η からサンプリング）．よって，Compress 関数による各 $\mathbb{Z}_q$ 係数におけるノイズ補正（式 (2.17) を参照）により

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

が成り立つ．ただし，2 段目の式変形については，各 $\mathbb{Z}_q$ 係数において式 (2.14) と (2.15) から

$$\left\lceil \frac{2}{q} \cdot \mu_i \right\rfloor \bmod 2 = \left\lceil \frac{2}{q} \cdot \left\lceil \frac{q}{2} \cdot m_i \right\rfloor \right\rfloor \bmod 2 = m_i \in \{0, 1\}$$

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

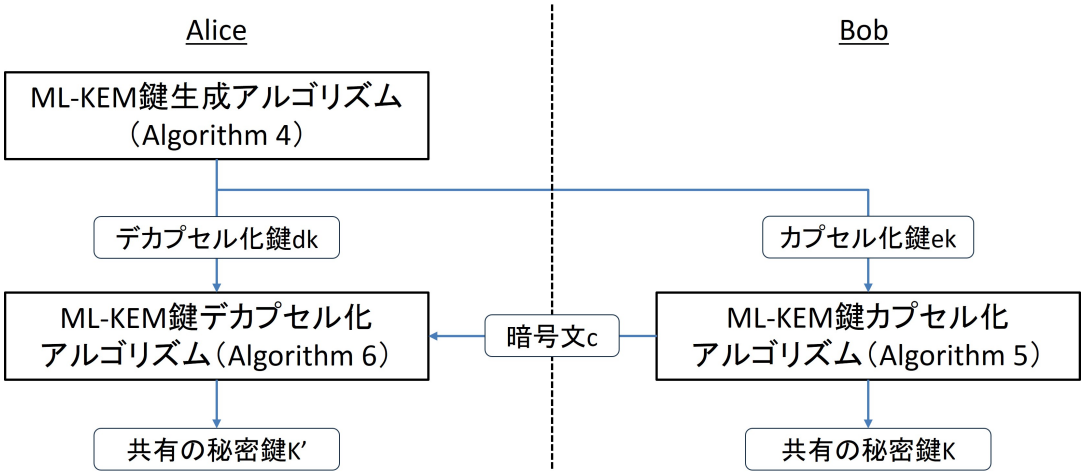


図 2.1　ML-KEM の全体の処理の流れ（文献 [64, Figure 1] を参照）

ML-KEM の処理概要

上記で構成した K-PKE 方式（Algorithm 1, 2, 3）を用いて，ML-KEM は下記の 3 つのアルゴリズムで構成される（ML-KEM の全体の処理の流れは図 2.1 を参照）．

■ML-KEM 鍵生成アルゴリズム　Algorithm 4 に，鍵生成アルゴリズム [64, Algorithm 16] の主な処理をまとめる．具体的には，K-PKE 鍵生成アルゴリズム（Algorithm 1）を用いて，入力する 2 つの乱数 d, z から，カプセル化鍵 $\mathtt{ek}$ とデカプセル化鍵 $\mathtt{dk}$ を出力する．

■ML-KEM 鍵カプセル化アルゴリズム　Algorithm 5 に，鍵カプセル化アルゴリズム [64, Algorithm 17] の主な処理をまとめる．具体的には，K-PKE 暗号化アルゴリズム（Algorithm 2）を用いて，カプセル化鍵 $\mathtt{ek}$ と乱数 m から共有の秘密鍵 K と暗号文 c を出力する．特に，m と $\mathtt{ek}$ のハッシュ値 (K, r) に対して，暗号文 c は $(\mathtt{ek}, m, r)$ から一意的に（つまり，決定的に）生成する．

■ML-KEM 鍵デカプセル化アルゴリズム　Algorithm 6 に，鍵デカプセル化アルゴリズム [64, Algorithm 18] の主な処理をまとめる．具体的には，K-PKE 復号アルゴリズム（Algorithm 3）を用いて，デカプセル化鍵 $\mathtt{dk}$ と暗号文 c から，共有の秘密鍵（のコピー）K' を出力する．特に，入力する暗号文 c が改竄されていないことを保証するために，K-PKE 暗号化アルゴリズム（Algorithm 2）で復号文から暗号文 c' を生成し，c と c' が一致するか検証する．ただし，上記の鍵カプセル化アルゴリズムと同じように，復号文 m' と $\mathtt{ek}$ のハッシュ値 (K', r') に対して，暗号文 c' は $(\mathtt{ek}, m', r')$ から一意的に生成する．

ML-KEM の正当性

鍵デカプセル化アルゴリズム（Algorithm 6）に入力する暗号文 c が，鍵カプセル化アルゴリズム（Algorithm 5）に入力する $(\mathtt{ek}, m)$ から正当に計算されたものとする．このとき，Algorithm 6

Algorithm 4 ML-KEM 鍵生成アルゴリズム [64, Algorithm 16]

入力： 2 つの乱数 d, z

出力： カプセル化鍵 $\mathtt{ek}$ とデカプセル化鍵 $\mathtt{dk}$

1: K-PKE 鍵生成アルゴリズム（Algorithm 1）で，乱数 d から $(\mathtt{ek}_{\mathtt{PKE}}, \mathtt{dk}_{\mathtt{PKE}})$ を生成

2: $\mathtt{ek} = \mathtt{ek}_{\mathtt{PKE}}$

3: $\mathtt{dk} = \big(\mathtt{dk}_{\mathtt{PKE}}, \mathtt{ek}, \mathsf{H}(\mathtt{ek}), z\big)$ ▷ H はハッシュ関数

4: **return** $\big(\mathtt{ek}, \mathtt{dk}\big)$

Algorithm 5 ML-KEM 鍵カプセル化アルゴリズム [64, Algorithm 17]

入力： カプセル化鍵 $\mathtt{ek}$ と乱数 m

出力： 共有の秘密鍵 K と暗号文 c

1: $(K, r) = \mathsf{G}\left(m \| \mathsf{H}(\mathtt{ek})\right)$ ▷ G はハッシュ関数

2: K-PKE 暗号化アルゴリズム（Algorithm 2）で，$(\mathtt{ek}, m, r)$ から暗号文 c を生成
▷ c は $(\mathtt{ek}, m, r)$ から一意的に生成されることに注意

3: **return** (K, c)

Algorithm 6 ML-KEM 鍵デカプセル化アルゴリズム [64, Algorithm 18]

入力： デカプセル化鍵 $\mathtt{dk} = (\mathtt{dk}_{\mathtt{PKE}}, \mathtt{ek}, \mathsf{H}(\mathtt{ek}), z)$ と暗号文 c

出力： 共有の秘密鍵 K

1: K-PKE 復号アルゴリズム（Algorithm 2）で，復号鍵 $\mathtt{dk}_{\mathtt{PKE}}$ と暗号文 c から，復号文 m' を生成

2: $(K', r') = \mathsf{G}\left(m' \| \mathsf{H}(\mathtt{ek})\right)$

3: $\bar{K} = \mathsf{J}(z \| c)$ ▷ J はハッシュ関数

4: K-PKE 暗号化アルゴリズム（Algorithm 2）で，$(\mathtt{ek}, m', r')$ から暗号文 c' を生成
▷ c' は $(\mathtt{ek}, m', r')$ から一意的に生成されることに注意

5: $c \neq c'$ の場合は，$K' = \bar{K}$ とおく

6: **return** K'

のステップ 1 で，c の復号文は $m' = m$ となる．これより，Algorithm 6 のステップ 2 は，鍵カプセル化アルゴリズムのステップ 1 と同じ組 (K, r) を生成する．また，Algorithm 6 のステップ 4 で，$(\mathtt{ek}, m', r') = (\mathtt{ek}, m, r)$ より，同じ暗号文 $c = c'$ を生成する．よって，Algorithm 6 は，鍵カプセル化アルゴリズムと同じ共有の秘密鍵

$$K' = K$$

を出力する．一方，入力する暗号文 c' が改竄されていれば，$c \neq c'$ なので，$K' = \bar{K} \neq K$ となる．

ML-KEM と CRYSTALS-Kyber との違い

ここでは，ML-KEM 方式と CRYSTALS-Kyber 方式 [13] の違いについてまとめておく（詳細は文献 [64, Appendix C] を参照）．

- Kyber 方式では，共有の秘密鍵 K は長さが可変な値として扱われていた．一方，ML-KEM 方式では，K の長さは 256 ビットに固定している．また，K は直接共通鍵として利用することも，秘密鍵生成の乱数シードとして用いることもできる．
- ML-KEM の鍵カプセル化と鍵デカプセル化のアルゴリズムでは，Kyber の第 3 ラウンド仕様書 [13] とは異なる藤崎-岡本変換を利用する．具体的には，ML-KEM 鍵カプセル化アルゴリズム（Algorithm 5）では共有する秘密鍵 K の導出において，暗号文 c のハッシュ値を含まない（具体的には，Algorithm 5 のステップ 1 で，入力する乱数 m とカプセル化鍵 ek のハッシュ値から K を生成）．また，ML-KEM 鍵デカプセル化アルゴリズムではその変更に合わせている（Algorithm 6 のステップ 2 を参照）．
- Kyber の第 3 ラウンドの仕様書 [13] では，鍵カプセル化アルゴリズム内の初期乱数 m は使う前にハッシュされる．具体的には，Kyber における鍵カプセル化アルゴリズムの 1 行目と 2 行目の間に，

  $$m \leftarrow H(m)$$

  の処理ステップがあった．一方，ML-KEM 鍵カプセル化アルゴリズム（Algorithm 5）では，m の生成には NIST 承認の乱数生成器が用いられるため，その処理は不必要で行わない．
- ML-KEM では，Kyber の第 3 ラウンドの仕様書 [13] にはなかった入力データの検証ステップを含む．例えば，ML-KEM 鍵カプセル化アルゴリズムでは，カプセル化キーを含むバイト配列が，モジュラー還元なしで q を法とする整数配列に正しくデコードされることを必要とする（ただし，上記の Algorithm 5 では，詳しく説明してない．入力データの整合性チェックに関しては，FIPS 仕様書 [64, §7] の Key pair check, Encapsulation key check, Decapsulation input check の段落をそれぞれ参照）．

2.2.3 ML-KEM における暗号パラメータ

表 2.1 に，ML-KEM における主な暗号パラメータ，対応する鍵・暗号文のサイズ，安全性レベルなどをまとめる．ただし，RBG（Random Bit Generator）強度は，乱数生成器が出力するビット列に対する攻撃困難性を表す．具体的には，(Module-)LWE の次元 $n = 256$ と剰余素数 $q = 3329$ は ML-KEM-512, -768, -1024 の 3 種類の暗号パラメータで共通であるが，主に 3 種類の階数パラメータ $k \in \{2, 3, 4\}$ により安全性レベルが異なる．特に，ML-KEM のパラメータ名は，

$$n \times k \in \{512, 768, 1024\}$$

表 2.1　ML-KEM における暗号パラメータ・安全性レベル・暗号文サイズなど（ただし，RBG（Random Bit Generator）強度は，乱数生成器が出力するビット列に対する攻撃困難性を表す）

ML-KEM パラメータ		ML-KEM-512	ML-KEM-768	ML-KEM-1024
暗号パラメータ	(n, q)	(256, 3329)	(256, 3329)	(256, 3329)
	k	2	3	4
	(η_1, η_2)	(3, 2)	(2, 2)	(2, 2)
	(d_u, d_v)	(10, 4)	(10, 4)	(11, 5)
デカプセル化（復号）失敗確率		$2^{-138.8}$	$2^{-164.8}$	$2^{-174.8}$
要求される RBG 強度（ビット）		128	192	256
NIST 安全性レベル [63]		レベル 1	レベル 3	レベル 5
サイズ（単位：バイト）	カプセル化鍵	800	1184	1568
	デカプセル化鍵	1632	2400	3168
	暗号文	768	1088	1568
	共有の秘密鍵	32	32	32

の値により名づけられている．また，各暗号パラメータ $(n, q, k, \eta_1, \eta_2, d_u, d_v)$ は下記のように選択されている（詳細は，文献 [13, Section 1.4] を参照）．

- ML-KEM 内の K-PKE 暗号アルゴリズム（Algorithm 2）において，256 ビットの平文を扱うので，n は 256 以上が必要であるため，$n = 256$ が選ばれている．
- 2.2.1 項で述べた NTT 処理を行うため，$n = 256 \mid q - 1$ を満たす小さな素数 q を選択する必要がある．この条件を満たす素数として，257 と 769 があるが，CCA 安全性に関する失敗確率が無視できない．具体的に，LWE ノイズを要因とする K-PKE の復号エラー率が無視できない大きさになり，CCA 安全性の帰着効率に影響する．これより，次に小さな素数である $q = 3329$ が選ばれている．
- 安全性レベル（ML-KEM の安全性を支える Module-LWE 問題の計算困難性）に応じて，階数 $k \in \{2, 3, 4\}$ を調整している（ML-KEM に対する具体的な攻撃計算量などは，後述の表 4.1 と 4.2 を参照）．
- その他のパラメータ η_1, η_2, d_u, d_v は安全性，暗号文サイズ，デカプセル化（復号）失敗確率のバランスを取るように選ばれている．特に，復号失敗を利用した攻撃（または，文献 [43] などの攻撃の改良）の脅威を避けるために，失敗確率が 2^{-128} より小さくなるようにパラメータの値が選ばれている．
  - η_1 は，秘密ベクトル $\boldsymbol{s}$ と公開鍵のノイズベクトル $\boldsymbol{e}$ と，暗号文における $\boldsymbol{y}$ の中心二項分布の大きさを定める．また，η_2 は，暗号文における $\boldsymbol{e}_1$ と e_2 のノイズを定める．具体的に，ML-KEM-512 では，$\eta_1 = 3 > \eta_2 = 2$ と設定されている（公開鍵に比べて，暗号文 $c = (\boldsymbol{u}, v)$ における $\boldsymbol{e}_1, e_2$ によるノイズは小さいが，$\boldsymbol{u}, v$ ともに Compress 関数

で圧縮するので，暗黙的にノイズが増大する）．一方，その他の ML-KEM-768, -1024 では $\eta_1 = \eta_2 = 2$ と設定されている．

– 上記では詳しく説明してないが，(d_u, d_v) は Compress/Decompress 関数，ByteEncode/ByteDecode 関数で使われるパラメータである．具体的には，(d_u, d_v) は暗号文サイズ圧縮のためのベクトル量子化に用いられる．

第 3 章

ML-KEM の安全性証明に関する調査結果

本章では，ML-KEM の安全性証明に関する調査結果を述べる．具体的には，古典ランダムオラクルモデル（ROM）と量子ランダムオラクルモデル（QROM）における Module-LWE 問題（定義 2.2）からの安全性帰着について述べる．

3.1 安全性仮定

ML-KEM の安全性を支える計算問題は，定義 2.2 の Module-LWE 問題である．ただし，ML-KEM における秘密 $\boldsymbol{s} \in R_q^k$ の成分多項式 s_i $(i = 1, \ldots, k)$ とノイズ多項式 $e \in R_q^k$ のすべての $\mathbb{Z}_q$ 係数は，中心二項分布 CBD_η $(\eta \in \{2, 3\})$ からサンプリングされる．ここで，B_η をすべての $\mathbb{Z}_q$ 係数が CBD_η からサンプルされる R_q 上のサンプル分布とする．ML-KEM の安全性証明においては，次の 2 種類のサンプルを識別する判定版の Module-LWE 問題を考える．

- 一様ランダムなサンプル $\left(\boldsymbol{a}_i, t_i\right) \leftarrow R_q^k \times R_q$
- Module-LWE サンプル $\left(\boldsymbol{a}_i, t_i\right) \in R_q^k \times R_q$（$\boldsymbol{a}_i \leftarrow R_q^k$ は一様ランダムにサンプルされ，式 (2.7) のように $t_i = \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + e_i \in R_q$ とする．ただし，$\boldsymbol{s} \leftarrow B_\eta^k$ はすべての Module-LWE サンプルに対し共通である一方，$e_i \leftarrow B_\eta$ は毎回選ばれる．）

より正確には，判定版の Module-LWE 問題に対する攻撃者（またはアルゴリズム）$\mathscr{A}$ に対して，

$$
\begin{aligned}
&\mathbf{Adv}^{\mathrm{mlwe}}_{m,k,\eta}(\mathscr{A}) \\
&= \left|\Pr\left[b' = 1 :\ \boldsymbol{A} \leftarrow R_q^{m \times k};\ (\boldsymbol{s}, \boldsymbol{e}) \leftarrow B_\eta^k \times B_\eta^m;\ \boldsymbol{t} = \boldsymbol{s}\boldsymbol{A}^\top + \boldsymbol{e};\ b' \leftarrow \mathscr{A}(\boldsymbol{A}, \boldsymbol{t})\right]\right. \\
&\qquad \left. - \Pr\left[b' = 1 :\ \boldsymbol{A} \leftarrow R_q^{m \times k};\ \boldsymbol{t} \leftarrow R_q^m;\ b' \leftarrow \mathscr{A}(\boldsymbol{A}, \boldsymbol{t})\right]\right| \geq 0
\end{aligned}
\tag{3.1}
$$

と定める．ただし，m は Module-LWE サンプル数で，$b' \leftarrow \mathscr{A}(\boldsymbol{A}, \boldsymbol{t})$ は攻撃者 $\mathscr{A}$ による出力結果とする．以下の ML-KEM の安全性証明においては，暗号文の形 (2.16) から，$m = k + 1$ の場合の $\mathbf{Adv}^{\mathrm{mlwe}}_{k+1,k,\eta}(\mathscr{A})$ が十分 0 に近い Module-LWE 仮定の下で考える．

3.2 ROM における Module-LWE からのタイトな帰着

次の定理が示すように，ランダムオラクルモデル（ROM）において，Module-LWE 仮定の下で，ML-KEM の基盤である公開鍵暗号方式 K-PKE（Algorithms 1, 2, 3 で構成）はタイトな IND-CPA 安全である（ただし，[13, Theorem 1, §4.3.1] において，“Kyber.CPAPKE” を “ML-KEM.K-PKE” に変更）．その証明は，Module-LWE 仮定の下で、公開鍵と暗号文が擬似ランダムであることから従う．

定理 3.1 ([13], Theorem 1)**.** XOF と G はランダムオラクルとする．このとき，任意の攻撃者 $\mathscr{A}$ に対して，$\mathscr{A}$ と同程度の処理能力を持つ攻撃者 $\mathscr{B}, \mathscr{C}$ が存在して，

$$\mathbf{Adv}^{\mathrm{cpa}}_{\text{ML-KEM.K-PKE}}(\mathscr{A}) \leq 2 \cdot \mathbf{Adv}^{\mathrm{mlwe}}_{k+1,k,\eta}(\mathscr{B}) + \mathbf{Adv}^{\mathrm{prf}}_{\mathrm{PRF}}(\mathscr{C})$$

を満たす．ただし，$\mathbf{Adv}^{\mathrm{cpa}}_{\text{ML-KEM.K-PKE}}$ と $\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{PRF}}$ はそれぞれ式 (3.1) と同じように定める．

ML-KEM（Algorithms 4, 5, 6 で構成）は K-PKE の（微調整した）藤崎-岡本変換から得られる．2 つのハッシュ関数 G,H をランダムオラクルとモデル化した仮定の下で，次の定理が示すように ML-KEM は IND-CCA2 安全である（ただし，[13, Theorem 2, §4.3.1] において，“Kyber.CCAKEM” を “ML-KEM” に変更した）．特に，次の定理におけるタイトな上界は，上記の定理 3.1 と文献 [42] の結果から得られる．

定理 3.2 ([13], Theorem 2)**.** XOF と 2 つのハッシュ関数 G, H はランダムオラクルとする．このとき，XOF，G, H のランダムオラクルに高々 q_{RO} 回問い合わせることができる古典攻撃者 $\mathscr{A}$ に対して，$\mathscr{A}$ と同程度の処理能力を持つ攻撃者 $\mathscr{B}, \mathscr{C}$ が存在して，

$$\mathbf{Adv}^{\mathrm{cca}}_{\text{ML-KEM}}(\mathscr{A}) \leq 2 \cdot \mathbf{Adv}^{\mathrm{mlwe}}_{k+1,k,\eta}(\mathscr{B}) + \mathbf{Adv}^{\mathrm{prf}}_{\mathrm{PRF}}(\mathscr{C}) + 4q_{RO} \cdot \delta$$

を満たす．ただし，$\mathbf{Adv}^{\mathrm{cca}}_{\text{ML-KEM}}$ は式 (3.1) と同じように定め，δ は ML-KEM の基盤である K-PKE における復号失敗確率とする．

3.3 QROM における Module-LWE からのノンタイトな帰着

次の定理が示すように，量子ランダムオラクルモデル（QROM）において，Module-LWE 仮定の下で，ML-KEM の基盤である公開鍵暗号方式 K-PKE が IND-CPA 安全であれば，ML-KEM は IND-CCA2 安全である（詳細は文献 [42, 69] を参照．定理 3.2 と同じように，[13, Theorem 3] において，“Kyber.CCAKEM” を “ML-KEM” に変更した）．

定理 3.3 ([13], Theorem 3)**.** XOF と 2 つのハッシュ関数 G,H は量子ランダムオラクルとする．このとき，XOF，G, H の量子ランダムオラクルに高々 q_{RO} 回問い合わせることができる量子攻撃者

$\mathscr{A}$ に対して，$\mathscr{A}$ と同程度の処理能力を持つ量子攻撃者 $\mathscr{B}, \mathscr{C}$ が存在して，

$$\mathbf{Adv}^{\mathrm{cca}}_{\text{ML-KEM}}(\mathscr{A}) \leq 4q_{RO} \cdot \sqrt{\mathbf{Adv}^{\mathrm{mlwe}}_{k+1,k,\eta}(\mathscr{B})} + \mathbf{Adv}^{\mathrm{prf}}_{\mathrm{PRF}}(\mathscr{C}) + 8q_{RO}^2 \cdot \delta \tag{3.2}$$

を満たす．

上記の定理における上界はノンタイトである．ゆえに，上記の定理は，量子ランダムオラクルモデルにおける ML-KEM の IND-CCA 安全性について漸近的な示唆を与えるだけである．

一方，標準的ではない仮定の下で，量子ランダムオラクルモデルにおいて，よりタイトな上界を得ることができる．具体的には，ML-KEM の基盤である公開鍵暗号方式 K-PKE の確定版が，量子ランダムオラクルモデルにおいて擬似ランダムと仮定する．つまり，暗号化時に利用するランダムコインが，$r = G(m)$ のように平文 m によって確定的に決定すると仮定する．また，確定版の暗号化における擬似ランダム性とは，ランダムに選ばれた平文の暗号文 (c_1, c_2) が，一様ランダムな $(\boldsymbol{u}, v)$ に対する暗号文 $\big(\mathsf{Compress}_q(\boldsymbol{u}), \mathsf{Compress}(v)\big)$ と計算量的に識別困難であることをいう．このとき，式 (3.2) よりタイトな上界

$$\mathbf{Adv}^{\mathrm{cca}}_{\text{ML-KEM}}(\mathscr{A}) \leq 2\mathbf{Adv}^{\mathrm{mlwe}}_{k+1,k,\eta}(\mathscr{B}) + \mathbf{Adv}^{\mathrm{pr}}_{\text{DK-PKE}}(\mathscr{C}) + \mathbf{Adv}^{\mathrm{prf}}_{\mathrm{PRF}}(\mathscr{D}) + 8q_{RO}^2 \cdot \delta$$

が成り立つ [13, §4.3.2]．ただし，DK-PKE は K-PKE の確定版とし，“pr” はその擬似ランダム性（Pseudo-Randomness）とする．

3.4 Module-LWE 問題以外への攻撃

上記の ML-KEM に対する安全性証明から，Module-LWE 問題を解くことなく ML-KEM を攻撃する方法として次がある．

- 安全性証明の帰着仮定である共通プリミティブを攻撃する．
- ML-KEM におけるデカプセル化（復号）失敗確率におけるノンタイト性を利用する．

3.4.1 共通鍵暗号プリミティブへの攻撃可能性について

前章では詳しく説明しなかったが，次のように共通鍵暗号プリミティブは FIPS 202 標準 [62] からの関数でインスタント化する．

- `XOF`：SHAKE-128
- `H`：SHA3-256
- `G`：SHA3-512
- $\texttt{PRF}(s, b)$：SHAKE-256$(s \| b)$（擬似ランダム関数 PRF: PseudoRandom Function）
- `KDF`：SHAKE-256

これらの共通の構成要素は，Keccak（SHA-3）から導出された関数群でインスタント化される．具

体的には，乱数シード ρ から一意的に $\boldsymbol{A}$ を展開する際は，SHAKE-128 を利用して一様擬似ランダムな成分をもつ行列を出力し，安全性を支える格子問題におけるバックドアを発生させない．また，ノイズ（誤差）生成時には，秘密と公開の入力を結合し，その結合値を SHAKE-256 に入力することで，安全な擬似ランダムな関数を構成する．これらの SHAKE の性質を破ることは，SHAKE の暗号解析における重大なブレークスルーで，KEM の中の SHAKE を他の擬似ランダム関数に置き換える必要があるが，現状そのような攻撃は現実的ではない．安全性証明では，SHAKE-128, SHA3-256, SHA3-512 をランダムオラクルとしてモデル化している．`XOF`, `G`, `H` の関数の SHAKE と SHA3 によるインスタンスを利用する攻撃は，一般に Keccak またはランダムオラクル証明における重大なブレークスルーであり，そのような攻撃は現状では現実的ではない．

3.4.2 復号失敗を利用した攻撃の可能性について

K-PKE，ML-KEM ともに復号エラーが起きた場合には，plaintext checking, key mismatch 等のオラクルを構成し，それをベースに鍵復元を行うサイドチャネル攻撃が数多く提案されている（例えば，文献 [66, 73, 77] を参照）．しかし，表 2.1 に示すように，ML-KEM における復号失敗確率 δ の見積もりは 2^{-128} 未満であり，実用的には無視できるほど小さい（IETF ドラフト [44, Section 8.1] も参照）．一方で，復号失敗を起こすような正規の暗号文（weak ciphertexts）を効率的に探索する手法も提案されているが [36]，ML-KEM の安全性に影響を与えるほど実用的なものではない．

第 4 章

Module 構造の考慮の有無に応じた計算量評価に関する調査結果

本章では，ML-KEM の安全性を支える Module-LWE 問題を解く攻撃法とその計算量見積もりを示す．具体的には，Module-LWE 問題を $\mathbb{Z}_q$ 上の LWE 問題に帰着し，さらに $\mathbb{Z}_q$ 上の LWE 問題を格子問題に帰着する．また，帰着した格子問題を解く現時点で最良とされる格子アルゴリズムの計算量評価に基づき，ML-KEM の暗号パラメータに対する攻撃計算量の見積もりを示す．本章では，特に断らない限り，すべてのベクトルは行ベクトルで統一する．また，$\mathbb{R}$ 成分のベクトル $\boldsymbol{v} = (v_1, \ldots, v_d) \in \mathbb{R}^d$ のノルムは，すべて Euclid ノルム

$$\|\boldsymbol{v}\| = \sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle} = \sqrt{v_1^2 + \cdots + v_d^2}$$

とする．

4.1 Module-LWE 問題の $\mathbb{Z}_q$ 上の LWE 問題への帰着

本節では，定義 2.2 の Module-LWE 問題を，通常の $\mathbb{Z}_q$ 上の LWE 問題の形に帰着できることを述べる（数学的には，$\mathbb{Z}_q$ 加群としての同型 $R_q^k \simeq (\mathbb{Z}_q^n)^k$ を明示的に与えることに相当する）．Module-LWE 問題における剰余環 $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ において，変数 X に関するベクトルを

$$\boldsymbol{X} = \left(1, X, X^2, \ldots, X^{n-1}\right) \in R_q^n$$

とおく．このとき，R_q の任意の元 $f(X) = f_0 + f_1 X + \cdots + f_{n-1} X^{n-1}$ とその係数ベクトル $\boldsymbol{f} = (f_0, f_1, \ldots, f_{n-1}) \in \mathbb{Z}_q^n$ に対して，

$$f(X) = \boldsymbol{f}\boldsymbol{X}^\top \in R_q$$

が成り立つ（右辺は 2 つのベクトル $\boldsymbol{f}, \boldsymbol{X}$ の内積）．また，$f(X)$ の係数ベクトル $\boldsymbol{f}$ に対する回転操作を

$$\mathrm{rot}(\boldsymbol{f}) \coloneqq \left(-f_{n-1}, f_0, f_1 \ldots, f_{n-2}\right) \in \mathbb{Z}_q^n \tag{4.1}$$

と定める．このとき，(1 回の）回転ベクトル $\mathrm{rot}(\boldsymbol{f})$ は，$f(X)$ に X を乗じた多項式 $Xf(X)$ の係数ベクトルである．より一般に，X^i を乗じた多項式 $X^i f(X)$ の係数ベクトルは，i 回の回転操作を施したベクトル $\mathrm{rot}^i(\boldsymbol{f}) = \mathrm{rot}\left(\mathrm{rot}(\cdots \mathrm{rot}\left(\boldsymbol{f}\right))\right)$ で与えられる．特に，環 R_q において $X^n = -1$ なので，n 回の回転ベクトル $\mathrm{rot}^n(\boldsymbol{f})$ は $-\boldsymbol{f}$ に一致する（つまり，$\mathrm{rot}^n(\boldsymbol{f}) = -\boldsymbol{f}$ である）．

定義 2.2 と同じように，Module-LWE 問題の秘密を $\boldsymbol{s}(X) = \big(s_1(X), s_2(X), \ldots, s_k(X)\big) \in R_q^k$ と表す．また，式 (2.7) の Module-LWE サンプルの組 $\big(\boldsymbol{a}(X), t(X)\big) \in R_q^k \times R_q$ に対して，$\boldsymbol{a}(X) = (a_1(X), a_2(X), \ldots, a_k(X))$ と表す．このとき，環 R_q において，

$$t(X) = \sum_{i=1}^{k} a_i(X) s_i(X) + e(X)$$

が成り立つことを意味する．さらに，R_q の元 $s_i(X), a_i(X)$ $(i = 1, \ldots, k)$ と $e(X), t(X)$ にそれぞれ対応する係数ベクトルを $\boldsymbol{s}_i, \boldsymbol{a}_i$ $(i = 1, 2, \ldots, k)$ と $\boldsymbol{e}, \boldsymbol{t} \in \mathbb{Z}_q^n$ とする．このとき，

$$\begin{aligned}
\boldsymbol{t}\boldsymbol{X}^\top = t(X) &= \sum_{i=1}^{k} a_i(X) s_i(X) + e(X) \\
&= \sum_{i=1}^{k} \boldsymbol{s}_i \boldsymbol{X}^\top a_i(X) + \boldsymbol{e}\boldsymbol{X}^\top \\
&= \sum_{i=1}^{k} \boldsymbol{s}_i \begin{pmatrix} a_i(X) \\ X a_i(X) \\ \vdots \\ X^{n-1} a_i(X) \end{pmatrix} + \boldsymbol{e}\boldsymbol{X}^\top = \sum_{i=1}^{k} \boldsymbol{s}_i \begin{pmatrix} \boldsymbol{a}_i \boldsymbol{X}^\top \\ \mathrm{rot}(\boldsymbol{a}_i)\boldsymbol{X}^\top \\ \vdots \\ \mathrm{rot}^{n-1}(\boldsymbol{a}_i)\boldsymbol{X}^\top \end{pmatrix} + \boldsymbol{e}\boldsymbol{X}^\top \\
&= \left(\sum_{i=1}^{k} \boldsymbol{s}_i \boldsymbol{A}_i + \boldsymbol{e} \right) \boldsymbol{X}^\top
\end{aligned}$$

が成り立つ（各 $j = 0, 1, \ldots, n-1$ に対し，$X^j a_i(X) = \mathrm{rot}^j(\boldsymbol{a}_i)\boldsymbol{X}^\top$ であることに注意）．ただし，各 $i = 1, 2, \ldots, k$ に対して，

$$\boldsymbol{A}_i = \begin{pmatrix} \boldsymbol{a}_i \\ \mathrm{rot}(\boldsymbol{a}_i) \\ \vdots \\ \mathrm{rot}^{n-1}(\boldsymbol{a}_i) \end{pmatrix} \in \mathbb{Z}_q^{n \times n}$$

とする．これより，変数ベクトル $\boldsymbol{X}$ の成分による集合 $\{1, X, X^2, \ldots, X^{n-1}\}$ は自由 $\mathbb{Z}_q$ 加群 R_q の基底（つまり，$R_q = \mathbb{Z}_q \oplus \mathbb{Z}_q X \oplus \cdots \oplus \mathbb{Z}_q X^{n-1}$）なので，

$$\boldsymbol{t} \equiv \sum_{i=1}^{k} \boldsymbol{s}_i \boldsymbol{A}_i + \boldsymbol{e} \pmod{q} \iff \boldsymbol{t} \equiv (\boldsymbol{s}_1 \mid \cdots \mid \boldsymbol{s}_k) \begin{pmatrix} \boldsymbol{A}_1 \\ \hline \vdots \\ \hline \boldsymbol{A}_k \end{pmatrix} + \boldsymbol{e} \pmod{q} \tag{4.2}$$

が成り立つ．各 $j = 0, 1, \ldots, n-1$ に対して，両辺の X^j の係数に対応する列を比較することで，次元 nk の $\mathbb{Z}_q$ 上の LWE サンプルの関係式が得られる．具体的には，$\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k$ の連結ベクトル

$$\boldsymbol{s} = (\boldsymbol{s}_1 \mid \cdots \mid \boldsymbol{s}_k) \in (\mathbb{Z}_q^n)^k \tag{4.3}$$

を秘密とした n 個の $\mathbb{Z}_q$ 上の nk 次元の LWE サンプル

$$\big(\boldsymbol{a}_j, t_j\big), \quad t_j \equiv \langle \boldsymbol{a}_j, \boldsymbol{s} \rangle + e_j \pmod{q} \quad (j = 0, 1, \ldots, n-1) \tag{4.4}$$

が得られる．ただし，$\boldsymbol{a}_j$ は式 (4.2) の右辺の $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_k$ を縦に連結した $nk \times n$ 行列の $j+1$ 列目のベクトル，t_j, e_j はそれぞれ $t(X), e(X) \in R_q$ の X^j 係数とする．また，攻撃者有利の観点から，任意に Module-LWE サンプルを生成し，その中から $\mathbb{Z}_q$ 上の LWE サンプルの m 個を行列表示

$$\big(\boldsymbol{A}, \boldsymbol{t}\big), \quad \boldsymbol{t} \equiv \boldsymbol{s}\boldsymbol{A}^\top + \boldsymbol{e} \pmod{q} \tag{4.5}$$

しておく．ただし，$\boldsymbol{A} \in \mathbb{Z}_q^{m \times nk}$，$\boldsymbol{t}, \boldsymbol{e} \in \mathbb{Z}_q^m$ とする．また，$\mathbb{Z}_q$ 上の LWE サンプル数 m は，一般に攻撃が最も有利となるものを想定する．この行列表示により，行列 $\boldsymbol{A}$ の行ベクトル $\boldsymbol{a}_j \in \mathbb{Z}_q^{nk}$ と秘密ベクトル $\boldsymbol{s} \in \mathbb{Z}_q^{nk}$ の内積 $\langle \boldsymbol{a}_j, \boldsymbol{s} \rangle$ について，式 (4.4) の関係が成り立つ．特に，ML-KEM では，秘密ベクトル $\boldsymbol{s}$ とノイズベクトル $\boldsymbol{e}$ のすべての $\mathbb{Z}_q$ 成分は，$\eta \in \{2, 3\}$ に対する $\mathbb{Z}_q$ 上の中心二項分布 CBD_η からサンプリングされる（ML-KEM における中心二項分布 CBD_η の具体的な計算手順については，2.2.2 項を参照）．

4.2 格子の基礎と格子アルゴリズム

本節では，格子の基礎と格子問題を解くための格子アルゴリズムについて簡単にまとめておく．

4.2.1 格子と基底

整数 $d \geq 2$ に対して，d 次元実ベクトル空間 $\mathbb{R}^d$ の一次独立な d 個のベクトル $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_d$ の整数係数の線形結合全体の集合

$$L = \left\{ \sum_{i=1}^{d} x_i \boldsymbol{b}_i \;\middle|\; x_i \in \mathbb{Z},\ 1 \leq i \leq d \right\} = \mathbb{Z}\boldsymbol{b}_1 + \cdots + \mathbb{Z}\boldsymbol{b}_d$$

を（完全階数の）d 次元の**格子**（lattice）とよぶ．特に，格子 L は $\mathbb{R}^d$ の（離散）加法部分群である．また，格子 L を生成する一次独立な d 個のベクトルの組 $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_d\}$ を**基底**（basis）とよび，各 $\boldsymbol{b}_i$ $(i = 1, 2, \ldots, d)$ を**基底ベクトル**（basis vector）とよぶ．さらに，行ベクトルで表した d 個の基底ベクトル $\boldsymbol{b}_i \in \mathbb{R}^d$ $(i = 1, 2, \ldots, d)$ を行として持つ $d \times d$ 行列

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_d \end{pmatrix} \in \mathbb{R}^{d \times d}$$

を格子 L の**基底行列**（basis matrix）とよぶ．2 次元以上の格子を生成する互いに異なる基底は無限に存在する．正確には，同じ格子を生成する 2 つの基底行列 $\boldsymbol{B}_1, \boldsymbol{B}_2$ に対し $\boldsymbol{B}_2 = \boldsymbol{V}\boldsymbol{B}_1$ を満たす $d \times d$ のユニモジュラ行列 $\boldsymbol{V}$ が必ず存在する．ただし，整数行列でその行列式が $\det(\boldsymbol{V}) = \pm 1$ である正方行列 $\boldsymbol{V}$ を**ユニモジュラ行列**（unimodular matrix）とよぶ．これより，2 次以上のユニ

モジュラ行列は無限に存在するため，2 次元以上の格子は互いに異なる基底を無限にもつ．また，格子 L の任意の基底行列 $\boldsymbol{B}$ を用いて，L の**体積**（volume）を

$$\mathrm{vol}(L) \coloneqq |\det(\boldsymbol{B})| > 0$$

と定める．ここで，互いに異なる格子基底行列はユニモジュラ行列で結ばれるので，格子の体積は基底行列の選び方には依存しない．格子 L の**第 1 逐次最小**（first successive minimum）は，L 上の最短な非零ベクトルのノルムを指し，$\lambda_1(L)$ の記号で表す．

双対格子とその基底

d 次元実ベクトル空間 $\mathbb{R}^d$ の完全階数の格子 $L \subset \mathbb{R}^d$ に対して，集合

$$\widehat{L} \coloneqq \left\{ \boldsymbol{x} \in \mathbb{R}^d \mid \langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathbb{Z},\ \forall \boldsymbol{y} \in L \right\}$$

を L の**双対格子**（dual lattice）とよぶ．また，L の基底行列 $\boldsymbol{B} \in \mathbb{R}^{d \times d}$ に対して，

$$\widehat{\boldsymbol{B}} \coloneqq \left(\boldsymbol{B}^\top\right)^{-1} = \left(\boldsymbol{B}^{-1}\right)^\top$$

は双対格子 $\widehat{L}$ の基底行列となる（つまり，$\widehat{\boldsymbol{B}}$ の d 個の行ベクトルは，$\mathbb{R}$ 上一次独立で，$\widehat{L}$ を生成する）．この $\widehat{\boldsymbol{B}}$ を**双対基底行列**（dual basis matrix）とよぶ．また，単位行列 $\boldsymbol{I}_d$ に対して，明らかに $\boldsymbol{B}\widehat{\boldsymbol{B}}^\top = \boldsymbol{I}_d$ を満たすので，L の体積とその双対格子 $\widehat{L}$ の体積について

$$\mathrm{vol}(L) \times \mathrm{vol}(\widehat{L}) = 1 \tag{4.6}$$

が成り立つ．

Gauss のヒューリスティックと格子の第 1 次逐次最小

d 次元実ベクトル空間 $\mathbb{R}^d$ 内の完全階数の格子 L に対して，体積 $\mathrm{vol}(C)$ を持つ任意の集合 $C \subseteq \mathbb{R}^d$ との共通部分に含まれる格子ベクトルの個数はおおよそ

$$\#(L \cap C) \approx \frac{\mathrm{vol}(C)}{\mathrm{vol}(L)}$$

であることが期待できる．これを **Gauss のヒューリスティック**（Gaussian Heuristic）とよぶ．特に，集合 C として，格子 L の第 1 次逐次最小 $\lambda_1(L)$ を半径に持ち中心が零ベクトル $\mathbf{0} \in \mathbb{R}^d$ の d 次元開球 $\mathcal{B}(\mathbf{0}, \lambda_1(L))$ をとると，おおよそ

$$\frac{\mathrm{vol}(C)}{\mathrm{vol}(L)} \approx \#(L \cap C) \approx 1$$

と期待できる．さらに，d 次元単位球の体積を ω_d とすると $\mathrm{vol}(C) = \omega_d \times \lambda_1(L)^d$ が成り立つので，

$$\lambda_1(L) \approx \left(\frac{\mathrm{vol}(L)}{\omega_d}\right)^{1/d} \sim \sqrt{\frac{d}{2\pi e}}\mathrm{vol}(L)^{1/d} \tag{4.7}$$

が成り立つと期待できる．ただし，記号 $\sim$ は $d \to \infty$ のとき両辺の比が 1 に収束することを意味する．また，d 次元単位球の体積 ω_d に関しては，ガンマ関数 $\Gamma(x)$ を用いると

$$\omega_d = \frac{\pi^{d/2}}{\Gamma\left(1+\frac{d}{2}\right)} \sim \left(\frac{2\pi e}{d}\right)^{d/2} \tag{4.8}$$

が成り立つことが知られている．

4.2.2 格子問題と格子基底簡約

格子問題は格子に関する計算問題で，以下で最も代表的な 2 つの格子問題である**最短ベクトル問題**（Shortest Vector Problem, SVP）と**最近ベクトル問題**（Closest Vector Problem, CVP）を述べておく．これらの格子問題の求解困難性は，格子暗号の根本的な安全性の根拠となっている．

定義 4.1 (最短ベクトル問題, SVP)**.** 格子 L の基底 $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d\}$ が与えられたとき，格子の最短な非零ベクトル $\boldsymbol{v} \in L$ を見つけよ．つまり，$\|\boldsymbol{v}\| = \lambda_1(L)$ を満たす格子ベクトル $\boldsymbol{v} \in L$ を見つけよ．

定義 4.2 (最近ベクトル問題, CVP)**.** 格子 L の基底 $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d\}$ と $\boldsymbol{t} \in \mathbb{R}^d \setminus L$ が与えられたとき，$\boldsymbol{t}$ に最も近い格子ベクトル $\boldsymbol{v} \in L$ を見つけよ．つまり，$\boldsymbol{t}$ との距離 $\|\boldsymbol{t} - \boldsymbol{v}\|$ を最小にする格子ベクトル $\boldsymbol{v} \in L$ を見つけよ．

上述の SVP・CVP や（Module-）LWE などの格子問題を解くのに必須の格子アルゴリズムとして，**格子基底簡約**（lattice basis reduction）がある．格子基底簡約は，与えられた格子 $L \subset \mathbb{R}^d$ の基底から，各ベクトル $\boldsymbol{b}_i$ が短く・互いのベクトルが直交に近い L の新しい基底 $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_d\}$ を見つける操作（アルゴリズム）である．明確な基準があるわけではないが，このような基底を「簡約基底」（reduced basis）または「良い基底」（good basis）とよぶ．具体的な基底簡約のアルゴリズムを紹介するために，**Gram-Schmidt の直交化**（Gram-Schmidt orthogonalization）を説明しておく．格子 L の基底 $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_d\}$ の Gram-Schmidt ベクトル $\boldsymbol{b}_1^\star, \boldsymbol{b}_2^\star, \ldots, \boldsymbol{b}_d^\star$ は，次のように再帰的に定める．

$$\begin{cases} \boldsymbol{b}_1^\star \coloneqq \boldsymbol{b}_1, \\ \boldsymbol{b}_i^\star \coloneqq \boldsymbol{b}_i - \displaystyle\sum_{j=1}^{i-1} \mu_{i,j} \boldsymbol{b}_j^\star, \quad \mu_{i,j} = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^\star \rangle}{\|\boldsymbol{b}_j^\star\|^2} \quad (i = 2, 3, \ldots, d). \end{cases}$$

Gram-Schmidt ベクトルについて，直交性 $\langle \boldsymbol{b}_i^\star, \boldsymbol{b}_j^\star \rangle = 0$ $(i \neq j)$ と，格子の体積について

$$\mathrm{vol}(L) = \prod_{i=1}^{d} \|\boldsymbol{b}_i^\star\| \tag{4.9}$$

が成り立つ．また，各 $2 \leq \ell \leq d$ に対して，$\mathbb{R}^d$ から $\mathbb{R}$-ベクトル空間 $\langle \boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{\ell-1} \rangle_{\mathbb{R}}$ の直交補空間への直交射影を

$$\pi_\ell : \mathbb{R}^d \longrightarrow \langle \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\ell-1} \rangle_{\mathbb{R}}^{\perp} = \langle \boldsymbol{b}_\ell^\star, \ldots, \boldsymbol{b}_d^\star \rangle_{\mathbb{R}}, \quad \pi_\ell(\boldsymbol{x}) = \sum_{i=\ell}^{d} \frac{\langle \boldsymbol{x}, \boldsymbol{b}_i^\star \rangle}{\|\boldsymbol{b}_i^\star\|^2} \boldsymbol{b}_i^\star$$

とする．また，便宜上 π_1 は恒等写像としておく．さらに，$(d-\ell+1)$ 個の一次独立な射影ベクトル $\pi_\ell(\boldsymbol{b}_\ell),\ldots,\pi_\ell(\boldsymbol{b}_d)$ で生成させる格子を $\pi_\ell(L)$ と記し，L の**射影格子**（projected lattice）とよぶ．$\pi_\ell(\boldsymbol{b}_\ell),\ldots,\pi_\ell(\boldsymbol{b}_d)$ の Gram-Schmidt ベクトルは $\boldsymbol{b}_\ell^\star,\boldsymbol{b}_{\ell+1}^\star,\ldots,\boldsymbol{b}_d^\star$ であるので，式 (4.9) と同様に，射影格子 $\pi_\ell(L)$ の体積は $\prod_{i=\ell}^{d}\|\boldsymbol{b}_i^\star\|$ で与えられる．

以下で，代表的な 2 つの格子基底簡約アルゴリズムを紹介しておく．

Lentra-Lenstra-Lovász（LLL）基底簡約

LLL 基底簡約 [53] は，簡約パラメータ $\frac{1}{4}<\delta<1$ に対して，次の 2 つの条件を満たす格子の基底 $\{\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\boldsymbol{b}_d\}$（LLL 簡約基底という）を見つけるアルゴリズムである．

(i) サイズ簡約されている．つまり，すべての Gram-Schmidt 係数が $|\mu_{i,j}|\leq\frac{1}{2}$ $(1\leq j<i\leq d)$ を満たす．
(ii) Lovász 条件を満たす．つまり，$\delta\|\boldsymbol{b}_{k-1}^\star\|^2\leq\|\pi_{k-1}(\boldsymbol{b}_k)\|^2$ $(k=2,3,\ldots,d)$ を満たす．

入力基底に対して，Lovász 条件が成り立たないとき，LLL 基底簡約内で隣り合う基底ベクトル $\boldsymbol{b}_{k-1},\boldsymbol{b}_k$ の交換を行い，(i) と (ii) の両方の条件を満たす基底（つまり，LLL 簡約基底）を見つける．また，LLL 基底簡約の時間計算量は，入力する基底が生成する格子の次元 d に関して多項式時間である．

Block Korkine-Zolotarev（BKZ）基底簡約

BKZ 基底簡約 [71] は，ブロックサイズ $\beta\geq 2$ による LLL 基底簡約の一般化である（$\beta=2$ の場合は LLL 基底簡約と本質的に同じ）．LLL 基底簡約に比べ，BKZ 基底簡約でより良い簡約基底を見つけることができるが，その計算量は β に関して指数時間である．具体的には，BKZ 基底簡約に入力するブロックサイズ β を増やすごとに，実行時間が非常に遅くなる一方，より短い基底ベクトルを出力する．より具体的には，ブロックサイズ $2\leq\beta\leq d$ に対して，BKZ 基底簡約は次の 2 つの条件を満たす格子 L の基底 $\{\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\boldsymbol{b}_d\}$（$\beta$-BKZ 簡約基底）を見つける．

(i) LLL 基底簡約と同様，基底はサイズ簡約されている．
(ii) すべての $1\leq j\leq d$ に対して，$\|\boldsymbol{b}_j^\star\|=\lambda_1(L_{[j:k]})$ を満たす．ただし，$k=\min(j+\beta-1,d)$ とし，$(k-j+1)$ 個の射影ベクトル $\pi_j(\boldsymbol{b}_j),\pi_j(\boldsymbol{b}_{j+1}),\ldots,\pi_j(\boldsymbol{b}_k)$ で生成される L のブロック射影格子を $L_{[j:k]}$ とする（ブロック射影格子 $L_{[j:k]}$ は，射影格子 $\pi_j(L)$ の部分格子である）．

入力基底に対して，BKZ 基底簡約のアルゴリズム内ではブロック射影格子 $L_{[j:k]}$ 上の SVP オラクルを繰り返しよびだし，(i) と (ii) の両方の条件を満たす基底（つまり，β-BKZ 簡約基底）を見つける．以下で述べるように，BKZ 基底簡約の出力基底と計算量はブロックサイズ β に依存する．

4.2.3 BKZ 基底簡約の出力基底と計算量

これまで BKZ 2.0 [26] や pump & jump BKZ（pnj-BKZ）[5] などの効率的な BKZ 基底簡約の改良アルゴリズムが提案され，格子に基づく暗号技術の安全性評価において頻繁に利用されている．ここでは，BKZ 基底簡約の出力基底と計算量評価の見積もりについて述べる (詳細は [2] を参照) .

BKZ 基底簡約の出力基底の見積もり

格子基底簡約アルゴリズムが出力する簡約基底の「良さ」を測る指標として Hermite 因子がある．d 次元格子 $L \subset \mathbb{R}^d$ の基底が与えられたとき，基底簡約アルゴリズムが出力する最短な基底ベクトル（多くの場合は第 1 基底ベクトル）を $\boldsymbol{b} \in L$ とする．このとき，その基底簡約アルゴリズムの **Hermite 因子**（Hermite factor）を

$$\gamma \coloneqq \frac{\|\boldsymbol{b}\|}{\mathrm{vol}(L)^{1/d}}$$

と定める．これは，Hermite 因子が小さいほど，その基底簡約アルゴリズムはより短い基底ベクトルを出力することを意味する．100 以上の高次元のランダム格子に対して，LLL や BKZ などの基底簡約アルゴリズムの Hermite 因子の d 乗根 $\gamma^{1/d}$（つまり，root Hermite 因子）は定数に収束することが実験的に知られている [38]．特に，高い次元 d のランダム格子において，高いブロックサイズ $\beta \geq 50$ に対する BKZ 基底簡約の root Hermite 因子はおおよそ

$$\gamma^{\frac{1}{d}} \approx \left(\omega_\beta^{-\frac{1}{\beta}}\right)^{\frac{1}{\beta-1}} \approx \left(\frac{\beta}{2\pi e}(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}} =: \delta_\beta \tag{4.10}$$

に従うことが実験的に知られている [26, 78]．ただし，ω_β は β 次元の単位球の体積とする（式 (4.8) を参照）．例えば，$\beta = 85$ で $\gamma^{1/d} \approx 1.01$ となる．この root Hermite 因子の見積もりを用いて，格子に基づく暗号技術の安全性評価対象の格子問題の求解で必要となる BKZ のブロックサイズ β を求めることができる．より具体的には，$\beta \geq 50$ かつ $\beta \ll d$ を満たす大きなブロックサイズ β に対して，d 次元のランダム格子 L の β-BKZ 簡約基底 $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d\}$ の Gram-Schmidt ベクトル $\boldsymbol{b}_1^\star, \ldots, \boldsymbol{b}_d^\star$ のノルムはおおよそ

$$\|\boldsymbol{b}_i^\star\| \approx \delta_\beta^{d-2i+1} \cdot \mathrm{vol}(L)^{\frac{1}{d}} \quad (i = 1, 2, \ldots, d) \tag{4.11}$$

に従うことが実験的に知られている（例えば，文献 [4, 25, 26, 78] を参照）．ただし，δ_β は式 (4.10) の値とする．また，これは Gram–Schmidt ベクトルの対数ノルム $\log \|\boldsymbol{b}_i^\star\|$ $(i = 1, 2, \ldots, d)$ が直線上に並ぶという**幾何級数仮定**（Geometric Series Assumption, GSA）[70] と Gauss のヒューリスティックの下で得られる結果である（実際には，後半の添え字 $i \approx d$ に対して，$\boldsymbol{b}_i^\star$ のノルムは式 (4.11) には従わない．詳細は文献 [4, 78] を参照）.

BKZ 基底簡約の計算量の見積もり

ブロックサイズ β を利用する際の BKZ 基底簡約の計算量は，β 次元の（ブロック射影）格子上の「SVP オラクルの計算量」と「呼び出し回数」の積で見積もることができる．β 次元格子上の SVP オラクルに適したアルゴリズムとして篩 (sieving) と数え上げ (enumeration) があり，数え上げアルゴリズムは β に関して超指数時間の処理コストであるのに対し，篩アルゴリズムは指数時間の処理コストであり漸近的に数え上げアルゴリズムよりも効率的である．（ただし，数え上げアルゴリズムの空間計算量が β に関して多項式的であるのに対し，篩アルゴリズムの空間計算量は β に関して指数関数的である．）具体的には，β 次元格子上の篩アルゴリズムの時間計算量は

$$2^{c\beta+o(\beta)}$$

で，Locally Sensitive Hashing（LSH）技術を利用した改良により [15, 49]，古典計算機上では $c = 0.292$ である（dimension-for-free の技術を用いなければ，実用上，隠れた準指数部分の因子は 1 以上である．詳細は [32, 58] を参照）．また，多くの篩アルゴリズムは Grover の探索アルゴリズムにより高速化されるため [48, 50]，量子計算機上では $c = 0.265$ と見積もられる（ただし，実用的に量子高速化が可能かは依然として根拠が弱い．詳細は [6] を参照）．一方，数え上げアルゴリズムの時間計算量は古典計算機上で

$$2^{c_1\beta\log\beta+c_2\beta+c_3} \quad \text{または} \quad 2^{c_1\beta^2+c_2\beta+c_3}$$

で，Grover の探索アルゴリズムにより量子計算機上ではその指数部分が半分になると見積もられる（定数 c_1, c_2, c_3 に関しては様々な評価値があり，具体的な値については [2, Table 4] を参照）．一方で，BKZ 基底簡約のアルゴリズム内の SVP オラクルの呼び出し回数については，入力するブロックサイズの β，または入力する格子次元 d に対し $8d$ と見積もることがある（例えば，文献 [1] を参照）．また，最小回数である 1 回の β 次元の SVP アルゴリズムの計算量困難性を**コア SVP 困難性**（Core-SVP hardness）とよび [10]，攻撃者に有利な条件設定で格子に基づく暗号方式の安全性を評価・比較することが多い（コア SVP 困難性による解析の正当性は，文献 [7] 内の解析と実験から検証済み）．

4.3 q-ary 格子と LWE 問題の求解法

本節では，$\mathbb{Z}_q$ 上の LWE 問題を解くための特殊な q-ary 格子について述べると共に，q-ary 格子を用いた LWE 問題の求解法についてまとめておく（q-ary 格子の詳細については，[16] を参照）．

4.3.1 q-ary 格子

奇素数 q に対して，$q\mathbb{Z}^m \subseteq L \subseteq \mathbb{Z}^m$ を満たす完全階数の m 次元格子 L を **q-ary 格子**（q-ary lattice）とよぶ．

2 つの自然数 ℓ, m に対して，$\ell \times m$ の整数行列 $\boldsymbol{M} \in \mathbb{Z}^{\ell \times m}$ に対する 2 つの m 次元 q-ary 格子を

$$\begin{aligned} \Lambda_q(\boldsymbol{M}) &\coloneqq \left\{ \boldsymbol{y} \in \mathbb{Z}^m : \exists \boldsymbol{s} \in \mathbb{Z}^\ell \text{ s.t. } \boldsymbol{y} \equiv \boldsymbol{s}\boldsymbol{M} \pmod q \right\}, \\ \Lambda_q^\perp(\boldsymbol{M}) &\coloneqq \left\{ \boldsymbol{y} \in \mathbb{Z}^m : \boldsymbol{y}\boldsymbol{M}^\top \equiv \boldsymbol{0} \pmod q \right\} \end{aligned} \tag{4.12}$$

と定める（詳細は [16] を参照）．これらの 2 つの集合は，共に $\mathbb{R}^m$ の離散加法部分群なので，格子である．正規化の差を除き，これら 2 つの q-ary 格子は互いに双対の関係にある．より正確には，

$$\Lambda_q^\perp(\boldsymbol{M}) = q\widehat{\Lambda_q(\boldsymbol{M})}, \quad \Lambda_q(\boldsymbol{M}) = q\widehat{\Lambda_q^\perp(\boldsymbol{M})} \tag{4.13}$$

が成り立つ．また，群準同型写像

$$f : \mathbb{Z}^m \longrightarrow (\mathbb{Z}/q\mathbb{Z})^\ell, \quad \boldsymbol{y} \longmapsto \boldsymbol{y}\boldsymbol{M}^\top \bmod q \tag{4.14}$$

の核は q-ary 格子 $\Lambda_q^\perp(\boldsymbol{M})$ である．ここで，群の準同型定理から

$$\mathrm{vol}(\Lambda_q^\perp(\boldsymbol{M})) = [\mathbb{Z}^m : \Lambda_q^\perp(\boldsymbol{M})] = \#\mathrm{Im}(f)$$

が成り立つ（$\#S$ は集合 S の要素数とする）．ただし，群の指数 $[\mathbb{Z}^m : \Lambda_q^\perp(\boldsymbol{M})]$ は格子の体積の比

$$\frac{\mathrm{vol}(\Lambda_q^\perp(\boldsymbol{M}))}{\mathrm{vol}(\mathbb{Z}^m)} = \mathrm{vol}(\Lambda_q^\perp(\boldsymbol{M}))$$

に一致することに注意する．これより，$\mathrm{Im}(f)$ は $(\mathbb{Z}/q\mathbb{Z})^\ell$ の部分群なので，体積 $\mathrm{vol}\left(\Lambda_q^\perp(\boldsymbol{M})\right)$ は q^ℓ を割る．また，元の格子と双対格子の体積の関係式 (4.6) から，$q^{m-\ell}$ は体積 $\mathrm{vol}\left(\Lambda_q(\boldsymbol{M})\right)$ を割ることが分かる（式 (4.13) の双対関係に注意）．さらに，ほとんどの多くの行列 $\boldsymbol{M}$ に対して，式 (4.14) の群準同型写像 f は全射で，その場合は

$$\mathrm{vol}\left(\Lambda_q^\perp(\boldsymbol{M})\right) = q^\ell, \quad \mathrm{vol}\left(\Lambda_q(\boldsymbol{M})\right) = q^{m-\ell}$$

が成り立つ．一方，q-ary 格子 $\Lambda_q(\boldsymbol{M})$ 上の任意のベクトルは $\boldsymbol{y} = \boldsymbol{s}\boldsymbol{M} + q\boldsymbol{z}$ $(\exists \boldsymbol{s} \in \mathbb{Z}^\ell, \exists \boldsymbol{z} \in \mathbb{Z}^m)$ とかけるので，その格子は $(\ell + m) \times m$ の整数行列

$$\begin{pmatrix} \boldsymbol{M} \\ q\boldsymbol{I}_m \end{pmatrix} \in \mathbb{Z}^{(\ell+m)\times m}$$

の一次従属な $(\ell + m)$ 個の行ベクトルで生成される．この生成行列の Hermite Normal Form（HNF）を計算して一次独立なベクトルを求めることで，m 次元 q-ary 格子 $\Lambda_q(\boldsymbol{M})$ の基底行列 $\boldsymbol{B} \in \mathbb{Z}^{m\times m}$ が得られる．また，双対基底の性質から，もう片方の q-ary 格子 $\Lambda_q^\perp(\boldsymbol{M})$ の基底行列は

$$(q\boldsymbol{B}^{-1})^\top \in \mathbb{Z}^{m\times m}$$

で得られる（式 (4.13) の双対関係より，q 倍を乗じる必要がある）．

4.3.2 LWE 問題の格子問題への帰着

ここでは，$\mathbb{Z}_q$ 上の LWE 問題の求解のための格子問題への帰着について述べる．特に，後述の ML-KEM パラメータに対する攻撃計算量の評価のために，Module-LWE 問題を $\mathbb{Z}_q$ 上の LWE 問題に帰着し行列表示した式 (4.5) を考える．

Primal 攻撃：探索 LWE 問題に対する求解

探索 LWE 問題を，目標ベクトルがある格子ベクトルに近いという条件下での CVP である **BDD**（Bounded Distance Decoding）問題に帰着して解く方法を紹介する．

定義 4.3 (BDD 問題)**.** 格子 L と目標ベクトル $\boldsymbol{t}$ に対して，ある $0 < \mu \leq \frac{1}{2}$ が存在し

$$\mathrm{dist}(\boldsymbol{t}, L) \coloneqq \min_{\boldsymbol{v} \in L} \|\boldsymbol{t} - \boldsymbol{v}\| < \mu \lambda_1(L)$$

を満たすと仮定する．格子 L の基底が与えられたとき，目標ベクトル $\boldsymbol{t}$ に最も近い格子ベクトル $\boldsymbol{v} \in L$ を見つけよ．

次元 nk の $\mathbb{Z}_q$ 上の m 個の LWE サンプル $(\boldsymbol{A}, \boldsymbol{t}) \in \mathbb{Z}_q^{m \times nk} \times \mathbb{Z}_q^m$（式 (4.5) を参照）は，関係式

$$\boldsymbol{t} \equiv \boldsymbol{s}\boldsymbol{A}^\top + \boldsymbol{e} \pmod{q}$$

を満たすので，式 (4.3) の秘密ベクトル $\boldsymbol{s} \in \mathbb{Z}_q^{nk}$ を見つける探索 LWE 問題は，$\boldsymbol{t} \in \mathbb{Z}_q^m$ を目標ベクトルとする q-ary 格子 $\Lambda_q(\boldsymbol{A}^\top)$ 上の BDD 問題とみなせる．具体的には，

$$\boldsymbol{t} = \boldsymbol{s}\boldsymbol{A}^\top + \boldsymbol{e} + q\boldsymbol{w} \quad (\exists \boldsymbol{w} \in \mathbb{Z}^m) \tag{4.15}$$

と表した目標ベクトルに対して，q-ary 格子上のベクトルを

$$\boldsymbol{v} = \boldsymbol{s}\boldsymbol{A}^\top + q\boldsymbol{w} \in \Lambda_q\big(\boldsymbol{A}^\top\big)$$

とおくと，$\boldsymbol{t} - \boldsymbol{v} = \boldsymbol{e}$ が成り立つ．また，ノイズベクトル $\boldsymbol{e} \in \mathbb{Z}^m$ のすべての成分が中心が 0 で標準偏差が $\sigma > 0$ の離散分布からサンプルされた場合，そのノルムはおおよそ

$$\|\boldsymbol{e}\| \approx \sigma\sqrt{m}$$

と見積もれる．ゆえに，目標ベクトル $\boldsymbol{t}$ との距離がおおよそ $\sigma\sqrt{m}$ となる q-ary 格子 $\Lambda_q(\boldsymbol{A}^\top)$ 上の格子ベクトル $\boldsymbol{v}$ を見つけることで，ノイズベクトル $\boldsymbol{e}$ を復元することができる．また，ノイズベクトル $\boldsymbol{e}$ が復元できた場合，$\boldsymbol{t} - \boldsymbol{e} \equiv \boldsymbol{s}\boldsymbol{A}^\top \pmod{q}$ の関係式にガウスの消去法を適用すれば，秘密ベクトル $\boldsymbol{s} \in \mathbb{Z}_q^{nk}$ を見つけることができる．

一方，BDD 問題の求解には，CVP を SVP に帰着する Kannan や Bai-Galbraith らの**埋め込み法**（embedding techniques）[45, 14] が実用的である．特に，ML-KEM に対しては，秘密ベクト

ル $\boldsymbol{s} \in \mathbb{Z}_q^{nk}$ のノルムが非常に短いので，Bai-Galbraith の埋め込み法 [14] が有効である．具体的には，$d = m + nk + 1$ 次の正方行列

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{O}_{m,nk} & q\boldsymbol{I}_m & \boldsymbol{0}_m^\top \\ \boldsymbol{I}_{nk} & \boldsymbol{A}^\top & \boldsymbol{0}_{nk}^\top \\ \boldsymbol{0}_{nk} & -\boldsymbol{t} & 1 \end{pmatrix} \in \mathbb{Z}^{d \times d} \tag{4.16}$$

を考える．ただし，$\boldsymbol{O}_{m,\ell}$ は $m \times \ell$ の零行列，$\boldsymbol{0}_\ell$ はすべての成分が 0 の長さ ℓ の行ベクトルとする．ここで，$\boldsymbol{B}$ の d 個の行ベクトルで生成される d 次元の格子を L とする．特に，L の基底行列の構成から，格子 L の体積は

$$\mathrm{vol}(L) = |\det(\boldsymbol{B})| = q^m$$

である．このとき，式 (4.15) より，

$$\boldsymbol{z} = \big(\boldsymbol{s} \mid -\boldsymbol{e} \mid 1\big) = \big(\boldsymbol{w} \mid \boldsymbol{s} \mid 1\big)\boldsymbol{B} \in L \tag{4.17}$$

である．つまり，秘密ベクトルとノイズベクトルを連結した非常に短いベクトル $\boldsymbol{z} \in \mathbb{Z}^d$ が，格子 L の非零な最短ベクトルとして埋め込まれる．（式 (4.7) から，L がランダムな格子であれば $\lambda_1(L) = O\left(q^{m/d}\right)$ と予想されるが，q が十分大きく，特に $\|\boldsymbol{z}\| \ll q^{m/d}$ の場合，$\boldsymbol{z}$ が格子 L の非零な最短ベクトルと予想される．）そこで，BKZ 基底簡約などの SVP を解く格子アルゴリズムを利用して，$\boldsymbol{z} \in L$ を復元することができれば，秘密ベクトル $\boldsymbol{s}$ とノイズベクトル $\boldsymbol{e}$ を同時に見つけることができる．このように，秘密ベクトル $\boldsymbol{s}$ を見つける探索 LWE 問題を，BDD 問題に帰着させたのちに，埋め込み法で解く方法は **primal 攻撃**とよばれる（例えば，文献 [2] を参照）．

注意 4.1. 近年，機械学習を利用した（$\mathbb{Z}_q$ 上の）LWE 問題に対する攻撃法とその改良法がいくつか提案されている [54, 72, 74]．ただし，それらの機械学習を利用した攻撃法は，成分が疎かつ小さい秘密ベクトルを持つ LWE 問題にのみに有効で，秘密ベクトルの成分（または係数）が中心二項分布 CBD_η でサンプリングされる ML-KEM には現状では脅威とはならない（成分が疎かつ小さい秘密ベクトルを持つ LWE インスタンスに対する機械学習を利用した攻撃を含む各種攻撃のベンチマーク実験については，文献 [75] を参照）．

Dual 攻撃：判定 LWE 問題に対する求解

次は，判定 LWE 問題を **SIS**（Short Integer Solution）問題に帰着して解く方法を紹介する．

定義 4.4 (SIS 問題)**.** 奇素数 q と $0 < \xi < q$ を満たす（小さい）実数 ξ を固定する．すべての成分が $\mathbb{Z}_q$ 上一様ランダムに選ばれた $\ell \times m$ 整数行列 $\boldsymbol{M}$ に対して，

$$\|\boldsymbol{x}\| \le \xi \quad \text{かつ} \quad \boldsymbol{x}\boldsymbol{M}^\top \equiv \boldsymbol{0} \pmod{q}$$

を満たす非零ベクトル $\boldsymbol{x} \in \mathbb{Z}^m$ を見つけよ．これは，q-ary 格子 $\Lambda_q^\perp(\boldsymbol{M})$ 上の短い非零ベクトルを見つける問題と言い換えることができる（具体的な q-ary 格子の構成については，式 (4.12) を参照）．

次元 nk の $\mathbb{Z}_q$ 上の m 個の LWE サンプル $(\boldsymbol{A}, \boldsymbol{t}) \in \mathbb{Z}_q^{m \times nk} \times \mathbb{Z}_q^m$（式 (4.5) を参照）に対して，$nk \times m$ の転置行列 $\boldsymbol{A}^\top$ に対する SIS 問題の十分短い解ベクトル

$$\boldsymbol{x} \in \Lambda_q^\perp\big(\boldsymbol{A}^\top\big) \iff \boldsymbol{x}\boldsymbol{A} \equiv \boldsymbol{0} \pmod{q}$$

が得られたとする．このとき，行列表示の m 個の LWE サンプル $(\boldsymbol{A}, \boldsymbol{t})$ は $\boldsymbol{t} \equiv \boldsymbol{s}\boldsymbol{A}^\top + \boldsymbol{e} \pmod{q}$ を満たすので，

$$\begin{aligned} \langle \boldsymbol{x}, \boldsymbol{t} \rangle &\equiv \langle \boldsymbol{x}, \boldsymbol{s}\boldsymbol{A}^\top + \boldsymbol{e} \rangle \\ &= \langle \boldsymbol{x}\boldsymbol{A}, \boldsymbol{s} \rangle + \langle \boldsymbol{x}, \boldsymbol{e} \rangle \equiv \langle \boldsymbol{x}, \boldsymbol{e} \rangle \pmod{q} \end{aligned} \tag{4.18}$$

が成り立つ．ここで，ノイズベクトル $\boldsymbol{e}$ のすべての成分が中心が 0 で標準偏差が $\sigma > 0$ の離散分布からサンプルされたとすると，

$$z = \langle \boldsymbol{x}, \boldsymbol{t} \rangle \bmod q = \langle \boldsymbol{x}, \boldsymbol{e} \rangle \in \mathbb{Z}_q$$

は標準偏差が σm の離散分布に従う．一方，組 $(\boldsymbol{A}, \boldsymbol{t}) \in \mathbb{Z}_q^{m \times nk} \times \mathbb{Z}_q^m$ が，LWE サンプルでなく $\mathbb{Z}_q$ 上一様ランダムにサンプルされたものであれば，$z = \langle \boldsymbol{x}, \boldsymbol{t} \rangle \bmod q$ は $\mathbb{Z}_q$ 上一様分布に従う．これより，判定 LWE 問題に対して，

$$\varepsilon = 4\exp\left(-2\pi^2\tau^2\right) \quad \left(\tau = \frac{\sigma m}{q}\right) \tag{4.19}$$

の advantage を持つ．このような判定 LWE 問題に対する解法は **dual 攻撃**とよばれる（例えば，文献 [2] を参照）．

primal 攻撃における Bai-Galbraith 埋め込み法のように，秘密ベクトル $\boldsymbol{s}$ が短い場合は（例えば，ML-KEM 方式に対しては），$(m + nk) \times nk$ の整数行列

$$\boldsymbol{A}' = \begin{pmatrix} \boldsymbol{A} \\ -\boldsymbol{I}_{nk} \end{pmatrix}$$

とおき，q-ary 格子

$$\Lambda' = \Lambda_q^\perp(\boldsymbol{A}'^\top) = \big\{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{Z}^m \times \mathbb{Z}^{nk} \mid \boldsymbol{x}\boldsymbol{A} \equiv \boldsymbol{y} \pmod{q}\big\}$$

を考えるのが有用である．実際，q-ary 格子 Λ' の基底行列に十分大きなブロックサイズ β の BKZ 基底簡約を適用し，短い非零ベクトル $\boldsymbol{b} = (\boldsymbol{x}, \boldsymbol{y}) \in \Lambda'$ を見つけたとする（一般に，$\boldsymbol{b}$ として，Λ' の β-BKZ 簡約基底の第 1 基底ベクトル $\boldsymbol{b}_1$ をとる）．このとき，式 (4.18) と同じように，

$$\begin{aligned} \langle \boldsymbol{x}, \boldsymbol{t} \rangle &\equiv \langle \boldsymbol{x}, \boldsymbol{s}\boldsymbol{A}^\top + \boldsymbol{e} \rangle \\ &= \langle \boldsymbol{x}\boldsymbol{A}, \boldsymbol{e} \rangle + \langle \boldsymbol{x}, \boldsymbol{e} \rangle \equiv \langle \boldsymbol{y}, \boldsymbol{s} \rangle + \langle \boldsymbol{x}, \boldsymbol{e} \rangle \pmod{q} \end{aligned}$$

が成り立つ．これより，秘密ベクトル $\boldsymbol{s}$ とノイズベクトル $\boldsymbol{e}$ のすべての成分が中心が 0 で標準偏差 $\sigma > 0$ の離散分布からサンプルされたとすると，上式から

$$|\langle \boldsymbol{x}, \boldsymbol{t} \rangle \bmod q| = |\langle \boldsymbol{x}, \boldsymbol{e} \rangle + \langle \boldsymbol{y}, \boldsymbol{s} \rangle| \lessapprox \sigma\ell \tag{4.20}$$

が成り立つ．ただし，$\ell = \|\boldsymbol{b}\| = \|(\boldsymbol{x}, \boldsymbol{y})\|$ とする．一方，4.3.1 項で述べた q-ary 格子の性質から，q-ary 格子 Λ' の次元は m で，体積は q^{kn} なので，式 (4.11) から

$$\ell = \|\boldsymbol{b}\| \approx \delta_\beta^{m+kn-1} \cdot q^{\frac{kn}{m+kn}}$$

と見積もれる．式 (4.19) と (4.20) より，ε の advantage をもつ攻撃者は，不等式

$$-2\pi^2\tau^2 \geq \ln\left(\frac{\varepsilon}{4}\right) \quad \left(\tau = \frac{\sigma\ell}{q}\right)$$

を満たす（最小の）ブロックサイズ β を入力とする BKZ 基底簡約の計算時間を必要とする．実際の攻撃では少なくとも $\frac{1}{2}$ の advantage が必要なため，攻撃者はおよそ $\frac{1}{\varepsilon^2}$ 個の Λ' の短いベクトルを生成して，攻撃の成功確率を増幅させる必要がある．特に，篩アルゴリズムでは $2^{0.2075\beta}$ 個のベクトルを生成するので，攻撃者は少なくとも

$$R = \max\left\{1,\ \frac{1}{2^{0.2075\beta}\varepsilon^2}\right\}$$

回の繰り返しを必要とする．ただし，篩アルゴリズムが出力するすべての格子ベクトルが非零な最短ベクトルと同程度に短いという保守的な（攻撃者に有利な）仮定の下での議論である（詳細は文献 [13, Section 5.1.3] を参照）．

注意 4.2. 成分が $0, \pm 1$ のいずれかの秘密ベクトル $\boldsymbol{s}$ を持つ（$\mathbb{Z}_q$ 上の）LWE 問題に対して，2021 年に May [60] は格子アルゴリズムと中間一致攻撃を組み合わせた求解法を提案した．具体的には，LWE 関係式 $\boldsymbol{t} \equiv \boldsymbol{s}\boldsymbol{A}^\top + \boldsymbol{e} \pmod q$ において，秘密ベクトルを 2 分割 $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2)$ して，

$$\boldsymbol{t} - \boldsymbol{s}_1\boldsymbol{A}_1^\top \equiv \boldsymbol{s}_2\boldsymbol{A}_2^\top + \boldsymbol{e} \pmod q \tag{4.21}$$

を考える．ただし，$\boldsymbol{A}_1, \boldsymbol{A}_2$ は，秘密鍵ベクトルの分割 $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2)$ に合わせて公開行列 $\boldsymbol{A}$ を分割した行列とする．上式の両辺に対して，中間一致攻撃を適用して，$(\boldsymbol{s}_1, \boldsymbol{s}_2)$ を探す（この求解法の進展として，primal 攻撃との組み合わせは文献 [41]，dual 攻撃との組み合わせは文献 [17] などを参照）．中間一致攻撃との組み合わせによる求解法は，成分が疎かつ（$0, \pm 1$ のような）小さい秘密ベクトル $\boldsymbol{s}$ を持つ LWE 問題にしか有効ではなく，$\boldsymbol{s}$ のすべての成分が中心二項分布 CBD_η ($\eta \in \{2, 3\}$) でサンプリングされる ML-KEM 方式には有効ではない（具体的には，ML-KEM の秘密ベクトル $\boldsymbol{s}$ の数え上げの計算量について，後述の 5.1 節を参照）．

4.4 ML-KEM パラメータに対する攻撃計算量の見積もり

本節では，ML-KEM の暗号パラメータ（2.2.3 項）に対する攻撃計算量を見積もる．文献 [8, 19] で述べられているように，$\mathbb{Z}_q$ 上の LWE 問題に対して様々な攻撃法がある．しかし，ML-KEM では，式 (2.16) の暗号文の形から，得られる $\mathbb{Z}_q$ 上の LWE サンプル数が最大 $(k+1)n$ なので，非常に多くの LWE サンプル数を必要とする BKW 型攻撃 [47] と線形攻撃 [12] を除外することができる．これより，ML-KEM に対しては，本質的には 4.3.2 項で説明した primal 攻撃・dual 攻撃（と BKZ 基底簡約による求解との組み合わせ）の 2 つの攻撃法だけが対象となる．

4.4.1 コア SVP 困難性による攻撃計算量の見積もり

ここでは，ML-KEM の暗号パラメータに対して，攻撃で必要となる BKZ 基底簡約のブロックサイズ β を見積もるとともに，BKZ 基底簡約内の 1 回の β 次元の SVP アルゴリズムのコア SVP 困難性 [10] による攻撃計算量を見積もる．また，文献 [13, Section 5.2] で議論されているように，コア SVP 困難性による見積もりでは dual 攻撃の方が primal 攻撃よりほんの少し計算量が低くなるが，実際にはより多くの攻撃計算量を要する．具体的には，上述の dual 攻撃において，篩アルゴリズムで指数関数的に多くの格子ベクトルを生成できると仮定しているが，それらの多くは $\sqrt{4/3}$ 倍程度長い．さらに，指数関数的に多くの短い格子ベクトルを生成できるという仮定は，篩アルゴリズムに関する近年の改良と整合性が取れない（例えば，dimension-for-free 改良 [32] など）．一方，これらの余分な短い格子ベクトルを仮定しない解析 [2] では，primal 攻撃よりも dual 攻撃の方がかなり計算コストがかかると予想している．そのため，以下では primal 攻撃の計算量についてのみ議論する．

4.3.2 項で説明した primal 攻撃では，式 (4.17) の形の LWE 問題の秘密とノイズの連結ベクトル $\boldsymbol{z}$ を，体積が q^m で次元が $d = m + nk + 1$ の Bai-Galbraith 埋め込み格子 L の最短ベクトルとして埋め込む．また，primal 攻撃では，格子 L の基底に BKZ 基底簡約アルゴリズムを適用することで，最短ベクトル $\boldsymbol{z}$ を見つけることを考える．ここで，$\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d\}$ を格子 L の β-BKZ 簡約基底とし，$\{\boldsymbol{b}_1^\star, \ldots, \boldsymbol{b}_d^\star\}$ をその Gram-Schmidt ベクトルとする．GSA と Gauss のヒューリスティックから，各 Gram-Schmidt ベクトルのノルムについて，式 (4.11) が成り立つとする（これは攻撃者に有利なモデルで，かなり楽観的な仮定である）．このとき，目的の最短ベクトル $\boldsymbol{z}$ の $d - \beta + 1$ の位置における射影ベクトル $\pi_{d-\beta+1}(\boldsymbol{z}) \in \pi_{d-\beta+1}(L)$ のノルムが

$$\begin{aligned} \sigma\sqrt{\beta} \approx \|\pi_{d-\beta+1}(\boldsymbol{z})\| \leq \|\boldsymbol{b}_{d-\beta+1}^\star\| \approx \delta_\beta^{2\beta-d-1} \cdot \mathrm{vol}(L)^{1/d} \\ \iff \sigma\sqrt{\beta} \leq \delta_\beta^{2\beta-d-1} \cdot q^{m/d} \end{aligned} \tag{4.22}$$

を満たせば，BKZ 基底簡約の第 1 基底ベクトルとして目的の $\boldsymbol{z} \in L$ を見つけることができる（探索 LWE 問題に対する BKZ による求解実験については，文献 [4, 7, 65] を参照）．ただし，式 (4.17) の形の $\boldsymbol{z}$ のノルムは

$$\|\boldsymbol{z}\| \approx \sigma\sqrt{kn + m} \approx \sigma\sqrt{d} \tag{4.23}$$

と見積もれ，その射影ベクトル $\pi_{d-\beta+1}(\boldsymbol{z})$ のノルムは，文献 [7, Section 4.1] から

$$\|\pi_{d-\beta+1}(\boldsymbol{z})\| \approx \sqrt{\frac{\beta}{d}} \cdot \|\boldsymbol{z}\| \approx \sigma\sqrt{\beta}$$

と見積もれる．また，δ_β は式 (4.10) の値とする．不等式 (4.22) を満たす最小の β が，primal 攻撃が成功する最小の BKZ 基底簡約のブロックサイズと期待される．

表 4.1 に，ML-KEM の 3 つの暗号パラメータ（2.2.3 項）に対して，不等式 (4.22) を満たす primal 攻撃に必要な最小の BKZ ブロックサイズ β と，BKZ 基底簡約のサブルーチンである β 次

表 4.1　ML-KEM の安全性を支える Module-LWE 問題に対するコア SVP による攻撃計算量見積もり [13, Table 4]（帰着する $\mathbb{Z}_q$ 上の LWE サンプル $(\boldsymbol{A}, \boldsymbol{b}) \in \mathbb{Z}_q^{m \times nk} \times \mathbb{Z}_q^m, \boldsymbol{b} = \boldsymbol{s}\boldsymbol{A}^\top + \boldsymbol{e}$ に対して，d 次元の Bai-Galbraith 埋め込み格子を用いた primal 攻撃の計算量見積もり）

ML-KEM パラメータ（$n \times k$ の値）	512	768	1024
$\boldsymbol{s}, \boldsymbol{e}$ に関する中心二項分布 CBD_{η_1} の η_1	$\eta_1 = 3$	$\eta_1 = 2$	$\eta_1 = 2$
攻撃に利用する格子次元 $d = m + nk + 1$	999	1419	1885
攻撃に必要な BKZ の最小ブロックサイズ β	406	626	878
コア SVP の古典計算量（ビット）	118	183	256
コア SVP の量子計算量（ビット）	107	166	232

元 SVP アルゴリズムの 1 回の計算量であるコア SVP 困難性の古典と量子の計算量見積もりをまとめる（詳細は [13, Table 4] を参照）．ただし，攻撃に必要なサンプル数 m，BKZ 基底簡約の最小ブロックサイズ β，コア SVP 困難性の古典計算量と量子計算量は下記のように見積もる．

- ML-KEM では，秘密ベクトル $\boldsymbol{s}$ と公開鍵のノイズベクトル $\boldsymbol{e}$ のすべての $\mathbb{Z}_q$ 係数は中心二項分布 CBD_{η_1} からサンプリングされる．具体的には，ML-KEM-512 の場合は $\eta_1 = 3$，ML-KEM-768, 1024 の場合は $\eta_1 = 2$ で，式 (4.23) に対して中心二項分布 CBD_{η_1} の標準偏差は $\sigma = \sqrt{\eta_1/2}$ である（つまり，分散は $\sigma^2 = \eta_1/2$ である）．特に，ML-KEM-512 の場合，暗号文におけるノイズ $\boldsymbol{e}_1, e_2$ は $\eta_2 = 2$ による CBD_{η_2} からサンプリングされるが，Compress 関数により暗号文におけるノイズは暗黙的に増幅するので，ここでは η_1 の値にのみ着目すればよい（参考程度であるが，$\eta_2 = 2$ でコア SVP の古典計算量を算出すると 112 ビットとなり，表 4.1 内の 118 ビットよりも 6 ビット下がる）．
- primal 攻撃に利用するサンプル数 m と不等式 (4.22) を満たす BKZ 基底簡約のブロックサイズ β の最良の組 (m, β) は，primal 攻撃が最も有効となるサンプル数 m を 1 から $(k+1)n$ の中から求めた上で，不等式 (4.22) を満たす最小のブロックサイズ β を求める．具体的には，GitHub：`https://github.com/pq-crystals/security-estimates` 内の Kyber.py コードから求まる（コード内では，dual 攻撃に必要な BKZ 基底簡約のブロックサイズも求めている）．
- また，BKZ 基底簡約のサブルーチンである β 次元におけるコア SVP 困難性（篩アルゴリズム）の計算量として，4.2.3 項から，古典計算機で $2^{0.292\beta}$，量子計算機で $2^{0.265\beta}$ と見積もる（これは攻撃者にかなり有利な計算量見積もりである）．

図 4.1 に，表 4.1 の ML-KEM の暗号パラメータに対するコア SVP の計算量見積もりの検証用 Sage コードを示す（Sage Math Cell：`https://sagecell.sagemath.org/` 上で動作可能）．具体的には，ML-KEM に関する暗号パラメータ (n, k, q, η_1) と表 4.1 にある攻撃に利用する最良のサンプル数 m の値を代入すれば，攻撃に必要な BKZ の最小ブロックサイズ β と，その時のコア SVP の古典計算量と量子計算量を算出する．ただし，$n \times k = 768, 1024$ の場合，不等式 (4.22) を

図 4.1　表 4.1 のコア SVP 困難性の計算量見積もりの検証用 Sage コード

```
k = 2; n = 256; q = 3329; eta = 3
sigma = RR(sqrt(eta/2))
m = 486; d = m+k*n+1

for b in range(100, 1000):
    A = sigma*sqrt(b)
    delta = ((math.pi*b)^(1.0/b)*b/(2*math.pi*exp(1)))^(1.0/(2*(b-1)))
    B = delta^(2*b-d-1)*q^(m/d)
    if RR(A) < RR(B):
        print("攻撃に必要な最小ブロックサイズ = ", b)
        print("古典計算量（ビット）= ", 0.292*b)
        print("量子計算量（ビット）= ", 0.265*b)
        break
```

満たす primal 攻撃に必要な BKZ の最小ブロックサイズ β は，表 4.1 の β から -1 または -2 程度ずれるが，コア SVP の計算量には大きなずれはない．

4.4.2　最新の技術と解析による攻撃計算量の見積もり

ここでは，BKZ 基底簡約におけるシミュレーションおよび progressive 化や dimension-for-free[32] などの最新の技術の効果を考慮した，ML-KEM の安全性を支える Module-LWE 問題に対する primal 攻撃に必要となる BKZ のブロックサイズ β の再見積もりを行う．また，文献 [6] の解析に基づくゲートコストの見積もりを示す．まず，BKZ 基底簡約の progressive 化によるオーバーヘッド定数を

$$C = \lim_{\beta\to\infty} \frac{\sum_{i=0}^{\beta} 2^{0.292i}}{2^{0.292\beta}} = \frac{2^{0.292}}{2^{0.292}-1} \approx 5.46$$

と定める．以下では，ML-KEM-512 の暗号パラメータ（$nk = 512$）に対してのみ議論する．

BKZ シミュレーションの利用

Primal 攻撃の成功条件に関する不等式 (4.22) は，GSA 仮定下の BKZ 簡約基底の Gram-Schmidt ベクトルのノルム評価 (4.11) に依存する．しかし，実際の Gram-Schmidt ベクトル $\boldsymbol{b}_1^\star, \ldots, \boldsymbol{b}_d^\star$ では，後半の添え字 $i \approx d$ に対して，$\boldsymbol{b}_i^\star$ のノルムは式 (4.11) には必ずしも従わない [4, 78]．そこで，leaky-LWE-estimator [29] の一部のシミュレーターを使う．これは progressive-BKZ 基底簡約 [5, 11] を利用しており，このシミュレーターでは攻撃に必要な BKZ ブロックサイズとして

$$\beta = 413$$

が必要であると見積もれる．（正確には，$\beta = 412$ から $\beta = 413$ にすることで，コストが $2^{0.292} \approx 1.224$ 倍増幅する一方で，成功確率は 1.373 倍増幅する．）また，利用する格子次元は $d = 1025$ である．β 次元の SVP に対する篩アルゴリズムの計算コストと比べて，progressive-BKZ

基底簡約の計算コストはおおよそ

$$C \cdot (d - \beta) = 5.46 \times (1025 - 413) \approx 3340 \tag{4.24}$$

倍に増加する．

Dimension-for-free 技術の効果

文献 [32] の dimension-for-free（d_{4f}）技術の効果を考慮すると，BKZ ブロックサイズ $\beta = 413$ に対して

$$d_{4f} = \frac{\beta \ln(4/3)}{\ln(\beta/(2\pi e))} \approx 37.3$$

を得る（整数に切り上げることで，$d_{4f} = 38$ とする）．これより，BKZ 基底簡約内で呼び出される β 次元の SVP オラクルとして，

$$\beta' = \beta - d_{4f} = 375$$

次元の篩アルゴリズムを用いればよい．（文献 [5] の “on-the-fly lifting” と “pump-down sieves” の 2 つのトリックを用いると，上記の d_{4f} の値よりもわずかに多くの free 次元を得られる可能性がある．実用的には，メモリを $2^{0.5}$ 程度削減できるが，計算時間への影響は限定的である．）

篩アルゴリズムのゲートコスト

古典および量子回路による篩アルゴリズムの計算コストの最新の解析 [6] では，“AllPairSearch” 関数に着目している．具体的には，篩に関する球面キャップとくさびの正確な体積を求め，最内側の繰り返し関数に対する正確なゲートカウントを計算し，パラメータの自動最適化を行うことで，古典と量子の計算コストを得る．最良の古典アルゴリズムに関して，文献 [6] の解析では，$\beta' = 375$ 次元における AllPairSearch に対しては，おおよそ

$$2^{137.4}$$

ゲートのコストと結論づけている．素朴な篩アルゴリズムでは多項式回の AllPairSearch の呼び出しが必要であるが，progressive 型の篩アルゴリズム [32, 49] では，実用的には比較的少ない呼び出し回数で十分である．そこで，progressive 型の篩アルゴリズムにおいて，各次元ごとに 1 回の AllPairSearch 関数を呼ぶと仮定すると，$\beta' = 375$ 次元までで

$$C \cdot 2^{137.4} \tag{4.25}$$

ゲートのコストがかかる．

最終的なゲートコスト

式 (4.24) と (4.25) から，最終的なゲートコストとして

$$G = (1025 - 413) \cdot C^2 \cdot 2^{137.4} = 2^{151.5}$$

表 4.2　BKZ シミュレーション [29] と dimension-for-free（d_{4f}）技術 [32] を考慮した ML-KEM の安全性を支える Module-LWE 問題に対する primal 攻撃計算量の再見積もりと文献 [6] によるゲートコストとメモリの見積もり（文献 [13, Table 4] を参照）

ML-KEM パラメータ（nk の値）	512	768	1024
$\boldsymbol{s}, \boldsymbol{e}$ に関する中心二項分布 CBD_{η_1} の η_1	$\eta_1 = 3$	$\eta_1 = 2$	$\eta_1 = 2$
攻撃に利用する格子次元 d	1025	1467	1918
攻撃可能な BKZ の最小ブロックサイズ β	413	637	894
篩アルゴリズムの SVP 次元 $\beta' = \beta - d_{4f}$	375	586	829
攻撃に必要なゲートコスト（ビット）	151.5	215.1	287.3
攻撃に必要なメモリ（ビット）	93.8	138.5	189.7
NIST 標準で要求される安全性レベル [63]	143	207	272
（古典ゲート数換算，ビット）	AES-128 相当	AES-192 相当	AES-256 相当

を得る．篩アルゴリズムで利用する格子ベクトルの各成分を 1 バイトで表現できると仮定すると，文献 [6] の解析に従えば，必要なメモリを見積もることが可能である．ML-KEM における 3 つの暗号パラメータに対しては，`https://github.com/lducas/leaky-LWE-Estimator/tree/NIST-round3` のスクリプトから算出できる．

表 4.2 に，BKZ シミュレーションおよび progressive 化や dimension-for-free 技術を考慮した ML-KEM の安全性を支える Module-LWE 問題に対する primal 攻撃計算量の再見積もりと，文献 [6] の解析に従ったゲートコストとメモリの見積もりを示す（文献 [13, Table 4] を参照）．

注意 4.3 (理想的な近傍探索)**.** 文献 [6] の解析では，文献 [15] の篩におけるバケットが球上完全に一様に分布すると仮定している．しかしながら，文献 [15] の篩アルゴリズムは，バケットの分布に存在するある構造を利用しているに違いない．元の解析では，各ペアを見つける成功確率は，その構造により $2^{\widetilde{O}(\sqrt{\beta})}$ の準指数よりも大きな因子には影響しないことを示している．しかしながら，この漸近的な解析はタイトではなく，準備的な実験では，実用的には成功確率のロスはそれほど大きくないことを示唆している．さらに，パラメータに依存するが，文献 [6] の理想化に比べて，文献 [15] の篩アルゴリズムはオーバーヘッドを持つ．具体的には，メモリ使用量を最小化するには，時間計算量として $2^{O(\beta/\log\beta)}$ のオーバーヘッドが生じる．原則的に，メモリ使用量と時間計算量にトレードオフがある．

4.4.3　Dual-sieve 攻撃とその影響

文献 [40] では，$\mathbb{Z}_q$ 上の LWE 問題に対する dual 攻撃とヒューリスティックな推定ステップを組み合わせた改良手法が提案されており，dimension-for-free 技術 [32] の利用と単一の篩によって多数の短い格子ベクトルを生成する手法が示されている．この LWE 問題に対する攻撃手法は dual-sieve 攻撃と呼ばれる．その後，文献 [59] では，FFT（Fast-Fourier-Transformation）

に基づく識別法を利用した dual 攻撃の改良が提案され，ML-KEM（Kyber [13]）に対する攻撃計算量が評価されている．この FFT に基づく dual-sieve 攻撃は dual-sieve-FFT 攻撃と呼ばれ，量子アルゴリズムによる亜種 [9] や符号理論からのアイデアを取り入れた改良 [24]，さらに Module-LWE の代数構造を利用した改良 [76] などが提案されている．文献 [40, 59, 9, 24] で提案された dual-sieve-FFT 攻撃とその改良の実用性については，文献 [35] で理論と実験の両面で検証が行われ，dual-sieve-FFT 攻撃の成功確率は実際よりかなり高く見積もられていると結論付けている．

近年，文献 [23] では，符号理論のアイデアに基づく新しい dual-sieve 攻撃が提案されている．具体的には，文献 [59] で用いられていた modulus switching 技術を，効率的な復号アルゴリズムに置き換える dual-sieve-FFT 攻撃の改良である．この新しい dual-sieve-FFT 攻撃による ML-KEM（Kyber）-512, 768, 1024 の安全性レベル [13, Table 4]（表 4.2 を参照）は，耐量子計算機暗号の NIST 標準化が要求する安全性レベル 1, 3, 5 に対応うる古典ゲート計算量 143, 207, 272 ビット [63] よりも少なくともそれぞれ

$$3.5,\ 11.9,\ 12.3 \text{ ビット}$$

は下回ると主張されている（詳細は [23, Table 5.1] を参照）．ただし，文献 [23] における dual-sieve-FFT 攻撃の計算量評価は，最近傍探索に関する文献 [15] による理想的な理論モデルに基づく（注意 4.3 を参照）．また，文献 [33] で言及されているように，文献 [15] による理想的な理論モデルでは復号コストを過小評価し，積符号が持つレート-歪み特性の非最適性を考慮してないため，いくつかのオーバーヘッドが隠れている．具体的には，380 次元における篩アルゴリズムに対して，最近傍探索の実際の計算量は，文献 [15] による理想的な理論モデルより 2^6 倍程度増加する（詳細は文献 [33] を参照）．

4.5 Module-LWE 問題に対する代数構造を利用した攻撃とその影響

Module-LWE 問題に対する上記の攻撃アルゴリズムは，ML-KEM 方式の構成における基礎環 $R = \mathbb{Z}[X]/(X^n + 1)$ $(n = 256)$ に内在する代数構造を一切利用していない．本節では，代数構造を利用した Module-LWE 問題に対する攻撃とその影響について述べる．

4.5.1 加群格子上の格子アルゴリズム

円分体 $K = R \otimes_{\mathbb{Z}} \mathbb{Q} \cong \mathbb{Q}(\zeta_{2n})$ のイデアル格子上の篩アルゴリズムは，処理性能とメモリ使用量の両面において有効である．具体的には，メモリ使用量については自由 $\mathbb{Z}$ 加群 $R = \mathbb{Z}[X]/(X^n+1)$ の階数 n の因子分だけ削減できる．また，処理性能については，近傍探索テクニックを利用するかどうかに依存するものの，$O(n)$ から $O(n^2/\log n)$ の間の因子分の高速化が期待できる．これは，イデアル格子の対称性から，式 (4.1) の回転操作を利用すれば，1 つの格子ベクトル（つまり，R

の元 $f(x)$ の係数ベクトル $\boldsymbol{f} \in \mathbb{Z}^n$）から同じノルムを持つ n 個の格子ベクトル

$$\mathrm{rot}^i(\boldsymbol{f}) \quad (i = 1, 2, \ldots, n)$$

を効率的に生成することができる．また，加群格子上の数え上げアルゴリズムについても，その計算量は漸近的に改善されることが文献 [46] において示されている．

また，加群格子上の BKZ 基底簡約アルゴリズム [61] は存在し，内部で呼び出す SVP オラクルは加群格子の対称性の恩恵を受ける可能性がある．しかし，下記のように，このアプローチによる本質的な恩恵を妨げる問題が数多くある．

- まず，$\mathbb{Z}$ 階数 $r \geq 2$ の加群格子上の BKZ 基底簡約を，ML-KEM の基礎環 $R \cong \mathbb{Z}^n$ 上で適用するためには，$r \mid n = 256 = 2^8$ の条件を満たす必要がある（扱う加群の $\mathbb{Z}$ 階数 r が小さいほど，加群格子の対称性から得られる恩恵は限定的となる）．さらに，$\mathbb{Z}$ 階数 r の加群格子上の BKZ 基底簡約のブロックサイズ β は

  $$r = \gcd\bigl(\beta, 256\bigr)$$

  の条件を満たす必要がある．この制約により，攻撃に最適と考えられるブロックサイズ β を自由に選択することが困難となる．特に，ブロックサイズ選択の自由度が低いため，加群格子上の BKZ 基底簡約の progressive 化に支障をきたす．また，dimension-for-free 技術と組み合わせる場合には，

  $$r = \gcd\bigl(\beta, d_{4f}, 256\bigr)$$

  の条件を満たす必要があるため，ブロックサイズ β の選択に更なる制限が課される．加えて，Kannan の埋め込み法を $\mathbb{Z}$ 階数が r の加群構造に適用する際，特別な調整が必要となり，全体の格子次元は 1 次元ではなく，少なくとも r 次元分増加させる必要がある．
- 次に，同じブロックサイズ β を利用したとしても，通常（つまり $\mathbb{Z}$ 格子上）の BKZ と加群版の BKZ が同程度の品質を持つ基底を出力するかどうかは明らかではない．（文献 [34] では，非構造化格子よりも加群格子上の BKZ 基底簡約はより多くのブロックサイズを必要とすると結論付けている．）これは，$\beta = 2$ の BKZ 基底簡約である LLL 基底簡約の場合でさえ未解決である（詳細は [31, 52] を参照）．その主な理由は，一般に代数体の整数環は Euclid 整域ではないので，$\mathbb{Z}$ 上の LLL 基底簡約を加群上に一般化することは困難である．特に，2 次元のガウス基底簡約に対応するサイズ基底簡約で行う divide-and-swap アルゴリズムを加群上で行うことができない．また，加群格子上で LLL や BKZ の基底簡約アルゴリズムを実用的に動作させるための実装基盤も，現時点では十分に整備されていない．

以上のような数多くの問題点を踏まえると，文献 [56, Appendix C] で言及されている通り，Module-LWE 問題を実際に解く際には，代数構造を利用しない $\mathbb{Z}$ 上の BKZ 基底簡約を用いるのが現時点では最も有効である．

4.5.2 イデアル格子上の SVP に対する量子アルゴリズム

イデアル格子上の SVP に対する量子アルゴリズムが提案されている [37, 22, 18, 27, 28]. しかし，文献 [28] では，Ring-LWE に対する量子攻撃に向けた障壁について言及しているが，Module-LWE では更なる障壁が生じると言及している．また，文献 [3] では，Ring-LWE から Module-LWE への帰着を構築し，それはあるパラメータにおける Ring-LWE に対する多項式時間アルゴリズムは Module-LWE に対する攻撃に変換できることを示唆している．しかしながら，実用的な観点から，この攻撃は加群格子の次元が増加するごとにかなり処理性能（効率性）が下がる．つまり，加群格子の次元の増加が暗号方式の安全性を高めることを示唆している．特に，ML-KEM-768 にこの帰着を適用すると，非常に大きな剰余と誤差を持つ Ring-LWE を導き，攻撃者に 1 個以上のサンプルを要求する．文献 [56, Appendix C] で言及されているように，イデアル格子上の SVP に対する量子アルゴリズムは，ML-KEM を含む格子暗号方式に対する実用的な攻撃につながる可能性は低いと考えられる．

第 5 章

ML-KEM に特化した解析手法とその評価に関する調査結果

本章では，ML-KEM に特化した解析手法とその評価に関する調査結果をまとめる．具体的には，ML-KEM の安全性を支える秘密ベクトル $\boldsymbol{s}$ が中心二項分布 CDB_η からサンプルされる Module-LWE 問題に対する攻撃アルゴリズムとその影響について述べる．

5.1 ML-KEM における秘密ベクトルの数え上げ計算量

2.2.2 項で述べたように，ML-KEM の安全性を支える Module-LWE 問題では，秘密ベクトル $\boldsymbol{s} \in \mathbb{Z}_q^{nk}$ の各成分は中心二項分布 CBD_η $(\eta = 2, 3)$ からサンプルされる．文献 [39] では，式 (4.21) を用いる May の秘密ベクトル探索アルゴリズム [60] を拡張し，ML-KEM における秘密ベクトル $\boldsymbol{s} \in \mathbb{Z}_q^{nk}$ を数え上げる計算量を解析している．具体的には，

$$N = nk \in \{512, 768, 1024\}$$

に対して，すべての成分が中心二項分布 CBD_η からサンプルされた秘密ベクトル $\boldsymbol{s} \in \mathbb{Z}_q^N$ を数え上げる時間・空間計算量はともに，

$$\begin{array}{ll} O\big(2^{0.36N}\big) & (\eta = 2 \text{ の場合}), \\ O\big(2^{0.37N}\big) & (\eta = 3 \text{ の場合}) \end{array} \tag{5.1}$$

と評価されている [39, Table 12, Appendix A]．一方，4.3.2 項で説明した primal 攻撃では，攻撃に必要な BKZ 基底簡約のブロックサイズ β は，表 4.1 から，おおよそ

$$\beta \approx \frac{4}{5}N$$

と見積もられる．また，BKZ 基底簡約の内部で β 次元の SVP オラクルとして用いられる篩アルゴリズムの時間計算量がおおよそ

$$O\big(2^{0.292\beta}\big)$$

であることと比較すると，式 (5.1) の数え上げ計算量は大きい．以上より，文献 [39] で提案された秘密ベクトルの数え上げアルゴリズムが，BKZ 基底簡約の内部で呼び出す篩アルゴリズムより（漸近的に）有効になることはなく，表 4.2 の ML-KEM に対する攻撃計算量に実質的な影響は与えない．

第 6 章

ML-KEM の暗号強度に関する考察

本章では，ML-KEM の暗号強度に関する考察をまとめる．第 3 章で議論したように，ML-KEM の帰着仮定である共通プリミティブであるハッシュ関数・擬似ランダム関数やデカプセル化（復号）失敗確率を利用することが実用的に困難であるため，ML-KEM の暗号強度は Module-LWE 問題の攻撃計算量に依存する．以下では，ML-KEM の安全性を支える Module-LWE 問題の攻撃計算量について考察する．

6.1 ML-KEM の安全性を支える Module-LWE 問題の攻撃計算量

ML-KEM の安全性を支える Module-LWE 問題では，秘密 $\boldsymbol{s}(X) = (s_1(X), \ldots, s_k(X)) \in R_q^k$ の成分多項式 $s_i(X)$ $(i = 1, \ldots, k)$ とノイズ $e(X) \in R_q$ のすべての $\mathbb{Z}_q$ 係数は，中心二項分布 CBD_η $(\eta \in \{2, 3\})$ からサンプルされる．現時点では，このような Module-LWE 問題に対する最良の攻撃法は，4.1 節で述べた方法により $\mathbb{Z}_q$ 上の nk 次元の LWE 問題に帰着した後，BKZ 基底簡約などの $\mathbb{Z}$ 格子上のアルゴリズムを適用するものである．また，ML-KEM の暗号文の形 (2.16) から，攻撃に利用可能な Module-LWE サンプル数は最大 $(k+1)$ であり，これらを $\mathbb{Z}_q$ 上の LWE サンプルに帰着すると，その個数は最大 $(k+1)n$ である．このサンプル数の制限により BKW 型攻撃と線形攻撃は適用できず，$\mathbb{Z}_q$ 上の LWE 問題に対する primal 攻撃と dual 攻撃のみが有効となる．また，4.4 節で議論したように，表 2.1 の ML-KEM の暗号パラメータ $nk \in \{512, 768, 1024\}$ に対しては，primal 攻撃と BKZ 基底簡約（内部 SVP オラクルは篩アルゴリズム）の組み合わせが有効である．攻撃者に有利な観点から，BKZ の progressive 化とシミュレーション [29]，dimension-for-free 技術 [32] などの効果を考慮すると，ML-KEM-512, 768, 1024 を攻撃するために必要な BKZ の最小ブロックサイズ β はそれぞれ

$$\beta = 413,\ 637,\ 894$$

と見積もられる（表 4.2 を参照）．さらに，それらの β までの progressive 型の BKZ 基底簡約において，内部 SVP オラクルとして呼び出す篩アルゴリズムの最内部にある繰り返し関数の文献 [6]

の解析に基づくゲートコストはそれぞれ

$$G = 2^{151.5},\ 2^{215.1},\ 2^{287.3} \tag{6.1}$$

と見積もられる（表 4.2 を参照）．これは，耐量子計算機暗号の NIST 標準化 [63] の安全性レベル 1, 3, 5 でそれぞれ要求される古典ゲート数

$$2^{143},\ 2^{207},\ 2^{272} \tag{6.2}$$

を上回る（詳細は [63, §4.A.5] を参照）．

■篩アルゴリズムの解析の精密化・改良による影響　式 (6.1) のゲートコスト評価は，文献 [15] の篩アルゴリズムの計算量に依存する．注意 4.3 で述べた通り，文献 [6] の理想的な近傍探索に比べて，文献 [15] の篩アルゴリズムはオーバーヘッドをもつ．文献 [13, Section 5.3, Summary] で述べられているように，篩アルゴリズムの多角的な解析の精密化と，将来的に予想されるアルゴリズム的改良を考慮すると，式 (6.1) のゲートコスト評価は $2^{-16} \sim 2^{14}$ 倍程度ずれる可能性がある．最悪の場合，式 (6.1) のゲートコスト評価は NIST 標準化で要求される式 (6.2) の古典ゲート数を下回る可能性があるが，これはあくまで攻撃者に最も有利な条件下の評価に過ぎない．実際には，文献 [56, §4.1.1] で指摘されているように，篩アルゴリズムのメモリアクセスのコストを現実的に反映した条件下で，NIST 標準化で要求される式 (6.2) の古典ゲート数は維持されると考えられる．（近年，文献 [79] で，文献 [15] の篩アルゴリズムのメモリアクセスのコストを従来のおおよそ 40% に削減する改良が提案されている．これは実用的な改良で，文献 [13, Section 5.3, Summary] で予想されている改良の範囲に収まるものと思われる．）

■最新の dual-sieve 攻撃による影響　近年，文献 [23] で符号理論のアイデアに基づく新しい dual-sieve 攻撃が提案されている．この dual-sieve 攻撃により，耐量子計算機暗号の NIST 標準化が要求する式 (6.2) の古典ゲート数よりも少なくとも $2^{3.5}$, $2^{11.9}$, $2^{12.3}$ 下回ると主張している [23, Table 5.1]．ただし，文献 [23] による解析は理想的な理論モデルに基づき，いくつかのオーバーヘッドが隠れている．また，LWE チャレンジ [30] に対する大規模な解読などで理論と実験の両面で検証されている primal 攻撃に比べて，dual 攻撃の実用的な解析は進んでおらず，dual-sieve 攻撃による実用性を検証している文献 [35] で述べられているように，dual-sieve 攻撃の成功確率は実際よりかなり高く見積もられている．これより，文献 [23] が主張する攻撃計算量評価は実際よりかなり低く見積もられている可能性が高く，primal 攻撃による式 (6.1) のゲートコスト評価には影響しないと思われる．

■ML-KEM に特化した攻撃手法による影響　ML-KEM の秘密鍵 $\boldsymbol{s}$ の成分多項式のすべての $\mathbb{Z}_q$ 係数が中心二項分布 CBD_η からサンプルされることを利用した攻撃が文献 [39] で提案されている．5.1 節で述べたように，その攻撃の計算量は primal 攻撃よりも大きいため，primal 攻撃による式 (6.1) のゲートコスト評価には影響しない．

6.2 代数構造を利用した格子アルゴリズムの影響

式 (6.1) のゲートコスト評価においては，ML-KEM の構成における基礎環 $R = \mathbb{Z}[X]/(X^n + 1)$ $(n = 256)$ の代数構造を利用していない．4.5.1 項で述べたように，環 R の代数構造を利用することで，篩アルゴリズムなどの SVP アルゴリズムの高速化が期待できる．しかし，加群格子上の BKZ 基底簡約アルゴリズムの内部 SVP オラクルとして呼び出すことにはいくつかの技術的障壁がある．具体的には，加群格子上の BKZ 基底簡約のブロックサイズ β の選択に制限があるため，primal 攻撃に最適な β を利用することが困難となる．また，加群格子上の BKZ 基底簡約アルゴリズムが，$\mathbb{Z}$ 格子上の BKZ 基底簡約と同程度の品質を持つ基底を出力するかどうかは不明である．（文献 [34] では，非構造化格子よりも加群格子上の BKZ 基底簡約はより多くのブロックサイズを必要とすると結論付けている．）さらに，加群格子上で BKZ 基底簡約を実用的に動作させるための実装基盤も現時点では十分に整備されていない．一方，イデアル格子上の SVP に対する量子アルゴリズムについても，4.5.2 項で述べたように Module-LWE に適用するには障壁があり，文献 [56, Appendix C] で言及されているように，ML-KEM に対する実用的な攻撃に繋がる可能性は低い．以上のように，ML-KEM に対する攻撃において，代数構造を利用した格子アルゴリズムは現時点で有効とはならない．

参考文献

[1] Martin R Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In *Advances in Cryptology–EUROCRYPT 2017*, volume 10211 of *Lecture Notes in Computer Science*, pages 103–129. Springer, 2017.

[2] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the {LWE, NTRU} schemes! In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 351–367. Springer, 2018. `https://estimate-all-the-lwe-ntru-schemes.github.io/docs/`.

[3] Martin R. Albrecht and Amit Deo. Large modulus Ring-LWE $\geq$ Module-LWE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 267–296. Springer, 2017.

[4] Martin R Albrecht and Léo Ducas. Lattice attacks on NTRU and LWE: A history of refinements. In *Computational Cryptography: Algorithmic Aspects of Cryptology*, volume 469 of *London Mathematical Society Lecture Note Series*, pages 15–40. Cambridge University Press, 2021.

[5] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 717–746. Springer, 2019.

[6] Martin R Albrecht, Vlad Gheorghiu, Eamonn W Postlethwaite, and John M Schanck. Estimating quantum speedups for lattice sieves. In *Advances in Cryptology–ASIACRYPT 2020*, volume 12492 of *Lecture Notes in Computer Science*, pages 583–613. Springer, 2020.

[7] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 297–322. Springer, 2017.

[8] Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[9] Martin R Albrecht and Yixin Shen. Quantum augmented dual attack. *arXiv preprint arXiv:2205.13983*, 2022. `https://arxiv.org/pdf/2205.13983`.

[10] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*, pages 327–343. USENIX Association, 2016.

[11] Yoshinori Aono, Yuntao Wang, Takuya Hayashi, and Tsuyoshi Takagi. Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator. In *Advances in Cryptology–EUROCRYPT 2016*, volume 9665 of *Lecture Notes in Computer Science*, pages 789–819. Springer, 2016.

[12] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *International Colloquium on Automata, Languages, and Programming (ICALP 2011)*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.

[13] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancréde Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber – algorithm specifications and supporting documentation (version 3.02). `https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf`, 2021.

[14] Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary LWE. In Willy Susilo and Yi Mu, editors, *Information Security and Privacy - 19th Australasian Conference, ACISP 2014, Wollongong, NSW, Australia, July 7-9, 2014. Proceedings*, volume 8544 of *Lecture Notes in Computer Science*, pages 322–337. Springer, 2014.

[15] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the twenty-seventh annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016)*, pages 10–24. SIAM, 2016.

[16] Buchmann Johannes Bernstein, Daniel J and Erik Dahmen. *Post-quantum cryptography*. Springer, 2009.

[17] Lei Bi, Xianhui Lu, Junjie Luo, and Kunpeng Wang. Hybrid dual and meet-LWE attack.

In *Information Security and Privacy (ACISP 2022)*, volume 13494 of *Lecture Notes in Computer Science*, pages 168–188. Springer, 2022.

[18] Jean-François Biasse and Fang Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields. In *ACM-SIAM Symposium on Discrete Algorithms (SODA 2016)*, pages 893–902. SIAM, 2016.

[19] Nina Bindel, Johannes Buchmann, Florian Göpfert, and Markus Schmidt. Estimation of the hardness of the learning with errors problem with a restricted number of samples. *Journal of Mathematical Cryptology*, 13(1):47–67, 2019.

[20] Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. On the hardness of module learning with errors with short distributions. *Journal of Cryptology*, 36(1):1–70, 2023.

[21] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

[22] Peter Campbell, Michael Groves, and Dan Shepherd. Soliloquy: A cautionary tale. ETSI 2nd quantum-safe crypto workshop, 2014, 2014. `https://docbox.etsi.org/workshop/2014/201410_CRYPTO/S07_Systems_and_Attacks/S07_Groves_Annex.pdf`.

[23] Kevin Carrier, Charles Meyer-Hilfiger, Yixin Shen, and Jean-Pierre Tillich. Assessing the impact of a variant of MATZOV's dual attack on Kyber. In *Advances in Cryptology–CRYPTO 2025*, volume 16000 of *Lecture Notes in Computer Science*, pages 444–476. Springer, 2025.

[24] Kevin Carrier, Yixin Shen, and Jean-Pierre Tillich. Faster dual lattice attacks by using coding theory. 2024. `https://hal.science/hal-04519755/document`.

[25] Yuanmi Chen. *Réduction de réseau et sécurité concrete du chiffrement completement homomorphe*. PhD thesis, Paris 7, 2013.

[26] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2011.

[27] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In *Advances in Cryptology–EUROCRYPT 2016*, volume 9666 of *Lecture Notes in Computer Science*, pages 559–585. Springer, 2016.

[28] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Short stickelberger class relations and application to ideal-svp. In *Advances in Cryptology–EUROCRYPT 2017*, volume 10210 of *Lecture Notes in Computer Science*, pages 324–348. Springer, 2017.

[29] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In *Advances in Cryptology–CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pages 329–358. Springer, 2020.

[30] TU Darmstadt and UC San Diego. LWE challenge. `https://www.latticechallenge.org/lwe_challenge/challenge.php`. 2025-12-31 閲覧.

[31] Gabrielle De Micheli and Daniele Micciancio. A fully classical LLL algorithm for modules. *Cryptology ePrint Archive, Paper 2022/1356*, 2022. `https://eprint.iacr.org/2022/1356.pdf`.

[32] Léo Ducas. Shortest vector from lattice sieving: a few dimensions for free. In *Advanves in Cryptology–EUROCRYPT 2018*, volume 10820 of *Lecture Notes in Computer Science*, pages 125–145. Springer, 2018.

[33] Léo Ducas. Estimating the hidden overheads in the BDGL lattice sieving algorithm. In *Post-Quantum Cryptography (PQCrypto 2022)*, volume 13512 of *Lecture Notes in Computer Science*, pages 480–497. Springer, 2022.

[34] Léo Ducas, Lynn Engelberts, and Paola de Perthuis. Predicting module-lattice reduction. In *Advances in Cryptology–ASIACRYPT 2025*, volume 16247 of *Lecture Notes in Computer Science*, pages 133–166. Springer, 2025.

[35] Léo Ducas and Ludo N Pulles. Does the dual-sieve attack on learning with errors even work? In *Advances in Cryptology–EUROCYPT 2023*, volume 14083 of *Lecture Notes in Computer Science*, pages 37–69. Springer, 2023.

[36] Jan-Pieter D' anvers and Senne Batsleer. Multitarget decryption failure attacks and their application to Saber and Kyber. In *Public-Key Cryptography (PKC 2022)*, volume 13177 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2022.

[37] Kirsten Eisenträger, Sean Hallgren, Alexei Kitaev, and Fang Song. A quantum algorithm for computing the unit group of an arbitrary degree number field. In *ACM Symposium on Theory of computing (STOC 2014)*, pages 293–302, 2014.

[38] Nicolas Gama and Phong Q Nguyen. Predicting lattice reduction. In *Advances in Cryptology–EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51. Springer, 2008.

[39] Timo Glaser and Alexander May. How to enumerate LWE keys as narrow as in Kyber/Dilithium. In *Cryptology and Network Security (CANS 2023)*, volume 14342 of *Lecture Notes in Computer Science*, pages 75–100. Springer, 2023.

[40] Qian Guo and Thomas Johansson. Faster dual lattice attacks for solving LWE with applications to CRYSTALS. In *Advances in Cryptology–ASIACRYPT 2021*, volume 13093 of *Lecture Notes in Computer Science*, pages 33–62. Springer, 2021.

[41] Minki Hhan, Jiseung Kim, Changmin Lee, and Yongha Son. Let's meet ternary keys on Babai's plane: A hybrid of lattice-reduction and meet-LWE. *Cryptology ePrint Archive,*

Paper 2022/1473, 2022. https://eprint.iacr.org/2022/1473.

[42] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371. Springer, 2017.

[43] Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In *Advances in Cryptology–CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 150–169. Springer, 2007.

[44] Internet Engineering Task Force (IETF). Post-quantum cryptography for engineers, February 2024. https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqc-engineers-03.

[45] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo, and Joel I. Seiferas, editors, *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 193–206. ACM, 1983.

[46] Jiseung Kim, Changmin Lee, and Yongha Son. Worst-case analysis of lattice enumeration algorithm over modules. *Cryptology ePrint Archive, Paper 2025/480*, 2025. https://eprint.iacr.org/2025/480.pdf.

[47] Paul Kirchner and Pierre-Alain Fouque. An improved BKW algorithm for LWE with applications to cryptography and lattices. In *Advances in Cryptology–CRYPTO 2015*, volume 9215 of *Lecture Notes in Computer Science*, pages 43–62. Springer, 2015.

[48] Thijs Laarhoven. Search problems in cryptography: from fingerprinting to lattice sieving. *PhD thesis, Eindhoven University of Technology*, 2016. https://pure.tue.nl/ws/files/14673128/20160216_Laarhoven.pdf.

[49] Thijs Laarhoven and Artur Mariano. Progressive lattice sieving. In *Post-Quantum Cryptography (PQCrypto 2018)*, volume 10786 of *Lecture Notes in Computer Science*, pages 292–311. Springer, 2018.

[50] Thijs Laarhoven, Michele Mosca, and Joop Van De Pol. Finding shortest lattice vectors faster using quantum search. *Designs, Codes and Cryptography*, 77(2):375–400, 2015.

[51] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015.

[52] Changmin Lee, Alice Pellet-Mary, Damien Stehlé, and Alexandre Wallet. An LLL algorithm for module lattices. In *Advances in Cryptology–ASIACRYPT 2019*, volume 11922 of *Lecture Notes in Computer Science*, pages 59–90. Springer, 2019.

[53] A. K. Lenstra, H. W. Lenstra, and Lovász L. Factoring polynomials with rational coeffi-

cients. *Mathematische Annalen*, 261(4):515–534, 12 1982.

[54] Cathy Yuanchen Li, Jana Sotáková, Emily Wenger, Mohamed Malhou, Evrard Garcelon, François Charton, and Kristin Lauter. SalsaPicante: A machine learning attack on LWE with binary secrets. In *ACM SIGSAC Conference on Computer and Communications Security (ACM CCS 2023)*, pages 2606–2620, 2023.

[55] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.

[56] National lnstitute of Standards and Technology (NIST). NIST IR 8413-upd1: Status report on the third round of the NIST post-quantum cryptography standardization process. 2022. `https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413-upd1.pdf`.

[57] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, 2010.

[58] Artur Mariano, Thijs Laarhoven, and Christian Bischof. A parallel variant of LDSieve for the SVP on lattices. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 23–30. IEEE, 2017.

[59] MATZOV. Report on the security of LWE: Improved dual lattice attack, 2022. `https://zenodo.org/records/6412487`.

[60] Alexander May. How to meet ternary LWE keys. In *Advances in Cryptology–CRYPTO 2021*, volume 12826 of *Lecture Notes in Computer Science*, pages 701–731. Springer, 2021.

[61] Tamalika Mukherjee and Noah Stephens-Davidowitz. Lattice reduction for modules, or how to reduce ModuleSVP to ModuleSVP. In *Advances in Cryptology–CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pages 213–242. Springer, 2020.

[62] National Institute of Standards and Technology (NIST). FIPS 202: SHA-3 standard: Permutation-based hash and extendable-output functions. `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf`, August 2015.

[63] National Institute of Standards and Technology (NIST). Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, 2016. `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf`.

[64] National Institute of Standards and Technology (NIST). FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. `https://nvlpubs.nist.gov/nistpubs/FIPS/`

NIST.FIPS.203.pdf, August 13, 2024.

[65] Eamonn W. Postlethwaite and Fernando Virdia. On the success probability of solving unique SVP via BKZ. In Juan A. Garay, editor, *Public-Key Cryptography - PKC 2021 - 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10-13, 2021, Proceedings, Part I*, volume 12710 of *Lecture Notes in Computer Science*, pages 68–98. Springer, 2021.

[66] Gokulnath Rajendran, Prasanna Ravi, Jan-Pieter D'anvers, Shivam Bhasin, and Anupam Chattopadhyay. Pushing the limits of generic side-channel attacks on LWE-based KEMs-parallel PC oracle attacks on Kyber KEM and beyond. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(2):418–446, 2023.

[67] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93. ACM, 2005.

[68] Oded Regev. The learning with errors problem (invited survey). In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, USA, June 9-12, 2010*, pages 191–204. IEEE Computer Society, 2010.

[69] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 520–551. Springer, 2018.

[70] Claus Peter Schnorr. Lattice reduction by random sampling and birthday methods. In *Symposium on Theoretical Aspects of Computer Science (STACS 2003)*, volume 2607 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2003.

[71] Claus-Peter Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66:181–199, 1994.

[72] Samuel Stevens, Emily Wenger, Cathy Li, Niklas Nolte, Eshika Saxena, François Charton, and Kristin Lauter. Salsa Fresca: Angular embeddings and pre-training for ML attacks on learning with errors. *arXiv preprint arXiv:2402.01082*, 2024. https://arxiv.org/pdf/2402.01082.

[73] Yutaro Tanaka, Rei Ueno, Keita Xagawa, Akira Ito, Junko Takahashi, and Naofumi Homma. Multiple-valued plaintext-checking side-channel attacks on post-quantum KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(3):473–503, 2023.

[74] Emily Wenger, Mingjie Chen, Francois Charton, and Kristin E Lauter. SALSA: Attack-

ing lattice cryptography with transformers. *Advances in Neural Information Processing Systems (NeurIPS 2022)*, 35:34981–34994, 2022.

[75] Emily Wenger, Eshika Saxena, Mohamed Malhou, Ellie Thieu, and Kristin Lauter. Benchmarking attacks on learning with errors. In *IEEE Symposium on Security and Privacy (SP)*, pages 279–297. IEEE, 2025.

[76] Han Wu and Guangwu Xu. Enhancing the dual attack against MLWE: Constructing more short vectors using its algebraic structure. *Cryptology ePrint Archive, Paper 2022/1661*, 2022. `https://eprint.iacr.org/2022/1661`.

[77] Keita Xagawa, Akira Ito, Rei Ueno, Junko Takahashi, and Naofumi Homma. Fault-injection attacks against NIST's post-quantum cryptography round 3 KEM candidates. In *Advances in Cryptology–ASIACRYPT 2021*, volume 13091 of *Lecture Notes in Computer Science*, pages 33–61. Springer, 2021.

[78] Yang Yu and Léo Ducas. Second order statistical behavior of LLL and BKZ. In Carlisle Adams and Jan Camenisch, editors, *Selected Areas in Cryptography (SAC 2017) - 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers*, volume 10719 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2017.

[79] Ziyu Zhao, Jintai Ding, and Bo-Yin Yang. Sieving with streaming memory access. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(2):362–384, 2025.

CRYPTREC: ML-KEM Evaluation Report

PQShield

Thomas Espitau, Shuichi Katsumata, Niels Samwel, Thom Wiggers, Wessel van Woerden, Timo Zijlstra

Executive Summary

Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) is a lattice-based key encapsulation mechanism (KEM), standardized by the U.S. National Institute of Standards and Technology (NIST) as Federal Information Processing Standard (FIPS) 203 [Nat24b] in August, 2024. It is derived from CRYSTALS-KYBER [Ava+21] and designed to provide confidentiality against attackers equipped with large-scale quantum computers. The evaluation covers the scheme's design rationale, provable security guarantees, concrete hardness against cryptanalytic attacks, implementation security regarding side-channels, and its applicability in secure network protocols such as TLS 1.3. The following is a summary of ML-KEM.

Construction: The scheme is based on the Fujisaki-Okamoto transform, transforming a base INDCPA secure public key encryption scheme called KPKE into an IND-CCA secure KEM. It has a non-zero but negligible decryption failure rate (2^{-139} to 2^{-174}), which is deemed operationally safe and does not pose a security risk under correct implementation.

Provable Security: It's security is based on the *Module Learning with Errors* (MLWE) problem, a structured variant of the standard LWE problem that balances efficiency with security. Tight reductions exist in the classical random oracle model (ROM). In the quantum ROM, while reductions are asymptotically looser, they still provide sufficient confidence in the scheme' s resistance to quantum adversaries.

Security Margins: Concrete cost estimates for the best known attacks (using conservative cost models such as Core-SVP and MATZOV) indicate that all ML-KEM parameter sets meet or exceed their targeted NIST security levels.

Algebraic Structure: Attacks exploiting the algebraic structure of module lattices (e.g., ideal lattice attacks) were analyzed. The report concludes that for the specific module ranks used in ML-KEM ($k \in \{2, 3, 4\}$), these structured attacks do not outperform generic lattice reduction techniques.

Performance: ML-KEM demonstrates excellent performance in software and hardware, significantly outperforming traditional elliptic-curve cryptography (ECC) in computation time, although key and ciphertext sizes are larger.

Side-Channel Attacks: Unprotected implementations are vulnerable to side-channel attacks, including differential power analysis and timing attacks on decryption failures.

Countermeasures: Effective countermeasures such as masking (arithmetic and boolean) and shuffling are available. While these introduce performance overheads, they are necessary for high-assurance deployments in hostile environments.

Application to TLS 1.3: ML-KEM is fully viable for TLS 1.3. The larger key sizes (encapsulation keys and ciphertexts) increase handshake traffic but result in negligible impact on overall connection latency in most network scenarios. The report discuss the use of "hybrid" key exchange mechanisms, combining ML-KEM with traditional elliptic-curve Diffie-Hellman (ECDH), as a robust transition strategy to mitigate risks associated with new cryptographic primitives.

Based on current cryptanalytic knowledge, ML-KEM is a robust and secure post-quantum KEM. Its theoretical foundations are sound, and its parameter sets provide adequate security margins against known classical and quantum threats. While it offers faster processing speeds compared to traditional cryptographic schemes, the increased key and ciphertext sizes may present implementation challenges for memory-constrained devices. However, it is judged to be viable for use in other general-purpose applications without issues. Furthermore, provided that implementations are rigorously protected against side-channel attacks, it is suitable for widespread deployment in government and critical infrastructure systems.

エグゼクティブ・サマリー

Module-Lattice-Based Key-Encapsulation Mechanism（ML-KEM）は，モジュール格子に基づく鍵カプセル化メカニズム（KEM）であり，2024 年 8 月に，米国国立標準技術研究所（NIST）によって連邦情報処理標準 FIPS203 として標準化された．本方式は CRYSTALS-KYBER [Ava+21] から派生したものであり，大規模な量子計算機に対しても安全であるように設計されている．本評価は，方式の設計根拠，証明可能安全性，暗号解読攻撃に対する具体的な困難性，サイドチャネルに関する実装セキュリティ，および TLS 1.3 などのインターネット通信を保護する暗号化プロトコルへの適用可能性を網羅している．以下，評価内容の概要を述べる．

構成：本方式は藤崎・岡本変換に基づいており，IND-CPA 安全な公開鍵暗号方式である KPKE を，IND-CCA 安全な KEM に変換している．本方式はゼロではないが無視可能な復号失敗率（2^{-139} から 2^{-174}）を持つ．これは運用上安全であるとみなされ，正しく実装されている限り安全性を損ねるものではない．

証明可能安全性：安全性は，*Module Learning with Errors*（MLWE）問題に基づいている．これは，効率性と安全性のバランスをとるために，標準的な LWE 問題を構造化した変種である．古典的ランダムオラクルモデル（ROM）においては緊密な帰着が存在する．量子 ROM においては，帰着は漸近的に緩くなるものの，量子敵対者に対する耐性について十分な信頼性を与えるものである．

セキュリティマージン：既知の最良の攻撃に対する具体的なコスト見積もり（Core-SVP や MATZOV などの保守的なコストモデルを使用）は，すべての ML-KEM パラメータセットが，目標とする NIST 安全性レベルを満たしているか，あるいは上回っていることを示している．

代数的構造：モジュール格子の代数的構造を利用する攻撃（例：イデアル格子攻撃）についても分析を行った．本報告書では，ML-KEM で使用される特定のモジュールランク（$k \in \{2, 3, 4\}$）において，これらの構造を利用した攻撃は，構造を利用しない一般的な格子基底簡約アルゴリズムを上回るものではないと結論付ける．

性能：ML-KEM はソフトウェアおよびハードウェアにおいて優れたパフォーマンスを示し，鍵や暗号文のサイズは大きくなるものの，計算時間については従来の楕円曲線暗号（ECC）より大幅に高速である．

サイドチャネル攻撃：対策が施されていない実装は，電力差分解析や復号失敗に対するタイミング攻撃などのサイドチャネル攻撃に対して脆弱である．

対策：マスキング（算術およびブール）やシャフリングといった効果的な対策が利用可能である．これらは効率性を損なう対策だが，敵対的な環境における安全性を保証するために不可欠である．

TLS 1.3 への適用：ML-KEM は TLS 1.3 において十分に利用可能である．鍵長（カプセル化鍵および暗号文）の増大によりハンドシェイクのデータ送量は増加するが，多くのネットワークシナリオにおいて，全体的な接続遅延時間への影響は無視できる範囲である．本報告書では，新しい暗号プリミティブに伴うリスクを軽減するための堅牢な移行戦略として，ML-KEM と従来の楕円曲線 Diffie-Hellman（ECDH）を組み合わせたハイブリッド鍵交換メカニズムの性能についても言及する．

現在の暗号解析に関する知見に基づき，ML-KEM は堅牢かつ安全な耐量子 KEM であると考える．その理論的基盤は健全であり，各パラメータセットは既知の古典，および，量子の暗号解析アルゴリズムに対して十分なセキュリティマージンを提供している．計算時間は従来の暗号方式に比べより高速である一方で，鍵や暗号文長は増加するため，メモリ制約があるデバイス等においては実装上の課題が生じる可能性がある．しかし，それ以外の用途においては問題なく利用できると判断される．また，サイドチャネル攻撃に対して厳密に保護された実装が行われることを前提として，政府および重要インフラシステムへの広範な展開にも適している．

Contents

1 Introduction and Outline

As part of the Post-Quantum Cryptography (PQC) project initiated by the National Institute of Standards and Technology (NIST), the United States has started a multi-year effort to identify, optimize and eventually standardize cryptographic primitives that remain secure in the presence of large-scale quantum computers. NIST announced in 2022 its intention to standardize a lattice-based Key Encapsulation Mechanism (KEM), CRYSTALS-KYBER [Ava+21], as one of the first post-quantum public-key primitives. In 2024, this effort culminated in the publication of the *Module-Lattice-Based Key-Encapsulation Mechanism* (ML-KEM) as Federal Information Processing Standards (FIPS) 203 [Nat24b], which specifies a slightly modified but fully standardized version of CRYSTALS-KYBER. From the viewpoint of government agencies and operators of critical infrastructure, ML-KEM is intended to serve as a general-purpose building block for post-quantum key establishment in a broad range of protocols and deployment scenarios.

On the technical side, ML-KEM is a highly optimized module version [LS15] of the LPR encryption scheme, which is based on the Ring-LWE problem [LPR10]. The LPR construction itself rests on a long and steadily growing line of work on lattice-based encryption schemes [Reg05; LP11] that relate the security of practical cryptosystems to well-studied worst-case problems on lattices. ML-KEM inherits this foundation while making specific design choices — such as employing a module structure rather than fully unstructured lattices — that aim to balance performance, implementation simplicity, and confidence in the underlying hardness assumptions. In particular, ML-KEM has been designed to be efficient on a wide range of platforms, from general-purpose CPUs to constrained devices, while still admitting constant-time implementations and side-channel hardened variants suitable for high-assurance environments.

From a security perspective, ML-KEM is known to satisfy IND-CCA security, a strong notion of active security in which a ciphertext ct* leaks nothing about the encapsulated key, even to an adversary that is given access to an oracle decrypting any ciphertext other than ct*. This property is particularly important in realistic deployment settings where adversaries may interact with decryption endpoints via network protocols, inject or modify ciphertexts in transit, or exploit protocol error messages.

The purpose of the present report is to give a structured and self-contained overview of ML-KEM from both a theoretical and a practical standpoint.

Outline of Report.

Section 2. We begin by recalling the necessary background and notation, including basic concepts from lattice-based cryptography, security notions for public-key encryption and KEMs, and the role of the (quantum) random oracle model in the analysis of ML-KEM.

Section 3. We then provide a high-level description of ML-KEM itself, starting from the underlying IND-CPA secure lattice-based encryption scheme and explaining how the Fujisaki-Okamoto transform is used to obtain an IND-CCA secure KEM. In this part we also summarize the standardized parameter sets and briefly discuss their intended security levels and performance characteristics.

Section 4. Next, we review the provable security results for the IND-CCA security of ML-KEM, outlining the main reductions from standard lattice problems and clarifying in which models and under which assumptions these guarantees hold. Where relevant, we distinguish between classical and quantum adversaries. We further discuss other advanced security features ML-KEM is known to satisfy.

Section 5. We then turn to practical cryptanalysis and concrete security estimates. This includes an overview of the best known quantumu attacks on the underlying lattice problems, the impact of decryption failures and weak-key considerations, and the resulting security margins for the parameter sets selected in FIPS 203.

Section 6. We address implementation details, focusing on both performance and physical security. We describe the algorithmic building blocks and analyze vulnerabilities to side-channel attacks. We further discuss necessary countermeasures, such as masking and shuffling, and present performance results in hardware that illustrate the cost of these protections.

Section 7. Finally, we examine the application of ML-KEM in the Transport Layer Security (TLS) 1.3 protocol. We evaluate the integration of ML-KEM and hybrid (Post-Quantum/Traditional) key exchange mechanisms, comparing them to traditional ECDH in terms of computation time and bandwidth. The section also covers real-world deployment challenges and surveys the availability of ML-KEM in popular cryptographic libraries.

2 Preliminaries

To keep the report self-contained, we provide a minimal presentation of the preliminaries. More details are provided in the FIPS 203 publication [Nat24b] and references therein.

2.1 Notations

We write $A\|B$ for the concatenation of values A and B. Boldface lower- and upper-case symbols such as $\mathbf{a}$ and $\mathbf{M}$ denote column vectors and matrices, respectively. $\mathbf{a}^\top$ and $\mathbf{M}^\top$ denote their transposes. For vectors $\mathbf{a}$ and $\mathbf{b}$, we may sometimes use $\langle \mathbf{a}, \mathbf{b} \rangle$ to denote the inner product $\mathbf{a}^\top \cdot \mathbf{b}$. For a real $x \in \mathbb{R}$, $\lceil x \rfloor$ is the rounding of x to the nearest integer, where if $x = y + 1/2$ for some $y \in \mathbb{Z}$, then $\lceil x \rfloor = y + 1$. For a vector $\mathbf{v} \in \mathbb{R}^d$, $\|\mathbf{v}\|$ denotes the L_2 norm.

For a set $\mathcal{M}$, we write $m \xleftarrow{\$} \mathcal{M}$ for the process of sampling m uniformly at random from $\mathcal{M}$. For a randomized algorithm $f : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$, we write $y \xleftarrow{\$} f(x)$ for the process of executing f on x with randomness $r \xleftarrow{\$} \mathcal{R}$. $\Pr[E : X]$ denotes the probability that event E occurs over the randomness of variable X, typically used to express the advantage of an adversary winning some security game.

When we say an algorithm $\mathcal{A}$ is efficient, we mean that $\mathcal{A}$ is either a classical probabilistic polynomial-time (PPT) algorithm or a quantum polynomial-time (QPT) algorithm. A function $f : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ is said to be negligible if for all c, there exists λ_0 such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_0$, i.e., it decays faster than the inverse of any polynomial at infinity.

2.2 The NIST Security Level

In the NIST PQC standardization project, NIST asked submitters to categorize the instances of their submitted schemes into one of five security levels, which relate to the difficulty of breaking the symmetric schemes AES and SHA-2. We provide the definitions given by NIST in Table 1 [Nat16].

Table 1: NIST's categorization of security levels.

Level	Security description
1	At least as hard to break as AES128 (exhaustive key search)
2	At least as hard to break as SHA256 (collision search)
3	At least as hard to break as AES192 (exhaustive key search)
4	At least as hard to break as SHA384 (collision search)
5	At least as hard to break as AES256 (exhaustive key search)

NIST asked submitters to focus on levels 1–3, leaving levels 4 and 5 for high-security instances. As the project progressed, most submissions settled on providing parameter sets for security levels 1, 3, and 5. For ML-KEM, parameter sets have been defined at security levels 1, 3, and 5. These can be compared to AES128, AES192, and AES256. Compared to elliptic-curve-based schemes, these levels should (roughly) be equivalent to the security given by the NIST curves P-256, P-384, and P-521 against classical (non-quantum) adversaries.

It is worth noting that choosing an appropriate security level means balancing security with operational costs. ML-KEM at security levels 3 and 5 require (significantly) more computation time and memory, and has larger messages than security level 1. Based on the current state of the art

in cryptanalysis, which this report will cover in detail, breaking ML-KEM at level 1 should be out of reach of any conceivable adversary. Higher security levels can, however, provide some insurance against advances in cryptanalysis.

2.3 Lattices

2.3.1 Cyclotomic Rings

Let n be a power-of-two integer and q a prime. Let $R = \mathbb{Z}[X]/(X^n + 1)$ be the cyclotomic ring of degree n and denote $R_q = R \bmod q$. For ML-KEM we consider $n = 256$ unless otherwise stated.

2.3.2 (Module)-Lattices

For a matrix $\mathbf{B} \in R^{k \times \ell}$ with linearly independent columns, we define the (right) R-submodule

$$L := \mathbf{B} \cdot R^\ell = \{\mathbf{B} \cdot x : x \in R^\ell\} \subseteq R^k$$

and call L an (integer) R-module lattice of rank ℓ, with $\mathbf{B}$ a basis of L. Intuitively, L consists of all R-linear combinations of the columns of $\mathbf{B}$, so changing the basis $\mathbf{B}$ changes the way we describe vectors in L but not the underlying set.

When $qR^k \subseteq L \subseteq R^k$ for some integer modulus $q \geqslant 2$, we call L a *q-ary lattice*. In this case we can identify vectors in L with elements of R_q^k by reducing their coefficients modulo q. We will often tacitly work with this reduction and think of vectors of L as living in R_q^k.

In the special case $n = 1$ we have $R = \mathbb{Z}$, and we recover classical (unstructured) integer lattices $L \subseteq \mathbb{Z}^k$. For a basis $\mathbf{B}$ of such a lattice L, we write

$$\det(L) := \left|\sqrt{\det(\mathbf{B}^\top \cdot \mathbf{B})}\right|$$

for the lattice determinant (the Euclidean volume of a fundamental parallelepiped of L), and

$$\lambda_1(L) := \min_{\mathbf{y} \in L \setminus \{0\}} \|\mathbf{y}\|$$

for the *first minimum*, i.e., the length of a shortest nonzero lattice vector.

Given a basis $\mathbf{B} \in R^{k \times \ell}$ and its associated anti-circulant matrix $\mathbf{M} \in \mathbb{Z}^{nk \times n\ell}$ (obtained via the coefficient embedding $R^k \hookrightarrow \mathbb{Z}^{nk}$), we associate to the R-module lattice $L := \mathbf{B} \cdot R^\ell$ of rank ℓ the unstructured integer lattice

$$L' := \mathbf{M} \cdot \mathbb{Z}^{n\ell} \subseteq \mathbb{Z}^{nk}$$

of rank $n\ell$. We then *define* the determinant and first minimum of L via those of L' by setting

$$\det(L) := \det(L') \qquad \text{and} \qquad \lambda_1(L) := \lambda_1(L')\,.$$

To measure how "orthogonal" a basis of L' is, we will repeatedly use the Gram–Schmidt orthogonalization of the columns of $\mathbf{M}$. Writing $\mathbf{b}_1, \ldots, \mathbf{b}_{n\ell}$ for these column vectors, their Gram–Schmidt orthogonalization is the sequence $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_{n\ell}$ defined recursively by

$$\tilde{\mathbf{b}}_1 := \mathbf{b}_1, \qquad \tilde{\mathbf{b}}_i := \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \tilde{\mathbf{b}}_j \quad \text{for } i \geqslant 2,$$

where

$$\mu_{i,j} := \frac{\mathbf{b}_i^\top \cdot \tilde{\mathbf{b}}_j}{\tilde{\mathbf{b}}_j^\top \cdot \tilde{\mathbf{b}}_j}.$$

By construction, the vectors $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_{n\ell}$ are pairwise orthogonal in $\mathbb{R}^{nk}$, and each $\mathbf{b}_i$ lies in the span of $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_i$. We write $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_{n\ell}$ for this Gram–Schmidt family throughout, and we will often use the norms $\|\tilde{\mathbf{b}}_i\|$ as a convenient quantitative measure of the quality of the basis.

2.3.3 Rounding

For an even (resp. odd) positive integer q, we define $x' = x \bmod^{\pm} q$ to be the unique element x' in the range $-\frac{q}{2} < x' \leqslant \frac{q}{2}$ (resp. $-\frac{q-1}{2} < x' \leqslant \frac{q-1}{2}$) such that $x' = x \bmod q$. For any positive integer q, we define $x' = x \bmod^{+} q$ to be the unique element x' in the range $0 \leqslant x' < q$ such that $x' = x \bmod q$. We simply write $x \bmod q$ when the representation is not important. Also, for an element $x \in \mathbb{Q}$, $\lceil x \rfloor$ denotes rounding to the nearest integer, where in case of a tie, we take the larger integer.

2.3.4 Sizes of Elements

For an element $x \in \mathbb{Z}_q$, we write $\|x\|_\infty$ to mean $|x \bmod^{\pm} q|$. Using this, we define the L_∞ and L_2 norms for

$$x(X) = \sum_{i=0}^{n-1} x_i X^i \in R_q$$

as

$$\|x\|_\infty = \max_i \|x_i\|_\infty, \quad \|x\| = \sqrt{\|x_0\|_\infty^2 + \cdots + \|x_{n-1}\|_\infty^2}.$$

Similarly, for vectors of polynomial ring elements $\mathbf{x} = (x_0, \ldots, x_{k-1}) \in R_q^k$, we define

$$\|\mathbf{x}\|_\infty = \max_i \|x_i\|_\infty, \quad \|\mathbf{x}\| = \sqrt{\|x_0\|^2 + \cdots + \|x_{k-1}\|^2}.$$

2.3.5 Compression and Decompression

We define the following compression and decompression algorithms for positive integers d and q such that $d < \lceil \log_2(q) \rceil$:

$$\begin{aligned} \mathsf{Compress}_d : \mathbb{Z}_q &\longrightarrow \mathbb{Z}_{2^d} \\ x &\longmapsto \left\lceil \frac{2^d}{q} \cdot x \right\rfloor \bmod^{+} 2^d. \end{aligned} \tag{1}$$

$$\begin{aligned} \mathsf{Decompress}_d : \mathbb{Z}_{2^d} &\longrightarrow \mathbb{Z}_q \\ y &\longmapsto \left\lceil \frac{q}{2^d} \cdot y \right\rfloor. \end{aligned} \tag{2}$$

For these functions, we have the following:

Lemma 2.1. *Let d and q be positive integers such that $d < \lceil \log_2(q) \rceil$. Then, for any $x \in \mathbb{Z}_q$, we have*

$$\left| x' - x \bmod^{\pm} q \right| \leqslant \left\lceil \frac{q}{2^{d+1}} \right\rfloor,$$

where $x' = \mathsf{Decompress}_d(\mathsf{Compress}_d(x))$.

When $\mathsf{Compress}_d$ or $\mathsf{Decompress}_d$ is used with $x \in R_q$ or $\mathbf{x} \in R_q^k$, the procedure is applied to each coefficient individually.

2.3.6 Hardness Assumption

The security of ML-KEM is based on the so-called Module Learning with Errors problem (MLWE).

Informal description of LWE. The (module) Learning with Errors (LWE/MLWE) problems ask us to recover a hidden linear relation from many *noisy* linear equations. In the simplest LWE setting, there is a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ and one is given samples of the form

$$(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q, \qquad b_i = \mathbf{a}_i^\top \cdot \mathbf{s} + e_i \bmod q,$$

where each $\mathbf{a}_i$ is uniform in $\mathbb{Z}_q^n$ and each e_i is a small "error" drawn from a narrow distribution [Reg05]. The goal is either to distinguish such samples from uniformly random pairs, or to recover the secret $\mathbf{s}$. Thus, LWE can be viewed as the problem of solving a linear system whose right-hand side has been perturbed by small but unknown noise. Conceptually, the problem is hard because the noise forbids to use algebraci algorithms such as Gaussian elimination. In the modular setting small errors can wrap around modulo q and make the resulting distribution of the b_i statistically very close to uniform.

Module Learning with error. The Module Learning with Errors (MLWE) problem is a structured variant of LWE in which the vectors and matrices live in the polynomial residue ring $R_q = \mathbb{Z}_q[X]/(X^n+1)$ and one works with k-dimensional modules over R_q instead of plain $\mathbb{Z}_q$-vector spaces [LS15]. This allows more compact keys and efficient use of the number-theoretic transform, while preserving essentially the same hardness guarantees via reductions from worst-case module lattice problems. From a lattice perspective, MLWE interpolates between the highly structured ideal lattices arising from RLWE and the general lattices underlying plain LWE [LS15].

Definition 2.2 (MLWE)**.** *Let m, k, q be integers and let χ be a probability distribution over R_q. The advantage of an adversary $\mathcal{A}$ against the* Module Learning with Errors $\mathsf{MLWE}_{q,m,k,\chi}$ *problem is defined as:*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MLWE}}(1^\lambda) = \left| \Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{A}\cdot\mathbf{s}+\mathbf{x}) = 1\right] - \Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{b}) = 1\right] \right|,$$

where $(\mathbf{A}, \mathbf{b}, \mathbf{s}, \mathbf{x}) \xleftarrow{\$} R_q^{m\times k} \times R_q^m \times \chi^k \times \chi^m$. The $\mathsf{MLWE}_{q,m,k,\chi}$ assumption states that any efficient adversary $\mathcal{A}$ has negligible advantage.

We consider a slight variant where the "secret" $\mathbf{s}$ and "noise" $\mathbf{x}$ are sampled from different distributions χ and χ', respectively. We denote this as the $\mathsf{MLWE}_{q,m,k,\chi,\chi'}$ assumption.

The parameters q, m, k are referred to as the modulus, number of samples, and the dimension (or module rank) of the MLWE problem. In the concrete instantiation underlying ML-KEM, one works over the ring $R = \mathbb{Z}[X]/(X^n+1)$ with $n = 256$ and $q = 3329$, and chooses $k \in \{2, 3, 4\}$ corresponding to the three parameter sets ML-KEM-512, ML-KEM-768, and ML-KEM-1024 [Nat24b]. The noise

and secret distributions are discrete, bounded, and very narrow: ML-KEM samples coefficients from centered binomial distributions B_{η_1}, B_{η_2} with small parameters $\eta_1, \eta_2 \in \{2, 3\}$.

Short secret and entropic LWE. When the secret is sampled from a small error distribution χ then this variant is also known as the small-secret MLWE assumption. This variant, used by ML-KEM, allows for some compression techniques (e.g., more compact public keys and seeds instead of explicit matrices) and is essentially equivalent to the MLWE assumption with a uniform secret and an increased number $m' = m + k$ of samples [App+09]. More generally, a number of works show that changing the secret distribution to another sufficiently entropic or short distribution does not substantially affect hardness: this is well understood for (ring-)LWE with secrets drawn from error-like or binary distributions [Bra+13; Mic18], and has recently been extended to the module setting for binary secrets as well [Bou+21; Bou+22]. These results support the use of small or bounded secrets as in ML-KEM.

Reductions. Definition 2.2 states the *decision* variant of MLWE. The *search* variant, given $(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{x})$ sampled as in Definition 2.2, asks to recover $\mathbf{s}$. Up to differences in the number of samples m, search and decision are equivalent: this was proved for LWE by Regev [Reg05] and extended to more general settings [Pei09; App+09; MM11; MP12; Bra+13], including RLWE and MLWE[LPR10; LS15]. The main confidence in the hardness of the LWE problem comes from the existence of worst-case to average-case reductions that show that if one can solve LWE, then one can solve certain worst-case general lattice problems. Regev [Reg05] gives a *quantum* reduction from SIVP and GapSVP in dimension n with polynomial approximation to average-case LWE with polynomial modulus, while classical reductions from GapSVP to LWE use either exponential modulus [Pei09] or dimension n^2 with polynomial modulus [Bra+13]. These results extend to structured variants, relating them to the worst-case ideal/module lattice problems. Langlois–Stehlé [LS15] show that (search or decision) MLWE over a cyclotomic ring R and rank k is as hard as approximating *module*-SIVP and *module*-GapSVP on R-modules of rank k, up to polynomial factors; later work gives essentially tight converse reductions, so MLWE is asymptotically as hard as these module problems.

Concrete hardness and best known attacks. For the concrete parameters of ML-KEM, hardness is assessed by comparing against the cost of the best known lattice-based attacks, most notably the primal, dual, and hybrid variants of lattice-reduction attacks [dv25]. We provide a systematic and up-to-date presentation of these techniques in Section 5.

Error distribution. The hardness of the $\mathsf{MLWE}_{q,m,k,\chi}$ problem depends on the error distribution χ. Classically, the coefficients of χ follow continuous [Reg05] or discrete [GKV10] Gaussian distributions, which is convenient for theory. For efficiency, other distributions are used in practice (bounded small support, efficient constant-time sampling, or better bounds/analyses). Worst-case to average-case reductions extend to this regime, e.g., to uniform errors on small or even binary support [MP13; DM13], or to any distribution with sufficient min-entropy [Bra+13]. These work only under strong bounds on the number of samples, which is inherent given efficient attacks with many samples [AG11]. State-of-the-art cryptanalysis, see Section 5.3.5, similarly suggests that in the bounded-samples regime the exact distribution matters little as long as it has enough entropy and produces secret and error vectors of comparable size.

For ML-KEM, the secret and error follow centered binomial distributions with parameters η_1 and η_2, respectively. For ML-KEM-512 we have $\eta_2 < \eta_1$, so the error is smaller than the secret, but the compression function adds deterministic errors that heuristically rebalance this. This can be viewed as an LWE problem combined with a *Learning With Rounding* (LWR) problem. Heuristically, current cryptanalysis suggests that this extra rounding adds some security, but it is usually ignored in reductions; we will discuss its impact on concrete security in Section 5.1.

2.4 Cryptographic Primitives

2.4.1 Public Key Encryption

We review the syntax and minimal definition of a public key encryption (PKE) scheme. We only define IND-CPA security for PKE below since we will not require the stronger IND-CCA security for this report.

Definition 2.3 (PKE). *A public key encryption* (PKE) *scheme with message space $\mathcal{M}$ consists of the following three PPT algorithms:*

$\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$**:** *The key generation algorithm takes the security parameter as input and outputs a pair of public and secret keys* $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{Encrypt}(\mathsf{pk}, \mathsf{m}) \to \mathsf{ct}$**:** *The (possibly randomized) encryption algorithm takes a public key* pk *and message* $\mathsf{m} \in \mathcal{M}$ *as input and outputs a ciphertext* ct.

$\mathsf{Decrypt}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{m}$**:** *The (deterministic) decryption algorithm takes a secret key* sk *and a ciphertext* ct *as input and outputs a message* $\mathsf{m} \in \mathcal{M}$.

It is common in lattice-based cryptography to define correctness to hold with all but negligible probability. While it is possible to construct a scheme with perfect correctness as is standard in classical cryptography, this incurs an efficiency loss which we typically want to avoid. Moreover, as we see later in Section 3.3, computing the exact correctness failure rate in lattice-based cryptography can be challenging in practice, as it requires precise control over how "noise" accumulates.

Definition 2.4 (δ-Correctness). *We say a* PKE *is δ-correct if for all $\lambda \in \mathbb{N}$ we have*

$$\mathbb{E}_{(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda)} \left[\max_{\mathsf{m} \in \mathcal{M}} \Pr\left[\mathsf{Decrypt}(\mathsf{sk}, \mathsf{ct}) = \mathsf{m} : \mathsf{ct} \xleftarrow{\$} \mathsf{Encrypt}(\mathsf{pk}, \mathsf{m}) \right] \right] \geqslant 1 - \delta,$$

where the probability is taken over the randomness of the encryption algorithm.

The following states that even if an adversary gets to choose two messages, the ciphertext does not reveal which message it encrypts. Since the adversary is not allowed access to the decryption oracle, this is sometimes called *passive* security.

Definition 2.5 (IND-CPA Security). *Let Π be a* PKE *scheme. For any adversary $\mathcal{A}$, we define the advantage for indistinguishability under chosen-plaintext attack (*IND-CPA*) security as follows:*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi,\mathcal{A}}(1^\lambda) := \left| \Pr\left[b = b' : \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda); \\ b \xleftarrow{\$} \{0,1\}; \\ (\mathsf{m}_0, \mathsf{m}_1, \mathsf{state}) \xleftarrow{\$} \mathcal{A}(\mathsf{pk}); \\ \mathsf{ct}^* \xleftarrow{\$} \mathsf{Encrypt}(\mathsf{pk}, \mathsf{m}_b); \\ b' \xleftarrow{\$} \mathcal{A}(\mathsf{pk}, \mathsf{ct}^*, \mathsf{state}) \end{array} \right] - \frac{1}{2} \right|,$$

We say a PKE *scheme* Π *is* IND-CPA *secure if the advantage* $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi,\mathcal{A}}(1^\lambda)$ *is negligible for any efficient adversary* $\mathcal{A}$*.*

2.4.2 Key Encapsulation Mechanisms

A key encapsulation mechanism (KEM) is a simplified PKE where the message is randomly sampled. It allows a sender (i.e., the one that generates the ciphertext) and a receiver (i.e., the one holding the decapsulation key) to agree on a so-called *session key*, which is typically a random bit string.

Below, in order to differentiate from PKE, we use terms such as encapsulation and decapsulation as opposed to encryption and decryption.

Definition 2.6 (KEM). *A key encapsulation mechanism* (KEM) *scheme with key space* $\mathcal{K}$ *consists of the following three PPT algorithms:*

$\mathsf{KeyGen}(1^\lambda) \to (\mathsf{ek}, \mathsf{dk})$**:** *The key generation algorithm takes the security parameter as input and outputs a pair of keys* $(\mathsf{ek}, \mathsf{dk})$*.*

$\mathsf{Encaps}(\mathsf{ek}) \to (\mathsf{k}, \mathsf{ct})$**:** *The encapsulation algorithm takes an encapsulation key* ek *as input and outputs a session key* $\mathsf{k} \in \mathcal{K}$ *and a ciphertext* ct*.*

$\mathsf{Decaps}(\mathsf{dk}, \mathsf{ct}) \to \mathsf{k}$**:** *The (deterministic) decapsulation algorithm takes a decapsulation key* dk *and a ciphertext* ct *as input and outputs a session key* $\mathsf{k} \in \mathcal{K}$*.*

Similarly to a PKE, we consider a correctness definition where some decapsulation failure is allowed.

Definition 2.7 (δ-Correctness). *We say a* KEM *is* δ*-correct if for all* $\lambda \in \mathbb{N}$ *we have*

$$\Pr\left[\mathsf{Decaps}(\mathsf{dk}, \mathsf{ct}) = \mathsf{k} : \begin{array}{c}(\mathsf{ek}, \mathsf{dk}) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda); \\ (\mathsf{k}, \mathsf{ct}) \xleftarrow{\$} \mathsf{Encaps}(\mathsf{ek})\end{array}\right] \geqslant 1 - \delta.$$

The following is the de-facto security standard for KEMs. It is defined analogously to IND-CPA security, except that the adversary is given access to a decapsulation oracle. On input a ciphertext, the oracle runs the decapsulation algorithm and gives back the output. In order not to trivialize the game, the decapsulation oracle will not take the challenge ciphertext as input. Since the adversary is allowed access to the decapsulation oracle, this is sometimes called *active* security.

Definition 2.8 (IND-CCA Security). *Let* Π *be a* KEM *scheme. For any adversary* $\mathcal{A}$*, we define the advantage for indistinguishability under chosen-ciphertext attack (*IND-CCA*) security as follows:*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi,\mathcal{A}}(1^\lambda) := \left|\Pr\left[b = b' : \begin{array}{c}(\mathsf{ek}, \mathsf{dk}) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda); \\ b \xleftarrow{\$} \{0,1\}; \\ (\mathsf{k}_0^*, \mathsf{ct}^*) \xleftarrow{\$} \mathsf{Encaps}(\mathsf{ek}); \\ \mathsf{k}_1^* \xleftarrow{\$} \mathcal{K}; \\ b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{Decaps}}(\cdot)}(\mathsf{ek}, \mathsf{k}_b^*, \mathsf{ct}^*)\end{array}\right] - \frac{1}{2}\right|,$$

where $\mathcal{O}_{\mathsf{Decaps}}(\cdot)$ *is an oracle that takes as input a ciphertext* ct *and outputs* $\mathsf{Decaps}(\mathsf{dk}, \mathsf{ct})$ *if and only if* $\mathsf{ct} \neq \mathsf{ct}^*$*. We say a* KEM *scheme* Π *is* IND-CCA *secure if the advantage* $\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi,\mathcal{A}}(1^\lambda)$ *is negligible for any efficient adversary* $\mathcal{A}$*.*

In the above, if security only holds against an adversary $\mathcal{A}$ that is not given access to the decapsulation oracle $\mathcal{O}_{\mathsf{Decaps}}$, it is called indistinguishability under chosen-plaintext attack (IND-CPA) secure.

2.4.3 Pseudorandom Functions

Lastly, we recall the definition of a pseudorandom function.

Definition 2.9 (Pseudorandom Function). *Let* $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ *be an efficiently computable function. For any adversary* $\mathcal{A}$*, we define the advantage for the* pseudorandomness *of the* PRF *as follows:*

$$\mathsf{Adv}^{\mathsf{rand}}_{\mathsf{PRF},\mathcal{A}}(1^\lambda) := \left|\Pr\left[\mathcal{A}^{\mathsf{RF}}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{PRF}(\mathsf{k},\cdot)}(1^\lambda) = 1\right]\right|,$$

where RF *is a random function sampled uniformly from the set of all functions from* $\mathcal{X}$ *to* $\mathcal{Y}$*, and* $\mathsf{k} \xleftarrow{\$} \mathcal{K}$*. We say a* PRF *is pseudorandom if the advantage* $\mathsf{Adv}^{\mathsf{rand}}_{\mathsf{PRF},\mathcal{A}}(1^\lambda)$ *is negligible for any efficient adversary* $\mathcal{A}$*.*

2.5 The Random Oracle Model

The security of many practical cryptographic primitives and protocols is analyzed in the *random oracle model* (ROM) [BR93]. This is an idealized framework used in cryptography to analyze the security of primitives and protocols, particularly those that rely on *cryptographic hash functions* (e.g., SHA-3). In this model, the hash function is replaced with a perfect random function that all entities, including adversaries, can only access as a black box through oracle queries. Concretely, if a hash function $\mathsf{H} : \{0,1\}^n \to \{0,1\}^m$ was used for a construction, this H will be replaced in the security proof by a function that is sampled uniformly at random from the set of all possible functions from $\{0,1\}^n$ to $\{0,1\}^m$. This abstracts our common belief that in practice, concrete cryptographic hash functions output random values, even if we know their full description.

Proving a scheme is secure in the ROM provides strong heuristic evidence that it will be secure in the real world when the oracle is instantiated with a well-designed cryptographic hash function (e.g., SHA-3). This is because the model captures the desired properties of a hash function, such as its output being unpredictable (pseudorandom) and deterministic. However, a proof in the ROM is not a guarantee of security in the standard model (where no such ideal oracles exist). Indeed, there are theoretical cryptographic schemes that are provably secure in the ROM but are demonstrably insecure no matter which real hash function is used to implement them. Despite this limitation though, the ROM remains an invaluable tool for designing and gaining confidence in the security of practical cryptographic schemes such as ML-KEM.

3 Overview of ML-KEM

In this section, we provide a minimal overview of ML-KEM necessary to understand its design principles. In particular, we ignore the implementation specifics such as how ring elements are represented as byte arrays or how ring elements are multiplied using so-called number-theoretic transforms (NTTs). These details do not impact its theoretical guarantees such as correctness, provable security, and hardness of the underlying mathematical problems. Implementation details and implementation-specific security are discussed in Section 6.

3.1 Design Principle

ML-KEM is based on the lattice-based cryptosystem from [LPR10; LP11]. While [LPR10] uses the highly structured *ring* LWE (RLWE) problem, leading to for instance NEWHOPE [Alk+16], [LP11] uses the unstructured standard LWE problem, leading to for instance Frodo [Bos+16]. The design of ML-KEM sits between these, using the somewhat structured *module* LWE (MLWE) problem.

The core cryptosystem. Let us go over the design principle of ML-KEM in a bottom-up manner. At its core, the encapsulation key is simply an MLWE instance:

$$(\mathsf{ek}, \mathsf{dk}) = \Big(\big(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}\big), \quad \mathbf{s} \Big) \in \Big(R_q^{k \times k} \times R_q^k \Big) \times R_q^k.$$

In general, $\mathbf{A}$ can be a non-square matrix but we only consider a square matrix for simplicity; ML-KEM also uses a square matrix.

A ciphertext is another MLWE instance:

$$\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1) = \big(\mathbf{A}^\top \cdot \mathbf{y} + \mathbf{e}_1, \quad \mathbf{b}^\top \cdot \mathbf{y} + e_2 + \mathsf{k} \cdot \lceil q/2 \rfloor\big) \in R_q^k \times R_q.$$

Here, the noises $(\mathbf{s}, \mathbf{e}, \mathbf{y}, \mathbf{e}_1, e_2)$ must be sampled from an appropriate distribution so that the MLWE problem is hard to solve, and the random key $\mathsf{k} \xleftarrow{\$} \{0,1\}^n$ is encoded as an element in the polynomial ring R_q whose i-th coefficient is k_i. Notice that since $\mathbf{b}, \mathsf{ct}_0$, and ct_1 are all MLWE instances, they are (informally) indistinguishable from random elements over R_q, establishing that the above construction achieves IND-CPA security.

To decapsulate, we first perform the following computation:

$$\begin{aligned} \mathsf{ct}_1 - \mathbf{s}^\top \cdot \mathsf{ct}_0 &= \big(\mathbf{b}^\top \cdot \mathbf{y} + e_2 + \mathsf{k} \cdot \lceil q/2 \rfloor\big) - \mathbf{s}^\top \cdot \big(\mathbf{A}^\top \cdot \mathbf{y} + \mathbf{e}_1\big) \\ &= \mathsf{k} \cdot \lceil q/2 \rfloor + \mathbf{e}^\top \cdot \mathbf{y} - \mathbf{s}^\top \cdot \mathbf{e}_1 + e_2 \mod q. \end{aligned}$$

Assuming the noises are small enough (i.e., $\|\mathbf{e}^\top \cdot \mathbf{y} - \mathbf{s}^\top \cdot \mathbf{e}_1 + e_2\|_\infty < q/4$), we can uniquely decode k by rounding each coefficient of the ring element $w = \mathsf{ct}_1 - \mathbf{s}^\top \cdot \mathsf{ct}_0$. That is, if the i-th coefficient of w is closer to 0 in absolute value compared to $\lceil q/2 \rfloor$, we decode to $\mathsf{k}_i = 0$, and otherwise we decode to $\mathsf{k}_i = 1$. Looking ahead, this assumption on the noise being small requires some care.

Notice that in the above, we obtain a KEM based on the unstructured LWE problem by setting $n = 1$, that is, $R_q = \mathbb{Z}_q$. While this results in a scheme based on a very conservative hardness assumption, the downside is that we can only encapsulate a one-bit key $\mathsf{k} \in \{0,1\}$. In contrast,

we obtain a KEM based on the highly structured RLWE problem by setting $k = 1$ and $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. We can encapsulate an n-bit key in one shot now but will have to rely on a more aggressive hardness assumption than the standard LWE assumption. Moreover, as n is required to be a power-of-two for implementation efficiency (i.e., allowing the use of NTT-based multiplication), the parameter choice for RLWE is quite sparse and hard to tune in practice.

The MLWE problem allows us to hit the right balance between these two cryptosystems lying at the opposite sides of the spectrum by tuning the values of n and k. In the specific case of the MLWE parameters used in ML-KEM, performance turns out to be very similar to the scheme based on RLWE since only a key of fixed size of 256 bits is necessary. This is because a 256-bit key will be used as a symmetric key to encrypt larger messages via the KEM-DEM framework (cf. [CS03]). Importantly, this allows ML-KEM to fix n to be 256 for all NIST security levels (cf. Table 3).

Upgrading it to IND-CCA security. As already mentioned, the above cryptosystem only offers weak IND-CPA security. Indeed, if an adversary is given access to a decapsulation oracle, it can trivially learn the key by submitting a ciphertext ct such that $\mathsf{ct} = \mathsf{ct}^* + (0, \Delta)$ for a small non-zero noise $\Delta \in R_q$, where ct^* is the challenge ciphertext.

The Fujisaki-Okamoto (FO) transform [FO99; FO13; HHK17] is used to upgrade such a weakly secure KEM to an IND-CCA-secure one. While there are numerous variations of the FO transform, we only informally explain one variant below. We will highlight the core parts of the FO transform that differ between variations, and provide the concrete FO transform used by ML-KEM in Section 3.2.

For convenience, let us view the above IND-CPA secure KEM as a PKE scheme, and explain how the FO transform upgrades this IND-CPA secure PKE into the desired IND-CCA secure KEM. At a high level, the FO transform modifies the encryption and decryption algorithm of the IND-CPA secure PKE as follows:

Encapsulation: $\mathsf{Encaps}(\mathsf{ek})$

- Sample a random message $\mathsf{m} \xleftarrow{\$} \{0,1\}^n$ for the PKE scheme.
- Derive a key k and randomness r for the PKE.Encrypt algorithm via $(\mathsf{k}, r) = \mathsf{G}(\mathsf{m})$, where G is a hash function.

  //* *Variations** // We can derive the key and randomness in a different manner such as by $\mathsf{G}(\mathsf{m}, \mathsf{ek})$ or $\mathsf{G}(\mathsf{m}, \mathsf{H}(\mathsf{ek}))$, where H is another hash function. This provides slightly stronger security than IND-CCA security and will be discussed in Section 4. Indeed, ML-KEM performs $\mathsf{G}(\mathsf{m}, \mathsf{H}(\mathsf{ek}))$. Another variation is to first derive only the randomness $r = \mathsf{G}(\mathsf{m})$, compute the PKE ciphertext ct, and then derive the final key as $\mathsf{k} = \mathsf{G}(\mathsf{m}, \mathsf{ct})$.
- Run $\mathsf{ct} = \mathsf{PKE.Encrypt}(\mathsf{pk}, \mathsf{m}; r)$, where ek is defined as the public key pk of the PKE scheme.
- Output $(\mathsf{k}, \mathsf{ct})$.

Decapsulation: $\mathsf{Decaps}(\mathsf{dk}, \mathsf{ct})$

- Run $\mathsf{m}' = \mathsf{PKE.Decrypt}(\mathsf{sk}, \mathsf{ct})$, where dk is defined as the secret key sk of the PKE scheme.
- Derive $(\mathsf{k}, r') = \mathsf{G}(\mathsf{m}')$.
- Re-encrypt and check if $\mathsf{PKE.Encrypt}(\mathsf{pk}, \mathsf{m}'; r') = \mathsf{ct}$.

- If the check passes, output k.

  //* *Variations** // We intentionally keep it agnostic in the case the check does *not* pass. We can output a special symbol $\perp$ indicating decapsulation failure, leading to the so-called *explicit* rejection approach. In contrast, we can output a random key k so as not to indicate whether a decapsulation failure occurred, leading to the so-called *implicit* rejection approach. ML-KEM uses implicit rejection.

Regardless of which variations are used, the core idea of the FO-transform is to recover the randomness used to run the encryption algorithm of the underlying IND-CPA secure PKE, and perform a re-encryption check during decapsulation. This intuitively guarantees that if an adversary outputs a ciphertext that decapsulates to a valid key, it must have known how the ciphertext was generated. To put it differently, we can prove IND-CCA security of the KEM relying only on the IND-CPA security of the PKE since the decapsulation oracle does not give away new information about the submitted ciphertext to the adversary. Formal details about the IND-CCA security of ML-KEM are given in Sections 4.1.1 and 4.1.2.

Notable optimizations. We wrap up this section by explaining some of the notable optimizations of ML-KEM to better understand the design choice made by ML-KEM in the next section. As mentioned earlier, we will not discuss implementation-specific optimizations such as how polynomial ring elements are stored in memory to minimize the number of NTT operations.

(i) Encapsulation key compression. ML-KEM adopts the approach taken by Alkim et al. [Alk+16] and generates the public matrix $\mathbf{A} \in R_q^{k \times k}$ in the encapsulation key ek via a hash function $\mathsf{H}_{\mathbf{A}}$. Concretely, $\mathbf{A}$ is replaced by a 256-bit seed $\rho \in \{0,1\}^{256}$ such that $\mathbf{A} = \mathsf{H}_{\mathbf{A}}(\rho)$. This effectively reduces the size of the encapsulation key almost for free.

(ii) Bit-dropping. As decryption only looks at the upper bits of the ciphertext $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1)$ (i.e., the lower bits are simply "noise"), we can safely round off the lower bits to compress the size of ct. This is a common technique used in LWE-based schemes (cf. [Pei09; PG14]). In ML-KEM, this is implemented using the functions $\mathsf{Compress}_d$ and $\mathsf{Decompress}_d$ from Section 2.3.5. Moreover, the rate of compression is different for the first and second halves of the ciphertext ct_0 and ct_1, respectively. These choices are made to balance between ciphertext size, security, and decapsulation failure probability.

It is worth mentioning that one can also perform bit-dropping on the vector $\mathbf{b}$ included in the encapsulation key. This has the effect of compressing the encapsulation key size while increasing the decapsulation failure probability. However, as analyzed in [Bos+18], it is unclear how to prove such a scheme secure from the MLWE problem, and this optimization is not implemented by ML-KEM.

(iii) Binomial noise. The LWE problem, including RLWE and MLWE, typically considers Gaussian noise, either rounded Gaussian [Reg05] or discrete Gaussian [Bra+13]. This is because these are the distributions for which we have theoretical guarantees on the hardness of the LWE problem under worst-case lattice problems. However, these noise distributions turn out to be either inefficient to implement (cf. [Bos+15]) or difficult to securely implement against side-channel attacks (cf. [Bru+16; Esp+17; PBY17]).

For practical lattice-based cryptosystems, state-of-the-art lattice cryptanalysis shows that the concrete hardness of the LWE problem does not depend on the exact distribution of the noise, but rather on the standard deviation of it (see Section 5 for the details). Therefore, a more pragmatic

option is to assume the hardness of the LWE problem with a noise distribution that can be easily, efficiently, and securely sampled. ML-KEM chooses the centered binomial distribution (cf. Section 2.3.6) used in [Alk+16].

Recall that ML-KEM relies on several MLWE instances, $\mathbf{b}, \mathsf{ct}_0$, and ct_1, where different rates of bit-dropping are performed on ct_0 and ct_1. As bit-dropping adds implicit noise created via the function $\mathsf{Compress}_d$, it can be interpreted as increasing the noise of the MLWE instances ct_0 and ct_1. ML-KEM utilizes this observation exclusively for the scheme with NIST level 1 security (i.e., ML-KEM-512) and samples the noise e_2 in ct_1 from a slightly smaller centered binomial distribution.

(iv) Decapsulation failure. Lastly, ML-KEM allows for a negligible decapsulation failure probability. While a parameter with a zero chance of decapsulation failure (i.e., perfect correctness) is appealing from a theoretical standpoint and mitigates specific attacks exploiting decapsulation failures, it will have a negative effect on either the concrete hardness of the MLWE problem (by significantly decreasing the noise) or the performance (by increasing the lattice dimension k to compensate for the loss in security).

In ML-KEM, the parameters are set such that the decapsulation failure probabilities are negligibly small; $2^{-138.8}, 2^{-164.8}$, and $2^{-174.8}$ for the NIST levels 1, 3, and 5 parameters. While there are attacks that can exploit these decapsulation failures, these are typically a much smaller threat than directly attacking ML-KEM. See Section 5.6 for more details.

3.2 Description of ML-KEM

Following the FIPS-203 publication [Nat24b], we first describe an IND-CPA secure PKE scheme called K-PKE (see Section 3.2.1, Figure 2), and then describe the IND-CCA secure ML-KEM obtained by performing the FO-transform on K-PKE (see Section 3.2.2, Figure 3). For reference, we provide an overview in Figure 1.

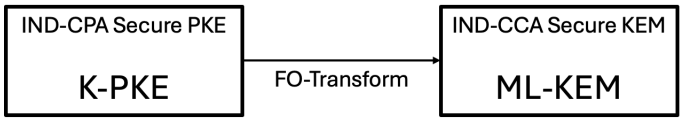


Figure 1: Constructing ML-KEM from K-PKE.

ML-KEM comes equipped with three parameter sets as given in Table 2. As explained in the previous section, the length of the shared key k is set to 32 bytes, or equivalently 256 bits, for all security levels. This means that the shared secret can be used directly in AES-256, though note that the *security* level will match the ML-KEM parameter set used and not the security level provided by AES-256.

For each of the security levels, the concrete parameters used by ML-KEM (and the underlying K-PKE) are listed in Table 3. We provide a short rationale for how each parameter was chosen:

- n is chosen to be 256 across all security levels. This relates to the fact that we only need to encapsulate a shared key of size 256 bits. Smaller values of n will require encoding more bits into one polynomial coefficient, which requires lowering noise levels for correctness, and therefore lower security. Larger values of n make it hard to fine tune the other parameters as we require n to be a power-of-two for efficient implementation using NTT-based multiplication.

Table 2: Overview of the sizes (in bytes) of keys and ciphertexts of ML-KEM and its decapsulation failure rate. NIST levels 1, 3, and 5 security correspond to ML-KEM-512, ML-KEM-768, and ML-KEM-1024, respectively.

	encapsulation key (ek)	decapsulation key (dk)	ciphertext (ct)	shared key (k)	decapsulation failure rate δ
NIST-1	800	1632	768	32	$2^{-138.8}$
NIST-3	1184	2400	1088	32	$2^{-164.8}$
NIST-5	1568	3168	1568	32	$2^{-174.8}$

Table 3: Overview of the parameters used by ML-KEM (and K-PKE). " - " indicates that it takes the same value as the value in its left cell. NIST levels 1, 3, and 5 security correspond to ML-KEM-512, ML-KEM-768, and ML-KEM-1024, respectively.

Notations	Explanation	Concrete Parameters		
		NIST-1	NIST-3	NIST-5
q	Modulus size	3329	-	-
R_q	Polynomial ring $\mathbb{Z}[X]/(X^n+1)$ of degree n	256	-	-
k	Module rank	2	3	4
d_u	Amount of bit-dropping on the first half of ciphertext	10	-	11
d_v	Amount of bit-dropping on the second half of ciphertext	4	-	5
B_{η_1}	Centered binomial distribution with parameter η_1	3	2	-
B_{η_2}	Centered binomial distribution with parameter η_2	2	-	-

- q is chosen to be a small prime satisfying $n \mid (q-1)$. This is required to enable fast NTT-based multiplication. While $q = 257$ and 769 satisfy this condition, the next smallest prime 3329 was chosen since the former two primes are too small to achieve negligible decapsulation failure probability.
- k is chosen to fix the lattice dimension as a multiple of n. This defines the hardness of the underlying MLWE problem. Put differently, changing k is the main mechanism allowing us to tune ML-KEM to different security levels.
- The remaining parameters $(d_u, d_v, B_{\eta_1}, B_{\eta_2})$ were chosen to balance between security, ciphertext size, and decapsulation failure probability.

3.2.1 Algorithms for K-PKE

K-PKE is the IND-CPA secure PKE used by ML-KEM. As explained in Section 3.1, K-PKE is similar to the PKE scheme that was introduced in [LPR10; LP11]. The pseudocode for K-PKE is given in Figure 2. The internal functions used by K-PKE are listed in Table 4.

3.2.2 Algorithms for ML-KEM

The pseudocode for ML-KEM is given in Figure 3, where it runs K-PKE internally and performs the FO transform. The internal functions used by K-PKE are listed in Table 4. As we explained

Table 4: Overview of the functions used by ML-KEM and K-PKE. The functions above the dashed line are used internally in ML-KEM, while those below are used internally in K-PKE.

Functions	Explanation
G	A function $\mathsf{G} : \{0,1\}^* \to \{0,1\}^{256} \times \{0,1\}^{256}$
J	A function $\mathsf{J} : \{0,1\}^* \to \{0,1\}^{256}$ for implicit rejection
H	A function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{256}$ to compress encapsulation key ek
$\mathsf{H}_{\mathbf{A}}$	A function $\mathsf{H}_{\mathbf{A}} : \{0,1\}^* \to R_q^{k \times k}$ to generate $\mathbf{A}$
$\mathsf{H}_{\mathbf{s},\mathbf{e}}$	A function $\mathsf{H}_{\mathbf{s},\mathbf{e}} : \{0,1\}^* \to R_q^k \times R_q^k$ to generate MLWE secrets $(\mathbf{s},\mathbf{e})$ from B_{η_1}
$\mathsf{H}_{\mathbf{y},\mathbf{e}_1,e_2}$	A function $\mathsf{H}_{\mathbf{y},\mathbf{e}_1,e_2} : \{0,1\}^* \to R_q^k \times R_q^k \times R_q$ to generate MLWE secrets $\mathbf{y}$ from B_{η_1} and noises $(\mathbf{e}_1, e_2)$ from B_{η_2}

$\mathsf{K\text{-}PKE.KeyGen}(; d)$

Input: randomness $d \in \{0,1\}^{256}$
Output: encryption key $\mathsf{ek}_{\mathsf{PKE}} \in R_q^k \times \{0,1\}^{256}$
Output: decryption key $\mathsf{dk}_{\mathsf{PKE}} \in R_q^k$

$(\rho, \sigma) := \mathsf{G}(d)$
$\mathbf{A} := \mathsf{H}_{\mathbf{A}}(\rho)$ // $\mathbf{A} \in R_q^{k \times k}$
$(\mathbf{s}, \mathbf{e}) := \mathsf{H}_{\mathbf{s},\mathbf{e}}(\sigma)$ // Sample $\mathbf{s}, \mathbf{e} \in R_q^k$ from $B_{\eta_1}^k$
$\mathbf{t} := \mathbf{A} \cdot \mathbf{s} + \mathbf{e}$
$\mathsf{ek}_{\mathsf{PKE}} := (\mathbf{t} \| \rho)$ // append $\mathbf{A}$ seed
$\mathsf{dk}_{\mathsf{PKE}} := \mathbf{s}$
return $(\mathsf{ek}_{\mathsf{PKE}}, \mathsf{dk}_{\mathsf{PKE}})$

$\mathsf{K\text{-}PKE.Encrypt}(\mathsf{ek}_{\mathsf{PKE}}, m; r)$

Input: encryption key $\mathsf{ek}_{\mathsf{PKE}} \in R_q^k \times \{0,1\}^{256}$
Input: message $m \in \{0,1\}^{256} \subset R_q$
Input: randomness $r \in \{0,1\}^{256}$
Output: ciphertext $\mathsf{ct} \in R_{2^{d_u}}^k \times R_{2^{d_v}}$

parse $(\mathbf{t} \| \rho) \leftarrow \mathsf{ek}_{\mathsf{PKE}}$
$\mathbf{A} := \mathsf{H}_{\mathbf{A}}(\rho)$ // regenerate $\mathbf{A} \in R_q^{k \times k}$
$\mathbf{y} := \mathsf{H}_{\mathbf{y},\mathbf{e}_1,e_2}(r)$ // Sample $\mathbf{y}, \mathbf{e}_1 \in R_q^k$ from $B_{\eta_1}^k$ // and $B_{\eta_2}^k$, respectively, and $e_2 \in R_q$ from B_{η_2}
$\mathbf{u} := \mathbf{A}^\top \cdot \mathbf{y} + \mathbf{e}_1$
$\mu := \mathsf{Decompress}_1(m)$
$v := \mathbf{t}^\top \cdot \mathbf{y} + e_2 + \mu$
$\mathsf{ct}_1 := \mathsf{Compress}_{d_u}(\mathbf{u})$
$\mathsf{ct}_2 := \mathsf{Compress}_{d_v}(v)$
return $\mathsf{ct} := (\mathsf{ct}_1 \| \mathsf{ct}_2)$

$\mathsf{K\text{-}PKE.Decrypt}(\mathsf{dk}_{\mathsf{PKE}}, \mathsf{ct})$

Input: decryption key $\mathsf{dk}_{\mathsf{PKE}} \in R_q^k$
Input: ciphertext $\mathsf{ct} \in R_{2^{d_u}}^k \times R_{2^{d_v}}$
Output: message $m \in \{0,1\}^{256} \subset R_q$

parse $(\mathsf{ct}_1 \| \mathsf{ct}_2) \leftarrow \mathsf{ct}$
$\mathbf{u}' := \mathsf{Decompress}_{d_u}(\mathsf{ct}_1)$
$v' := \mathsf{Decompress}_{d_v}(\mathsf{ct}_2)$
parse $\mathbf{s} \leftarrow \mathsf{dk}_{\mathsf{PKE}}$
$w := v' - \mathbf{s}^\top \cdot \mathbf{u}'$
$m := \mathsf{Compress}_1(w)$
return m

Figure 2: Simplified algorithms for the IND-CPA secure K-PKE: K-PKE.KeyGen, K-PKE.Encrypt, and K-PKE.Decrypt. Above, we implicitly assume $\mathsf{m} \in \{0,1\}^{256}$ is encoded as a polynomial over R_q. See Figure 8 for the concrete specification.

ML-KEM.KeyGen()

Output: encapsulation key $\mathsf{ek} \in R_q^k$

Output: decapsulation key $\mathsf{dk} \in (R_q^k)^2 \times (\{0,1\}^{256})^2$

1 : $d \xleftarrow{\$} \{0,1\}^{256}$

2 : $z \xleftarrow{\$} \{0,1\}^{256}$

3 : $(\mathsf{ek}_{\mathsf{PKE}}, \mathsf{dk}_{\mathsf{PKE}}) \xleftarrow{\$} \mathsf{K\text{-}PKE.KeyGen}(\ ; d)$

// run key generation for K-PKE using randomness d

4 : $\mathsf{ek} := \mathsf{ek}_{\mathsf{PKE}}$

5 : $\mathsf{dk} := (\mathsf{dk}_{\mathsf{PKE}} \| \mathsf{ek} \| \mathsf{H}(\mathsf{ek}) \| z)$

6 : **return** $(\mathsf{ek}, \mathsf{dk})$

ML-KEM.Encaps(ek)

Input: encapsulation key $\mathsf{ek} \in R_q^k$

Output: shared secret key $\mathsf{k} \in \{0,1\}^{256}$

Output: ciphertext $\mathsf{ct} \in R_{2^{d_u}}^k \times R_{2^{d_v}}$

1 : $m \xleftarrow{\$} \{0,1\}^{256}$

2 : $(\mathsf{k}, r) := \mathsf{G}(m \| \mathsf{H}(\mathsf{ek}))$

3 : $\mathsf{ct} := \mathsf{K\text{-}PKE.Encrypt}(\mathsf{ek}, m; r)$

// encrypt m using K-PKE with randomness r

4 : **return** $(\mathsf{k}, \mathsf{ct})$

ML-KEM.Decaps(dk, ct)

Input: decapsulation key $\mathsf{dk} \in (R_q^k)^2 \times (\{0,1\}^{256})^2$

Input: ciphertext $\mathsf{ct} \in R_{2^{d_u}}^k \times R_{2^{d_v}}$

Output: shared secret key $\mathsf{k} \in \{0,1\}^{256}$

1 : **parse** $(\mathsf{dk}_{\mathsf{PKE}} \| \mathsf{ek} \| h \| z) \leftarrow \mathsf{dk}$

2 : $m' := \mathsf{K\text{-}PKE.Decrypt}(\mathsf{dk}_{\mathsf{PKE}}, \mathsf{ct})$

3 : $(\mathsf{k}', r') := \mathsf{G}(m' \| h)$

4 : $\bar{\mathsf{k}} := \mathsf{J}(z \| \mathsf{ct})$

5 : $\mathsf{ct}' := \mathsf{K\text{-}PKE.Encrypt}(\mathsf{ek}_{\mathsf{PKE}}, m'; r')$ // re-encrypt m' using derived randomness r'

6 : **if** $(\mathsf{ct} \neq \mathsf{ct}')$ **then**

7 : $\quad \mathsf{k}' \leftarrow \bar{\mathsf{k}}$

8 : **return** k'

Figure 3: Simplified algorithms for ML-KEM: ML-KEM.KeyGen, ML-KEM.Encaps, and ML-KEM.Decaps running K-PKE in Figure 2 as a sub-routine. See Figure 9 for the concrete specification.

in Section 3.1, there are small variations of the FO transform. Notice that ML-KEM.Encaps first hashes the encapsulation key ek by H, and then generates the session key and encryption randomness as $\mathsf{G}(m \| \mathsf{H}(\mathsf{ek}))$. Hashing the ek can be useful in situations where the message is not yet known as it can be precomputed; if not for the hash, then $\mathsf{G}(m \| \mathsf{ek})$ can only be computed once m is defined. Moreover, notice that $\mathsf{H}(\mathsf{ek})$ is included as part of the decapsulation key dk. This allows for a faster decapsulation process since the decryptor does not need to compute $\mathsf{H}(\mathsf{ek})$.

Another notable design choice is that ML-KEM performs *implicit* rejection. This is done by including a random seed $z \in \{0,1\}^{256}$ in the decapsulation key. In algorithm ML-KEM.Decaps, if the re-encryption check fails (cf. line 6), it outputs a "fake" session key $\bar{\mathsf{k}} := \mathsf{J}(z \| \mathsf{ct}) \in \{0,1\}^{256}$ as

opposed to a special symbol $\perp$ explicitly indicating a decapsulation failure. As z is hidden to the outside world, $\bar{\mathsf{k}}$ looks random assuming J is a pseudorandom function.

Remark 3.1 (Implicit vs Explicit Rejection). Alternatively to above, one can consider a variant of ML-KEM that performs *explicit* rejection which outputs a special symbol $\perp$ explicitly indicating a decapsulation failure. It is clear that if a KEM that performs explicit rejection is secure, it is also secure if it performs implicit rejection (cf. [Bin+19]). This is intuitive by noticing that we can replace the explicit rejection symbol $\perp$ by any value without deteriorating its security. On the other hand, while there are some implications, e.g., [HK25], we do not know whether the other direction holds unconditionally. Even to this date, there are only a handful of research on KEMs with explicit rejections [JZM19a; Don+22; HHM22; HM24; HK25]. This is one of the historical reasons why researchers focused on ML-KEM with implicit rejection; this was easier to formally prove the security compared to its explicitly rejecting counterpart. Moreover, from an implementation perspective, one can argue that implicit rejection is a better design as it "hides" whether a decapsulation failure occurred — this gives less attack surface to the adversary. However, we note that this argument is limited when using KEMs as a building block (e.g., key exchange). This is because if the protocol using the KEM prematurely terminates, this indirectly indicates that the KEM did not decapsulate properly.

3.3 Correctness

The correctness of ML-KEM has been theoretically and analytically analyzed in [Bos+18]. Theoretically, the decryption failure rate δ of ML-KEM is expressed as follows. Below, [Bos+18, Theorem 1] provides the decryption failure rate δ of the underlying K-PKE, and [HHK17] shows that the same δ is inherited to ML-KEM in the (quantum) random oracle model.

Theorem 3.2 ([Bos+18, Theorem 1] and [HHK17]). *Let $\mathbf{s}, \mathbf{e}, \mathbf{y} \in R_q^k$ be sampled from $B_{\eta_1}^k$, $\mathbf{e}_1 \in R_q^k$ from $B_{\eta_2}^k$, and $e_2 \in R_q$ from B_{η_2}. Also, let $\mathbf{c}_u \xleftarrow{\$} F_{d_u}^k$ and $c_v \xleftarrow{\$} F_{d_v}$ be distributed according to the distribution F_d defined as follows:*

Let F_d be the following distribution:

- *Choose uniformly random $y \xleftarrow{\$} R$.*
- *Output $(y - \mathsf{Decompress}_d(\mathsf{Compress}_d(y))) \bmod^{\pm} q$.*

Denote the decryption failure rate δ as

$$\delta := \Pr\left[\|\mathbf{e}^\top \cdot \mathbf{y} + e_2 + c_v - \mathbf{s}^\top \cdot \mathbf{e}_1 - \mathbf{s}^\top \cdot \mathbf{c}_u\|_\infty \geqslant \lceil q/4 \rfloor\right].$$

Then ML-KEM is δ-correct.

Notice that the above provides only an asymptotic guarantee on the decryption failure rate δ, and it is non-trivial to exactly compute δ in a closed form. As such, [Bos+18] also provides a Python script (`Kyber.py`) to compute a tight upper bound on δ, available online at https://github.com/pq-crystals/security-estimates. The parameters in Table 2 correspond to those upper bounds.

Remark 3.3 (Heuristic Arguments). It is worth highlighting that the formal proof of Theorem 3.2 relies on a heuristic argument. More concretely, in the proof of [Bos+18, Theorem 1], they assume $\mathbf{u}$ and v generated in K-PKE.Encrypt are distributed uniformly random — this is reflected in the

distribution F_d in Theorem 3.2. While this is informally argued by relying on the hardness of the MLWE assumption, there is no formal reduction for this claim. This issue has been explicitly pointed out in [Alm+24; Kre24; Bar+25]. While we can formally prove ML-KEM to be correct assuming MLWE as shown in [Bar+25], this results in a significantly worse bound than the one computed heuristically. For instance, for ML-KEM with NIST security level 3 (i.e., ML-KEM-768), the heuristic decapsulation failure rate is $2^{-164.8}$, while the provable rate is 2^{-80} (see [Bar+25, Table 1] for more details). Formally proving the heuristic decapsulation failure rate for ML-KEM is considered an open problem.

4 Provable Security of ML-KEM

In this section, we review the theoretical results establishing IND-CCA security of ML-KEM, and survey results covering more advanced form of security such as anonymity and KEM binding properties.

4.1 IND-CCA Security

There have been more than a dozen of papers on the IND-CCA security of ML-KEM (or variants thereof). One of the main reasons for so many papers on this topic is due to the complication caused when proving IND-CCA security of ML-KEM in the *quantum* random oracle model (QROM) [Bon+11] — indeed, if we only consider *classical* results, then there are only a few papers we need to cover. QROM is a type of random oracle model (cf. Section 2.5) where the adversary is able to perform quantum queries to the random oracle. This corresponds to the real-world setting where a quantum adversary can run the cryptographic hash function such as SHA-3 in superposition. While many natural properties in the classical ROM are expected to hold in the QROM, there are contrived examples proving otherwise, e.g., [Bon+11; YZ21]. Considering that the main motivation of ML-KEM was to prepare a KEM secure against quantum adversaries, the (theoretical side of the) cryptographic community naturally tended towards proving the more ambitious IND-CCA security in the QROM.

The other reason is caused by the numerous variants of the FO transform as explained in Section 3.1. While these variants can be proven secure in a similar manner in the classical ROM, this is not the case in the QROM. As such, many incomparable proof techniques have been developed over the past years to overcome the difficulty arising in QROM proofs. Adding further to this complication is the fact that ML-KEM performs a different type of FO transform compared to CRYSTALS-KYBER [Ava+21], the KEM that was submitted to the NIST PQC project. As such, while there are papers explicitly explaining the security of CRYSTALS-KYBER, they do not translate directly to ML-KEM. This has lead non-experts unsure of what the exact reference establishing IND-CCA security of ML-KEM is.

Below, we first review the classical result on IND-CCA security and then review the more nuanced post-quantum results.

4.1.1 In the Classical ROM

For any security proofs on the FO transform, be it in the classical or quantum ROM, we first need to establish the security of the base PKE scheme K-PKE. The following theorem is a refinement of [Ava+21, Theorem 1] which holds both in the classical and quantum ROM. We note that we can rely on the result from [Ava+21] since while the FO transform used by ML-KEM and CRYSTALS-KYBER differ, the underlying K-PKE is identical.

Theorem 4.1 (IND-CPA Security of K-PKE, [Ava+21, Theorem 1]). *Assume* $\mathsf{H}_{\mathbf{A}}$ *is modeled as a random oracle and* $\mathsf{H}_{\mathbf{s},\mathbf{e}}$ *and* $\mathsf{H}_{\mathbf{y},\mathbf{e}_1,e_2}$ *are pseudorandom functions. Then, for any adversary* $\mathcal{A}$ *against the* IND-CPA *security of* K-PKE *as in Figure 2, there exist adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *against the* $\mathsf{MLWE}_{q,k,k,B_{\eta_1}}$ *and* $\mathsf{MLWE}_{q,k+1,k,B_{\eta_1},B_{\eta_2}}$ *problems, respectively, and an adversary* $\mathcal{C}$ *against the pseudorandomness of* $\mathsf{H}_{\mathbf{s},\mathbf{e}}$ *and* $\mathsf{H}_{\mathbf{y},\mathbf{e}_1,e_2}$ *such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{K\text{-}PKE},\mathcal{A}}(1^\lambda) \leqslant \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_1}(1^\lambda) + \mathsf{Adv}^{\mathsf{MLWE}}_{\mathcal{B}_2}(1^\lambda) + \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{C}}(1^\lambda).$$

Above, the first $\mathsf{MLWE}_{q,k,k,B_{\eta_1}}$ (resp. second $\mathsf{MLWE}_{q,k+1,k,B_{\eta_1},B_{\eta_2}}$) assumption in the theorem statement guarantees that the public key (resp. ciphertext) is pseudorandom. Moreover, $\mathsf{H}_{\mathbf{A}}$ is

required to be modeled as a random oracle since the reduction needs to program $\mathsf{H}_{\mathbf{A}}$ to output the MLWE challenge matrix $\mathbf{A} \in R_q^{k \times k}$ on input the public seed ρ; this can be done efficiently both in the classical and quantum ROM.

The next step is identifying the specific type of FO-transform used by ML-KEM. Once this is done, we can invoke the correct theorem establishing the IND-CCA security of the given FO-transformed KEM. As discussed in [MX23], while ML-KEM relies on a slightly different description, it is syntactically equivalent to the standard $\mathsf{FO}_m^{\not\perp}$-transform in [HHK17]. More concretely, we have the following:

- The FO-transform used by ML-KEM uses a single hash function G to compute both the session key k and the encryption randomness r. In contrast, the $\mathsf{FO}_m^{\not\perp}$-transform in [HHK17] uses two separate hash functions. However, these two computations are equivalent when the corresponding hash functions are modeled as independent random oracles with appropriate outputs lengths.
- Another difference is that the FO-transform used by ML-KEM uses the hash $\mathsf{H}(\mathsf{ek})$ to compute (k, r). In contrast, the $\mathsf{FO}_m^{\not\perp}$-transform ignores this input. Since the IND-CCA security notion only assumes a single-user setting (rather than a multi-user setting where the game considers many encapsulation keys ek), this difference can be trivially incorporated into the proof of the $\mathsf{FO}_m^{\not\perp}$-transform KEM without any notable changes.

To summarize, we can rely on the results in [HHK17; Hö20] regarding the $\mathsf{FO}_m^{\not\perp}$-transform to establish the IND-CCA security of ML-KEM in the classical ROM. Formally, we have the following theorem. Here, we note [Hö20] is a refined version of [HHK17].

Theorem 4.2 (IND-CCA **Security of ML-KEM in Classical ROM, [HHK17; Hö20]).** *Assume* K-PKE *is* δ*-correct and* IND-CPA *secure. Then, for any adversary* $\mathcal{A}$ *against the* IND-CCA *security of* ML-KEM *that makes at most* q_{RO} *classical queries to the random oracle, there exists an adversary* $\mathcal{B}$ *against the* IND-CPA *security of* K-PKE *such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{ML\text{-}KEM},\mathcal{A}}(1^\lambda) \leqslant 3 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{K\text{-}PKE},\mathcal{B}}(1^\lambda) + (q_{\mathsf{RO}} + 1) \cdot \delta + \frac{q_{\mathsf{RO}}}{2^{256}}.$$

Notice the result is *tight* in the sense that the IND-CCA security of ML-KEM is tightly bounded by the IND-CPA security of K-PKE. I.e., if there exists an adversary that breaks the IND-CCA security of ML-KEM with advantage ϵ, then there exists another adversary that breaks the IND-CPA security of K-PKE with a similar advantage. As we see in the next section, this is no longer the case in QROM.

Further notice that the decryption failure rate δ has an explicit implication in the IND-CCA security guarantee. Put differently, while $\delta \approx 2^{64}$ may be an acceptable decryption failure rate from a usability standpoint, the above theorem would not establish IND-CCA security anymore since the number of random oracle queries q_{RO} is typically expected to be larger than 2^{64}. That is, we arrive at a trivial upper bound $\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{ML\text{-}KEM},\mathcal{A}}(1^\lambda) \leqslant 1$. In fact, this is not an artifact of the security proof, but as we will see in Section 5.6, there is a concrete attack exploiting such high decryption failure rate.

4.1.2 In the Quantum ROM

As mentioned above, there have been many numerous works on the IND-CCA security of the FO-transform in the QROM, e.g., [TU16; SXY18; Jia+18; Höv+20; Zha19; JZM19a; JZM19b; Bin+19; Kuc+20; Don+22; HHM22; Che+23; HM24; GLX24; HK25]. We do not plan to go over the details of these works, and refer to survey papers such as [ZLJ25] for a general overview on the topic.

At a high level, the reasons why there are so many different papers on this topic can be summarized as follows:

(i) Tightness of the security reduction. As we saw in Theorem 4.2 in the classical ROM, we know that ML-KEM is as secure as K-PKE with almost no reduction loss. Since K-PKE is also as secure as MLWE (cf. Theorem 4.1), this means we can simply focus on how hard the MLWE problem is when assessing the hardness of ML-KEM.

While this is something we would like to have in the quantum setting, we currently do not know how to do this. For instance, in some known proofs, the upper bound for $\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{ML\text{-}KEM},\mathcal{A}}(1^\lambda)$ may look something like $\sqrt{q_{\mathsf{RO}} \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{K\text{-}PKE},\mathcal{B}}(1^\lambda)} + q_{\mathsf{RO}} \cdot \sqrt{\delta}$, e.g., [JZM19b]. More concretely, even if there exists an adversary that breaks the IND-CCA security of ML-KEM with probability ϵ, we can only assume an adversary that breaks the IND-CPA security of K-PKE (and hence break the MLWE problem) with probability $\epsilon^2/q_{\mathsf{RO}}$. For a small (yet non-negligible) ϵ and large q_{RO}, this is far less optimal compared to the guarantees we get from the classical proofs, and there have been numerous papers trying to bring the tightness loss of the quantum proof as close to the classical proof.

That said, we would like to emphasize that this does not imply that ML-KEM is less secure when considering quantum adversaries; these loose bounds may simply by an artifact of our proof methodology, and the parameters of ML-KEM are set under the common belief that similar bounds in Theorem 4.2 hold in the quantum setting.

(ii) Additional assumptions on K-PKE. In the classical setting, the only thing we assumed from K-PKE was that it is correct and IND-CPA secure. In the quantum setting, we may assume a bit more properties from K-PKE to make the security proof tighter, e.g., spreadness [FO99; FO13; HHK17], disjoint simulatable [SXY18], injective [Bin+19; Kuc+20]. While some are known to be easily implied from the design of K-PKE, some are far less obvious, requiring numerical analysis [Din+22].

(iii) Implicit or explicit rejection. As we already mentioned in Remark 3.1, in the classical setting, whether ML-KEM performs implicit or explicit rejection has almost no consequence to the security proof. However, in the quantum setting, due to our limited proof technique in the QROM, this makes a notable impact. Indeed, most of the proofs in the QROM critically uses the fact that the KEM performs implicit rejection (as in ML-KEM), and only a few work with explicit rejection exists [JZM19a; Don+22; HHM22; HM24; HK25].

In summary, it is not always clear what the *best* quantum proof of ML-KEM is since there are several tradeoffs. Below, we present two representative and incomparable security proofs of ML-KEM in the QROM. We note that the following results hold generally for any KEM derived from standard $\mathsf{FO}^{\not\perp}_m$-transform in [HHK17]; as explained in the previous section, the FO-transform performed by ML-KEM can be thought of as the $\mathsf{FO}^{\not\perp}_m$-transform.

Theorem 4.3 (IND-CCA **Security of ML-KEM in QROM, [GLX24, Corollary 1]).** *Assume* K-PKE *is δ-correct and* IND-CPA *secure. Then, for any adversary $\mathcal{A}$ against the* IND-CCA *security of* ML-KEM *that makes at most q_{RO} quantum queries to the random oracle, there exists an adversary $\mathcal{B}$ against the* IND-CPA *security of* K-PKE *and an adversary $\mathcal{C}$ against the pseudorandomness of* J *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{ML\text{-}KEM},\mathcal{A}}(1^\lambda) \leqslant \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{C}}(1^\lambda) + 2\cdot\sqrt{q_{\mathsf{RO}}}\cdot(8\cdot q_{\mathsf{RO}}+3)\cdot\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{K\text{-}PKE},\mathcal{B}}(1^\lambda)\\ + 16\cdot(4\cdot q_{\mathsf{RO}}+1)\cdot\delta + 4\cdot\sqrt{q_{\mathsf{RO}}}\cdot\delta + 16\cdot\sqrt{q_{\mathsf{RO}}}\cdot(8\cdot q_{\mathsf{RO}}+3)\cdot\frac{10\cdot q_{\mathsf{RO}}+1}{2^{256}}.\end{aligned}$$

Theorem 4.4 (IND-CCA **Security of ML-KEM in QROM, [JZM19b, Theorem 1]).** *Assume* K-PKE *is δ-correct and* IND-CPA *secure. Then, for any adversary $\mathcal{A}$ against the* IND-CCA *security of* ML-KEM *that makes at most q_{RO} quantum queries to the random oracle, there exists an adversary $\mathcal{B}$ against the* IND-CPA *security of* K-PKE *and an adversary $\mathcal{C}$ against the pseudorandomness of* J *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{ML\text{-}KEM},\mathcal{A}}(1^\lambda) \leqslant \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{C}}(1^\lambda) + 4\cdot q_{\mathsf{RO}}\cdot\sqrt{\delta}\\ + 2\cdot\sqrt{(2\cdot q_{\mathsf{RO}}+1)\cdot\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{K\text{-}PKE},\mathcal{B}}(1^\lambda) + 2\cdot\frac{(2\cdot q_{\mathsf{RO}}+1)^2}{2^{256}}}.\end{aligned}$$

It is clear that both theorems establish the IND-CCA security of ML-KEM relying on the δ-correctness and IND-CPA security of K-PKE. Here, we note that unlike in the classical setting, there is a bound concerning the pseudorandomness of J. The only reason why this does not show up in the classical setting is because we can simply assume J is implemented by a (classical) random oracle — while we can similarly do this in the quantum setting, it is much easier to assume J is a PRF.

While both theorems provide the same asymptotic guarantee, the qualitative meaning differs. Denoting $\epsilon_{\mathsf{cca}} = \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{ML\text{-}KEM},\mathcal{A}}(1^\lambda)$ and $\epsilon_{\mathsf{cpa}} = \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{K\text{-}PKE},\mathcal{B}}(1^\lambda)$, and ignoring all other components, we get $\epsilon_{\mathsf{cpa}} \gtrsim \epsilon_{\mathsf{cca}}\cdot q_{\mathsf{RO}}^{-1.5}$ and $\epsilon_{\mathsf{cpa}} \gtrsim \epsilon_{\mathsf{cca}}^2\cdot q_{\mathsf{RO}}^{-1}$ from the first and second theorems, respectively. If $\epsilon_{\mathsf{cca}} \approx 1$, i.e., an adversary breaks the IND-CCA security of ML-KEM with very high advantage, then the second theorem gives us a better proof as it translates to an adversary that breaks the MLWE problem with higher advantage. In contrast, if $\epsilon_{\mathsf{cca}} \approx q_{\mathsf{RO}}^{-1}$, then the first theorem gives us a better proof. We refer to the papers we cited at the beginning of this section for more details on the different types of asymptotic guarantees.

4.2 Other Security Properties

While IND-CCA security is the most important security property for any KEM, there are other notions of security that may be important in other contexts. We survey several additional security properties ML-KEM is known to satisfy.

Multi-user security. Standard security definitions (and concrete security analysis) only considers the security of KEMs with a single encapsulation key, i.e., single-user setting. In particular, it does not make any claims of the security of KEMs where an adversary can observe many encapsulation key, i.e., the multi-user setting. In the design of ML-KEM, two decisions were made to aim at improving security against attackers targeting multiple users (cf. [Ava+21, Section 4.5.3]).

- The matrix $\mathbf{A}$ is re-generated for each encapsulation key. This is in contrast to viewing $\mathbf{A}$ as a public parameter like the modulus q or degree n, used by all the users in the system. This protects against an attacker attempting to break many keys at the cost of breaking one key, by finding a vulnerability in the matrix $\mathbf{A}$.
- The FO-transform used by ML-KEM hashes the encapsulation key ek to generate the session key k and encryption randomness r. Making the randomness r dependent of the encapsulation key protects agains pre-computation attacks that attempt to break one out of many keys.

Anonymity and robustness. One common way to use a KEM is to use it as a PKE via the KEM-DEM paradigm [CS03]. For some applications of PKEs (and of KEMs for some degree), one may consider *anonymity* [Bel+01] and *robustness* [ABN10]. Roughly, the former guarantees that a ciphertext hides the receiver's information (i.e., the encapsulation key), a useful property for privacy-enhancing technologies such as anonymous credential systems [CL01] and anonymous authenticated key exchanges [Boy+09; Fuj+13; Fuj+15; SSW20]. In contrast, the later is an orthogonal property roughly stating that only the intended receiver can obtain a meaningful message from a ciphertext, a property having applications to searchable encryption [Abd+05] and auctions [Sak00].

There have been several works studying the properties needed by the underlying KEM for the PKE to satisfy anonymity and robustness [Moh10; Xag22; GMP22; MX23]. Since the FO-transform performed by ML-KEM can be thought of as the $\mathsf{FO}^{\not\perp}_m$-transform in [HHK17], the results of [Xag22; GMP22] indicate that if ML-KEM is *strong collision-free* CCA secure and K-PKE is *strongly disjoint-simulatable* and δ-correct for a negligible δ, then the resulting PKE obtained via the KEM-DEM paradigm with an appropriate symmetric key encryption scheme (i.e., DEM) is anonymous and robust. These properties are proven to hold even in the QROM; see also [Xag22, Section 3.1] and [GMP22, Section 5.4], where note that while these results focus on CRYSTALS-KYBER, the same arguments hold for ML-KEM.

KEM binding. [CDM24] systematically explores the possible *binding* properties of KEMs. The above mentioned robustness is one type of the binding properties explored. Slightly more formally, robustness in [Xag22; GMP22] guarantees that an adversary given two honestly generated encapsulation keys cannot find a ciphertext such that it decapsulates to a valid session key under the two decapsulation keys. While robustness guarantees that a ciphertext is bound to an honest KEM key, we can think of numerous other types of binding properties. For instance, say an adversary is given an honestly generated ciphertext under an encapsulation key of Alice. We may not want the adversary to be able to produce another ciphertext that decapsulates to the same session key under a decapsulation key of Bob. Such a re-encapsulation attack can occur regardless of whether the KEM is robust since the issue is that the session key is not bound to the KEM key (and ciphertext). This typically manifests as an explicit attack when using the KEM at the protocol level where two parties compute the same session key despite disagreeing on their respective partners, e.g., [KS23]. Since ML-KEM hashes the encapsulation key when deriving the session key, such re-encapsulation attacks are known to be mitigated. We refer to [CDM24; Sch24] for a more detailed discussion on all the different types of the binding properties and which properties ML-KEM is expected to satisfy.

5 Practical Cryptanalysis of ML-KEM

This section evaluates the concrete security of ML-KEM by analyzing the hardness of the underlying MLWE problem against all known classical and quantum attack families. We model the induced error distribution (including bit-dropping), map MLWE to standard lattice problems, and review primal, dual, hybrid, and structure-exploiting attacks. The official ML-KEM security estimates confirm that all standardized parameter sets (ML-KEM-512, -768, -1024) comfortably meet or exceed their targeted NIST levels 1, 3, and 5 under both classical and quantum cost models.

5.1 (Practical) Error Modeling of ML-KEM

We consider an $\mathsf{ML\text{-}KEM}_{q,m,k,\chi_1,\chi_2}$ instance $(\mathbf{A}, \mathbf{b}, \mathbf{s}, \mathbf{x})$ where $\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{x}$ with short secret $\mathbf{s} \xleftarrow{\$} \chi_1^k$ and error $\mathbf{x} \xleftarrow{\$} \chi_2^m$. In the IND-CPA security proof in Theorem 4.1 of K-PKE, underlying the final IND-CCA security proof of ML-KEM, we saw that we have to consider two parametrizations: $m = k$ samples with $\chi_1 = \chi_2 = B_{\eta_1}$, and $m = k + 1$ samples with $\chi_1 = B_{\eta_1}$ and $\chi_2 = B_{\eta_2}$ respectively. As the influence of the number of samples is small, we will conservatively assume that $m = k + 1$ for our security estimates. This leaves the difference in the error distribution χ_2 with parameter η_1 or η_2 for the keygen and encapsulation respectively.

For ML-KEM-512 we have that $\eta_2 < \eta_1$, while for the other two instantiations we have $\eta_1 = \eta_2$. For ML-KEM-512 we would thus have to consider the easier instance $\mathsf{ML\text{-}KEM}_{q,k+1,k,B_{\eta_1},B_{\eta_2}}$ coming from the encapsulation procedure. The reduction to this instance is not entirely tight, however, in particular it ignores the additional errors coming from bit-dropping. For the concrete security estimate of ML-KEM-512 we therefore consider two situations that could be seen as conservative or reasonable respectively.

Conservative model (referred to as ML-KEM-512$^{\text{no drop}}$): we ignore the additional errors coming from the bit-dropping and derive our security estimate based on the parameters above.

Reasonable model (referred to simply as ML-KEM-512): We will see that the efficiency of the best attacks depends on the norms of $\mathbf{s}$, and of the induced error $\mathbf{x}'$ after bit-dropping. The former depends on the variance of B_{η_1} which is $\eta_1/2$, while the latter depends on the variance $\eta_2/2$ of B_{η_2} plus the increase due to the bit-dropping. The bit-dropping is a deterministic procedure, but due to the randomness of $\mathbf{A}, \mathbf{s}, \mathbf{x}$ it is still reasonable to consider the impact on the error as a random procedure with some variance per coordinate. In fact, only applying this deterministic error is known as the *Learning With Rounding assumption*. Taking account of this variance, the variance of the norms of the coordinates $\mathbf{x}'$ is in fact always significantly larger than $\eta_1/2$. The reasonable assumption therefore, which is also followed by the authors in the Kyber proposal, is to simply assume that $\chi_2 = \chi_1$.

Note that for ML-KEM-768 and ML-KEM-1024 this was already the case and thus the conservative and reasonable regimes are already equivalent; we therefore only consider this single regime in the following.

5.2 MLWE as a Lattice Problem

The search variant of the MLWE problem can be interpreted as a Bounded Distance Decoding (BDD) problem in a random family of q-ary R-lattices.

Definition 5.1 (BDD). *Let $\mathbf{B} \in R^{d\times d}$ be a basis of a full-rank lattice L, let $\mathbf{e} \in R^d$ be a small error such that $\|\mathbf{e}\| < \frac{1}{2}\lambda_1(L)$, and let $\mathbf{t} \in \mathbf{e} + L$ be a target. Given $(\mathbf{B}, \mathbf{t})$, compute the error $\mathbf{e}$.*

The BDD error $\mathbf{e} \in \mathbf{t} + L$ is the unique error satisfying $\|\mathbf{e}\| \leqslant \frac{1}{2}\lambda_1(L)$, as $\lambda_1(L)$ is the minimum distance between two distinct lattice points. To see that a $\mathsf{MLWE}_{q,m,k,\chi_1,\chi_2}$ instance $(\mathbf{A}, \mathbf{b} = \mathbf{A}\cdot\mathbf{s} + \mathbf{x}, \mathbf{s}, \mathbf{x})$ is indeed a BDD instance we first consider the following q-ary lattice

$$\Lambda_q(\mathbf{A}) := \{(\mathbf{z}, \mathbf{y}) \in R^m \times R^k : \mathbf{z} \equiv \mathbf{A}\mathbf{y} \bmod q\} \subset R^{m+k} \text{ with basis } \mathbf{B} := \begin{pmatrix} qI_m & \mathbf{A} \\ 0 & I_k \end{pmatrix}.$$

Due to $\Lambda_q(\mathbf{A})$ being q-ary we have that $qR^{m+k} \subset \Lambda_q(\mathbf{A})$ and thus we can simply consider its coefficients as being elements of R_q. For the target $\mathbf{t} := (\mathbf{b}, 0) \in R_q^{m+k}$ we can write

$$\mathbf{t} = (\mathbf{b}, 0) = (\mathbf{A}\cdot\mathbf{s} + \mathbf{x}, 0) = (\mathbf{x}, -\mathbf{s}) + (\mathbf{A}\cdot\mathbf{s}, \mathbf{s}) \in (\mathbf{x}, -\mathbf{s}) + \Lambda_q(\mathbf{A}),$$

where $\mathbf{e} := (\mathbf{x}, -\mathbf{s}) \leftarrow \chi_2^m \times \chi_1^k$ is a small error for appropriate distributions χ_1, χ_2. $\|(\mathbf{x}, -\mathbf{s})\| < \frac{1}{2}\lambda_1(\Lambda_q(\mathbf{A}))$ we thus indeed obtain a BDD instance $(\mathbf{B}, \mathbf{t})$ with unique error $(\mathbf{x}, -\mathbf{s})$. To understand how small the error $(\mathbf{x}, -\mathbf{s})$ is compared to the first minimum $\lambda_1(\Lambda_q(\mathbf{A}))$ we consider $\Lambda_q(\mathbf{A})$ as an unstructured $\mathbb{Z}$-lattice of dimension $d = (m+k)n$. The random lattice $\Lambda_q(\mathbf{A})$ typically follows the Gaussian Heuristic, which says that a lattice L of dimension d typically satisfies

$$\lambda_1(L) \approx \mathrm{gh}(L) := \frac{\det(L)^{1/d}}{\mathrm{vol}(\mathcal{B}_d)^{1/d}} \approx \sqrt{d/2\pi e}\cdot\det(L)^{1/d},$$

where $\mathrm{vol}(\mathcal{B}_d)$ is the volume of a d-dimensional unit ball. For $\Lambda_q(\mathbf{A})$ we have $\det(\Lambda_q(\mathbf{A})) = q^{mn}$ and thus we expect that

$$\lambda_1(\Lambda_q(\mathbf{A})) = \mathrm{gh}(\Lambda_q(\mathbf{A})) \approx \sqrt{(m+k)n/2\pi e}\cdot q^{m/(m+k)}.$$

The length $\|(\mathbf{x}, -\mathbf{s})\|$ of the error $(\mathbf{x}, -\mathbf{s}) \xleftarrow{\$} \chi_1 \times \chi_2$ depends on their precise distributions. Assuming $\chi_1 = \chi_2$ with a variance of σ^2 per coefficient we expect that

$$\mathbb{E}\left[\|(\mathbf{x}, -\mathbf{s})\|^2\right] = n(k+m)\cdot\sigma^2,$$

and thus $\|(\mathbf{x}, -\mathbf{s})\|/\lambda_1(\Lambda_q(\mathbf{A})) \approx \sqrt{2\pi e\sigma^2}\cdot q^{-m/(m+k)}$. For ML-KEM σ^2 is a small constant, and $m/(m+k) \geqslant \frac{1}{2}$, so the error is a factor at least $\Omega(\sqrt{q})$ smaller than the first minimum of the lattice.

This gap, along with the lattice dimension, will be the most important factor for the final security estimates. Now that we have rephrased the underlying MLWE problem as a purely geometrical problem over a lattice, we review the known techniques used to tackle it, using lattice reduction algorithms.

5.3 Lattice Attacks and Estimates

In this section we will consider the current best attacks on the *unstructured* lattice problems underlying the security of ML-KEM. The concrete estimate of the cost of these attacks determines the bit security of the scheme. We will discuss the exploitation of the algebraic structure coming from the module later in Section 5.4.

5.3.1 Lattice Reduction and SVP Solvers

Lattice reduction. The currently best known attacks on the BDD problem underlying ML-KEM all reduce to the computation of short lattice vectors, also known as the *Shortest Vector Problem* (SVP), and of a good basis of the lattice, known as *lattice reduction*. Lattice reduction algorithms reduce the problem of finding a good basis of a lattice of dimension d to the computation of short vectors in lattices of a lower dimension $\beta \ll d$. They give a trade-off between the shortness of the basis and the dimension β. The history of lattice reduction is quite old and can be traced back to the early work of Hermite in the 18th century. While there exists a plethora of variants of reduction algorithms, in the following we only consider the Block-Korkine-Zolotarev (BKZ) reduction algorithm [Sch87], which gives the most effective trade-off in practice.

On the output quality and its estimation. The following lemma gives a good estimation of the geometry of the lattice basis it outputs.

Lemma 5.2 (Profile under the Geometric Series Assumption (GSA)). *Let* $\mathbf{B}$ *be a BKZ-*β *reduced basis of dimension* d*, then under the Geometric Series Assumption and the Gaussian heuristic the Gram-Schmidt vectors* $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_d$ *of* $\mathbf{B}$ *satisfy*

$$\log(\|\tilde{\mathbf{b}}_i\|) \approx \frac{d+1-2i}{2} \cdot \log(\alpha_\beta) + \frac{\log(\det(\mathbf{B}^T\mathbf{B}))}{2d},$$

for any $1 \leqslant i \leqslant d$ *and where* $\alpha_\beta = \mathrm{vol}(\mathcal{B}_d)^{-\frac{2}{\beta(\beta-1)}}$.

The Geometric Series Assumption in Lemma 5.2 gives a heuristic indication of the behavior of the lattice reduction algorithm BKZ on the basis of the lattice. One can observe that by increasing the dimension β, also referred to as the *blocksize*, we decrease the norm $\|\mathbf{b}_1\| = \|\tilde{\mathbf{b}}_1\|$ of the first basis vector. For $\beta = 2$ we obtain an exponential approximation with $\|\mathbf{b}_1\| \leqslant 2^{O(d)} \cdot \mathrm{gh}(L(\mathbf{B}))$, while for $\beta = d$ we obtain $\|\mathbf{b}_1\| \approx \mathrm{gh}(L)$. However, BKZ furthermore gives guarantees on the Gram-Schmidt norms of the rest of the basis vectors, which will be important for some attacks.

The GSA is a good first-order approximation which is often sufficient for asymptotic estimates. For more precise concrete estimates one can use simulators which, on input a starting basis, try to emulate the BKZ reduction process [CN11; YD17; BSW18]. This captures the more precise, sometimes probabilistic, nature of BKZ, and has in particular a big effect on the first and last few Gram-Schmidt norms. Additional care has to be taken in the case of q-ary lattices, as the initial basis has non-standard Gram-Schmidt norms $q, \ldots, q, 1, \ldots, 1$ that form a so-called "Z-shape", see [AD21] for more details. In certain regimes, this can lead to additional weaknesses, for example when q is large for NTRU [ABD16; KF17; DW21], or when q is small [DEP23].

Progressive reduction. In practice, BKZ is almost never run with a fixed large blocksize from scratch. Instead, implementations use *progressive* BKZ [Aon+16; Xia+22], where the blocksize is increased gradually, e.g. $\beta_1 < \beta_2 < \cdots < \beta_t$. At each stage one performs several *rounds* (also called *tours*) of BKZ-β_j until the shape of the Gram–Schmidt norms stabilizes, before moving on to the next larger blocksize. This strategy amortizes the cost of the expensive high-dimensional SVP calls and leads to noticeably better reduced bases for a given total running time. In concrete security estimates, one typically accounts for this by charging the cost of all intermediate rounds towards the final target blocksize.

The SVP routine in BKZ. To compute a BKZ reduced basis we have to consider the exact SVP problem, that of computing the shortest vector of length $\lambda_1(L)$ in a lattice $L \subset \mathbb{R}^\beta$ of dimension β.

Definition 5.3 (SVP). *Let* $\mathbf{B} \in \mathbb{R}^{\beta \times \beta}$ *be a basis of a full-rank lattice* L*. Given* $\mathbf{B}$*, compute a shortest non-zero vector* $\mathbf{y} \in L$ *such that* $\|\mathbf{y}\| = \lambda_1(L)$*.*

Typically, one considers the case of random lattices, known to be the hardest instances. Here we have $\lambda_1(L) \approx \mathrm{gh}(L)$. There are two broad categories of heuristic algorithms to solve SVP in such random lattices: enumeration algorithms which take super-exponential time $\beta^{O(\beta)}$ and polynomial memory, and sieving algorithms which take single-exponential $2^{O(\beta)}$ time and memory.

Enumeration algorithms are branch-and-bound algorithms which recursively reduce the enumeration of short vectors in a lattice L to the enumeration of many short vectors in a lower-dimensional projected lattice $\pi(L)$. The precise projections and bounds used can have a significant impact on the total size of the enumeration tree, and therefore on the time complexity. The asymptotic best algorithm by Kannan [Kan83] takes time $\beta^{\beta/2e+o(\beta)}$ in the worst case [HS07; HS08]. Later, this was improved by [Alb+20a] to $\beta^{\beta/8+o(\beta)}$ in the setting of lattice reduction, as was already heuristically indicated as a lower bound in [Ngu10; HS10]. Further exponential $2^{O(\beta)}$ improvements followed from pruning of the search tree [SH95; GNR10], and from better parametrization within lattice reduction [Aon+16; Agg+20; LN25; Alb+21]. Overall, the currently best concrete estimation of the enumeration cost inside lattice reduction is given by $2^{0.125\beta\log(\beta)-0.654\beta+25.84}$ [Alb+21]. Enumeration algorithms benefit directly from the quantum speed-up by Grover's algorithm, leading to an asymptotic time cost of $\beta^{\beta/16+o(\beta)}$.

Sieving algorithms run in single-exponential time at the cost of a single-exponential memory usage. The core idea is to keep track of a long list of somewhat short lattice vectors $S \subset L$, while trying to find pairs $x, y \in S$ of distinct close lattice vectors such that $x - y$ is shorter than x or y. This new shorter vector is then inserted into the list S, replacing a longer vector. Heuristically, this process continues until the shortest vector is found whenever $|S| \geqslant (4/3)^{\beta/2+o(\beta)} = 2^{0.2075\beta+o(\beta)}$ [NV08]. As one needs to check all pairs of vectors in the list, the time complexity is at least $|S|^2 \geqslant (4/3)^{\beta+o(\beta)} = 2^{0.415\beta+o(\beta)}$. A long line of research on nearest-neighbour search methods decreased the time complexity further to $(3/2)^{\beta/2+o(\beta)} = 2^{0.292\beta+o(\beta)}$ [Laa15; BGJ15; BL16; Bec+16], which was shown to be optimal in this context [KL21]. High overheads in practice initially made sieving algorithms mostly of theoretical interest. This changed, however, after several (heuristic) sub-exponential [Duc18] and polynomial improvements [LM18; Alb+19]. The idea of [Duc18] is that lattice sieving does not only give a single shortest vector, but $|S| = (4/3)^{\beta/2+o(\beta)}$ of them. By finding these many short vectors in a lower-dimensional projected lattice $\pi(L)$ we can hope that at least one of them lifts to the shortest vector in L, where $\pi(L)$ can heuristically be of dimension $\beta - f$ for $f \approx \frac{\ln(4/3)\beta}{\ln(\beta/2\pi e)}$, essentially obtaining f "dimensions for free". The works [LM18; Duc18] furthermore propose applying lattice sieving progressively, by starting with finding short vectors in a smaller dimensional lattice, while increasing the lattice dimension step by step. As a result, the vectors in the set S are already somewhat short once the highest dimensions are reached, reducing the number of iterations needed there. These and other implementation improvements combined now make sieving algorithms superior to enumeration algorithms, both in theory and in practice already from dimension 90 or higher [Alb+19]. All current record SVP computations are achieved by sieving algorithms [Alb+19; DSW21; ZDY25]. While heuristic sieving algorithms are performing well in practice now, their sub-exponential factors in the time complexity, hidden in the $o(\beta)$ in the exponent, are still not entirely understood [Duc22b]. Furthermore, the influence of memory access costs when using exponential memory makes it unclear what the cost will be in cryptographic dimensions. For example, due to these memory access costs, the asymptotic best algorithm [Bec+16]

is currently beaten in practice by asymptotically worse algorithms [DSW21; ZDY25]. Still, for estimating the security of ML-KEM, one assumes the complexity $2^{0.292\beta+o(\beta)}$ of the asymptotically best algorithm, which can be seen as a conservative estimate.

Low memory and quantum sieving. Another line of research tries to reduce the memory complexity while increasing the time complexity, by looking at differences of k-tuples of vectors for $k \geqslant 3$ [BLS16; HKL18]. For $k = 3$ this leads, for example, to a time complexity of $2^{0.338\beta+o(\beta)}$ using $2^{0.1887\beta+o(\beta)}$ memory [CL23]. Grover-like algorithms can similarly be used to improve the time complexity of sieving algorithms. However, contrary to enumeration, they achieve far from the possible quadratic speed-up. A line of works [LMP15; Laa16; CL21; Hei21; Bon+23] only improved the classical time complexity of $2^{0.292d+o(d)}$ down to $2^{0.2563+o(d)}$. More concrete estimates seem to indicate that, even in unit cost quantum memory models, these exponential speed-ups are tenuous at best for cryptographic dimensions [Alb+20b; Dor+24]. With all the expected overhead of quantum computations, quantum sieving is so far not a threat. Furthermore, all sieving algorithms based on pairs, and thus also quantum algorithms, have a fundamental lower bound of $2^{0.2075d+o(d)}$ on the number of vectors that need to be stored.

We now move back to solving the MLWE problem.

5.3.2 Primal Attack

The primal attack solves the BDD problem underlying MLWE by reducing it to a unique Shortest Vector Problem (uSVP), which in turn is resolved using BKZ. It does so by embedding the BDD instance $\mathbf{t} \in \mathbf{e} + L$ in a lattice L of dimension d into a lattice L' of dimension $d+1$ for which the unique shortest vector $\mathbf{v} \in L'$ corresponds to the BDD error $\mathbf{e}$.

Concretely, let $\mathbf{B}$ be a basis of the lattice $L \subset \mathbb{R}^d$ and let $\mathbf{t} = \mathbf{B} \cdot \mathbf{y} + \mathbf{e} \in \mathbb{R}^d$ be a BDD instance with $\|\mathbf{e}\| < \frac{1}{2}\lambda_1(L)$. We now consider the lattice $L' \subset \mathbb{R}^{d+1}$ with the following basis:

$$\mathbf{B}' := \begin{pmatrix} \mathbf{B} & \mathbf{t} \\ 0 & c \end{pmatrix} \quad \text{for some constant } c > 0.$$

Note that we can construct $\mathbf{B}'$ using the public information $\mathbf{B}$ and $\mathbf{t}$, and that $\det(L') = c \cdot \det(L)$ has a similar determinant. We do have that

$$\mathbf{B}' \cdot \begin{pmatrix} -\mathbf{y} \\ 1 \end{pmatrix} = \begin{pmatrix} -\mathbf{B} \cdot \mathbf{y} \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{t} \\ c \end{pmatrix} = \begin{pmatrix} -\mathbf{B} \cdot \mathbf{y} \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{B} \cdot \mathbf{y} + \mathbf{e} \\ c \end{pmatrix} = \begin{pmatrix} \mathbf{e} \\ c \end{pmatrix} \in L',$$

and thus L' contains the vector $(\mathbf{e}, c)$. As $\mathbf{e}$ is small, and for suitably chosen c, the BDD error $\mathbf{e}$ thus corresponds to the short vector $(\mathbf{e}, c) \in L'$. As $\|\mathbf{e}\| < \frac{1}{2}\lambda_1(L)$, and for suitable $c > 0$, we actually expect that $(\mathbf{e}, c)$ is the unique shortest vector in L'.

We now proceed to recover the shortest vector $(\mathbf{e}, c) \in L'$. Recall that typically BKZ requires blocksize $\beta = d+1$ to recover the shortest vector of a lattice of dimension $d+1$. Under the premise that the shortest vector is unusually short $\|\mathbf{e}, c\| \ll \mathrm{gh}(L')$, BKZ however recovers it already with a much lower blocksize [GN08; AFG14; Alk+16].

Lemma 5.4 (BKZ for uSVP, [Alk+16]). *Let L be a lattice of dimension d and let $\mathbf{v} \in L$ be a shortest vector satisfying $\|\mathbf{v}\| \ll \mathrm{gh}(L)$. Then, under the GSA (Lemma 5.2), BKZ with blocksize β heuristically recovers $\mathbf{v}$ if*

$$\sqrt{\frac{\beta}{d}} \cdot \|\mathbf{v}\| < \mathrm{vol}(\mathcal{B}_d)^{\frac{d+1-2\beta}{\beta(\beta-1)}} \cdot \det(L)^{1/d}.$$

For example, asymptotically we get that if $\|\mathbf{v}\| = \mathrm{gh}(L)/\Theta(\sqrt{d})$, as will be the case for ML-KEM parameters, we only require a blocksize of $\beta = \frac{d}{2} + o(d)$. This (roughly) explains why the underlying BDD problem related to the key generation of ML-KEM corresponds to a lattice problem in dimension $(k+m)n = 1024, 1536$ and 2048 for ML-KEM-512, ML-KEM-768 and ML-KEM-1024 respectively.

Beyond these asymptotics Lemma 5.4 gives much more precise estimates that one can use for concrete estimates of the required blocksize β, and therefore of the cost of the primal attack. These estimates can be refined in several additional ways. First, the term $\sqrt{\frac{\beta}{d}} \cdot \|\mathbf{v}\|$ relates to the expected length of the shortest vector $\mathbf{v}$ after projecting it away from the first $d - \beta$ basis vectors. Instead of using the expectation one can model this length as a random variable under certain heuristic distributions on the vector [Dac+20; PV21]. Secondly, the right part $\mathrm{vol}(\mathcal{B}_d)^{\frac{d+1-2\beta}{\beta(\beta-1)}} \cdot \det(L)^{1/d}$ directly relates to the expectation of the Gram-Schmidt norm $\|\tilde{\mathbf{b}}_{d-\beta+1}\|$ of the basis after BKZ-β reduction following Lemma 5.2. One could replace this rough estimate by the already mentioned BKZ simulators [CN11; YD17; BSW18]. Thirdly, for the case of LWE, to obtain the lowest blocksize estimate β from Lemma 5.4, it can be beneficial to not use all the samples. For example, for ML-KEM parameters the optimal number of samples is often slightly below $m = k$, and therefore an optional $k+1$-th sample does not influence the security estimate.

Further improvements in the runtime can be made by a two-step approach [Xia+24], where the basis is first reduced using several SVP calls in dimension β, after which a slightly larger SVP call in dimension $\beta' > \beta$ is made to recover the final solution. This compensates for the fact that BKZ makes many calls, compared to only a single SVP call that is needed for the last step.

5.3.3 Dual Attack

The dual attack uses vectors in the dual lattice to distinguish between BDD targets that lie close to the lattice, and targets that lie far away from the lattice [MR09; Alk+16; Alb17]. For a lattice $L \subset \mathbb{R}^d$ its dual lattice L^* is given by those vectors that have an integer inner product with all lattice vectors, i.e.,

$$L^* := \{\mathbf{y} \in \mathbb{R}^d : \forall \mathbf{v} \in L, \langle \mathbf{y}, \mathbf{v} \rangle \equiv 0 \bmod 1\}.$$

As a result, if we look at a BDD instance $\mathbf{t} = \mathbf{v} + \mathbf{e} \in \mathbb{R}^d, \mathbf{v} \in L$ and a dual vector $\mathbf{y} \in L^*$, then

$$\langle \mathbf{t}, \mathbf{y} \rangle \equiv \langle \mathbf{v}, \mathbf{y} \rangle + \langle \mathbf{e}, \mathbf{y} \rangle \equiv \langle \mathbf{e}, \mathbf{y} \rangle \bmod 1.$$

Now if both $\mathbf{e}$ and $\mathbf{y}$ are short, then $|\langle \mathbf{e}, \mathbf{y} \rangle| \leqslant \|\mathbf{e}\| \cdot \|\mathbf{y}\|$ is biased towards 0, or equivalently, $\langle \mathbf{t}, \mathbf{y} \rangle$ is expected to be close to an integer. Otherwise, if $\mathbf{t}$ is a uniform target, then $\langle \mathbf{t}, \mathbf{y} \rangle \bmod 1$ is also uniform. The differences in these distributions can be used to distinguish if $\mathbf{t}$ is a BDD instance or a uniform target.

The *dual attack* generally proceeds as follows: first a list $S \subset L^*$ of short dual vectors is computed, after which $\langle \mathbf{t}, \mathbf{y} \rangle \bmod 1$ is computed for all $\mathbf{y} \in S$, and a statistical test is performed to decide if $\mathbf{t}$ is most likely a BDD instance or a uniform target. Note that the computation of the list $S \subset L^*$ can be seen as a preprocessing step with time complexity $> |S|$, after which computing the inner

products only has a cost of $|S|$. It is therefore common that one performs the preprocessing once, to then solve multiple decisional BDD instances. We will discuss how these multiple decisional BDD instances are created in the next section on hybrid attacks.

In the last years there has been a significant increase in attention for the dual attack. This was especially the case due to claims of strong improvements to the concrete costs of dual attacks [EJK20; GJ21; Li+21; MAT22], indicating improvements over the primal attack. In particular, the report by MATZOV [MAT22] made a large impact, claiming to push KYBER (now ML-KEM) below the NIST security levels. These attacks relied on the assumption that the vectors $\mathbf{y} \in S$, and in particular their inner products $\langle \mathbf{t}, \mathbf{y} \rangle \bmod 1$, all behave independently and can be analysed as such. In [DP23b; BW25] this *independence heuristic* was questioned and shown to be false in certain regimes, and in particular in the regime of MATZOV's claims [MAT22]. Several follow-up works improved the analysis without the independence heuristic, either provable or backed up by experiments [DP23a; PS24]. Furthermore, the impact on the MATZOV attack was investigated further in [Car+25].

5.3.4 Hybrid Attacks

Hybrid attacks are a combination of standard lattice attacks and combinatorial guessing attacks: part of the secret is guessed, reducing one instance of the problem to multiple easier instances in a *lower-dimensional* or *sparser* lattice. In certain circumstances, especially when the secret has low entropy, this can give a good trade-off.

Hybrid attacks were first considered in the context of NTRU [How07; Hir+09], but were later extended to LWE [Alb17]. Concretely, consider a lattice L, a BDD instance $\mathbf{t} = \mathbf{v} + \mathbf{e}$, and a dual vector $\mathbf{w} \in L^*$ for which we know that $\langle \mathbf{e}, \mathbf{w} \rangle \in T$ for some small set T. Now, if we know that $\langle \mathbf{e}, \mathbf{w} \rangle = c \in T$, then we obtain an equation $\langle \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{t} - \mathbf{e}, \mathbf{w} \rangle = \langle \mathbf{t}, \mathbf{w} \rangle - c = c'$ on $\langle \mathbf{v}, \mathbf{w} \rangle$, where c' is known. Note that $\langle \mathbf{v}, \mathbf{w} \rangle = c'$ defines some hyperplane H in which the solution lies, i.e., $\mathbf{v} \in H \cap L$. Now for any $\mathbf{x} \in H \cap L$ we obtain a lattice $L' = -\mathbf{x} + (H \cap L)$ of dimension $d - 1$ and volume $\det(L') = \|\mathbf{w}\| \cdot \det(L)$. Furthermore, the original BDD instance is translated into a new BDD instance $\pi_H(\mathbf{t}) - \mathbf{x}$ on L' with solution $\mathbf{v} - \mathbf{x}$, from which $\mathbf{v}$ can be derived. By guessing the value of $c \in T$, we can thus turn the d-dimensional BDD instance into $|T|^t$ BDD instances over the $d - t$-dimensional lattice L'. One can repeat this step t times, leading to $|T|^t$ instances over a $d - t$ dimensional lattice. Alternatively, if $T = T' + q\mathbb{Z}$, we can instead consider a modular constraint $\langle \mathbf{e}, \mathbf{w} \rangle \equiv c \bmod q$, leading to T' BDD instances in a d-dimensional but sparser lattice L'.

For LWE as a BDD instance in the q-ary lattice $\Lambda_q(\mathbf{A})$, one can, for example, always consider a dual unit vector $\mathbf{w}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$, such that $\langle \mathbf{w}_i, \mathbf{e} \rangle = e_i$ is a coefficient of the BDD error. The number of guesses depends on the precise error or secret distribution used, e.g. for binary LWE we have $|T|^t = 2^t$, while for ML-KEM we have $|T|^t \geqslant 5^t$ to reduce the dimension by t. Note that, depending on the distribution, one does not necessarily have to consider all guesses; for example, for very sparse distributions it can be sufficient to only guess errors with a few non-zero coefficients [Alb17; ACW19].

What remains is to solve the $|T|^t$ BDD instances in the $d - t$ dimensional lattice L'. This is also known as Batch-BDD or the Bounded Distance Problem with Preprocessing (BDDP), as one can perform some pre-computation on L' that can be amortized over all BDD instances. Here one can again consider both a primal or a dual attack approach.

For the hybrid primal attack one would typically first reduce the lattice L', after which Babai's nearest plane algorithm [ACW19; Kar+25b; PV25], or more advanced BDDP algorithms [EK20; DLW20; Ber22; Ber23; Kar+25a], are used to solve the BDD instances. Especially interesting is the

combination of this approach with the *randomized iterative slicer*, which solves each BDDP instance with a relatively small amortized cost, obtaining an exponential asymptotic speed-up [Ber23] if $|T|$ is finite. More precisely, asymptotically the bit security is reduced by a factor $(1-\mathcal{K})$ [Ber23; Kar+25a] where $\mathcal{K} = (1+\frac{\mathcal{H}(\chi)}{0.058})^{-1}$ and $\mathcal{H}(\chi)$ is the Shannon entropy of the error distribution χ on a single coordinate. For ML-KEM we have, for example, $\mathcal{H}(B_3) = 2.333$ and $\mathcal{K} \approx 0.024$, which would amount to roughly 3.5, 5 and 6.5 bits for the three security levels respectively. This hybrid attack was also shown to outperform the standard primal attack in practice [Kar+25a] by a small factor. At this point, however, it remains unclear whether this is a result of implementation details or of an asymptotic speed-up, and therefore also what the concrete impact will be on cryptographic dimensions.

For the hybrid dual attack one typically proceeds by pre-computing a list $S \subset (L')^*$ of short dual vectors. The inner products with the targets and the dual vectors are then used to distinguish which ones are real BDD targets, i.e., lie close to the lattice, and which ones are not. Note that one only uses the dual attack as a distinguisher here, to determine which of the guesses was correct. Note that such a hybrid version is almost always better than a standard dual attack as it amortizes the cost of computing the many dual vectors over several targets. After the first hybrid dual attacks [Alb17; EJK20; Kar+25b] several improvements followed. Firstly, note that one has to compute the inner product between a large list of $|S|$ vectors and a list of $|T|^t$ targets, leading to a cost of $O(|S| \cdot |T|^t)$ inner products. Typically, these targets are highly structured; for example, they might form an additive group modulo the lattice L. In these cases one can use FFT techniques to improve the cost down to $\tilde{O}(\max\{|S|, |T|^t\})$ [GJ21]. While these group structures often do not appear in the case of small or sparse errors or secrets, one can additionally either only guess their most significant bits or perform modulus switching techniques to still fall into this case and decrease the size of $|T|$ [GJ21; MAT22]. In [Car+25] this was further improved by even better coding-theoretic techniques to replace the modulus switching step.

5.3.5 Aurora-Ge and BKW

We shortly discuss the Aurora-Ge [AG11] and BKW [BKW00; KF15] algorithms. These generally require many samples or an error distribution with a very small support. For ML-KEM, after converting to plain LWE, we obtain at most $(m+k)d \leqslant (2k+1)d$ samples, which, along with the error support of size at least 5, is far from sufficient to make these attacks competitive.

The Aurora-Ge [AG11] attack proceeds by modelling the LWE problem as a system of polynomial equations, which is then solved using standard linearization or Gröbner bases techniques. The idea is that if the LWE errors e_i take at most k values $C = \{c_1, \ldots, c_k\}$, then $f_i(X) := f_i(X_1, \ldots, X_n) = b_i - \sum_{j=1}^{n} A_{ij} \cdot X_j = e_i \in C$, and thus for each sample we obtain a polynomial equation $\prod_{l=1}^{k}(f_i(X) - c_i) = 0$ of degree k in n variables. As such a system typically has $\Omega(n^k)$ distinct monomials, one requires at least that number of samples and thus equations to linearise and solve the system. Gröbner basis techniques [Alb+14; Ste24] allow for further trade-offs between the time cost and the number of samples. For ML-KEM we have $k \geqslant 5$ and only on the order of $O(n)$ samples, making linearization not applicable and the Gröbner basis attack far from competitive, both asymptotically and concretely.

The BKW attack, initially developed by Blum, Kalai and Wasserman for LPN [BKW00], is a combinatorial attack on LWE that runs in sub-exponential time when many samples are available [KF15]. The idea is to find many pairs of LWE samples $(b_i, a_i), (b_j, a_j) \in \mathbb{Z} \times \mathbb{Z}^n$ such that the first $k > 0$

coefficients of $a_i - a_j$ are 0. Removing these coefficients, we obtain a new LWE sample $(b_i - b_j, a_i - a_j)$ of LWE dimension $n - k$, but with the same (truncated) secret. We can repeat this process until the LWE problem becomes feasible. Such an attack has two problems: firstly, the error $e_i - e_j$ of the new LWE sample is larger than the original one, and secondly, we need about $m = \Theta(\sqrt{q^k})$ samples to find even a single pair that collides on the first k coefficients. To use this technique recursively we would constantly need about $m = \Theta(q^k)$ samples. The parameter k cannot be too small, as otherwise the error would increase too much. All in all, while some further trade-offs are possible, for typical parameters the BKW algorithm requires a large number of samples to perform well, and is therefore not applicable to ML-KEM.

5.4 Structured Attacks

Structured attacks aim to exploit the algebraic structure of ML-KEM's underlying module-LWE $\mathsf{MLWE}_{q,m,k}$ instance – namely that the public-key lattice is an R-module for the 2-power cyclotomic ring $R = \mathbb{Z}[x]/(x^{256}+1)$ of degree $n = 256$ with small module rank $m+k \in \{4, \ldots, 9\}$. When dealing with such structured lattices, two families of attacks and algorithms are relevant.

(i) ***Algebraic attacks on ideal lattices.*** Several results exploit cyclotomic-ideal structure – e.g., recovering short generators of principal ideals and finding "mildly short" vectors, sometimes in quantum polynomial time for related problems – showing *quantum* speedups for *ideal-SVP/PIP*-type tasks [Cra+16; CDW17]. However, these do not translate into attacks on the average-case *module-LWE* instances underlying ML-KEM, and no concrete parameter break or asymptotic improvement against ML-KEM's choices is known.

(ii) ***Module-/ideal-aware lattice reduction.*** Dedicated reductions for module and ideal lattices (e.g., module-LLL) preserve the R-module structure and can yield constant-factor savings (better preprocessing, slightly altered slope predictions) compared to treating the lattice as unstructured, but they do not provide an asymptotic advantage over state-of-the-art BKZ/sieving at ML-KEM dimensions; concrete estimates for ML-KEM therefore still model cost essentially as generic lattice reduction in dimension $n(m+k)$ after coefficient embedding [Lee+19; KEF20; MS20; DEP25].

We give more details about these attacks in Sections 5.4.1 to 5.4.3.

Scope and caveats. Known "subfield/weak-ring" attacks target non-cyclotomic or specially structured rings, or overstretched-modulus NTRU variants, and do not apply to ML-KEM's power-of-two cyclotomic ring and small module rank; indeed ML-KEM' s specification selects parameters precisely to avoid such pathologies [ABD16; KF17; DW21]. In short, while structure enables efficient implementation (NTT, automorphisms) and motivates specialized algorithms for some *ideal*-lattice problems, there is currently no structured attack that asymptotically outperforms generic lattice attacks on ML-KEM' s module-LWE; security estimates therefore continue to rely on generic BKZ/sieving models with, at most, modest constant-factor allowances for structure.

5.4.1 Ideal Lattices

The special case of rank 1 module lattices, also known as *ideal lattices*, can be seen as the most structured case among all module lattices. While this makes them interesting in terms of efficiency for cryptographic purposes and in terms of structure for, e.g., Fully Homomorphic Encryption

schemes [Gen09], it also could introduce vulnerabilities. Indeed, for ideal lattices there is a gap between the best classical and the best quantum algorithms: while classical algorithms can efficiently achieve an approximation ratio of $2^{O(n)}$ using LLL, quantum algorithms achieve a ratio of $2^{O(\sqrt{n})}$ for ideal lattices over cyclotomic fields [Cra+16; CDW17; DPW19; CDW21].

The same quantum algorithm for ideal lattices also applies to general number fields [PHS19] although only heuristically, with up to a $2^{O(n)}$ preprocessing cost depending on the number field, and with a potentially worse approximation ratio depending on the discriminant of the number field.

A special case of the above algorithms occurs in the setting when one considers a principal ideal $I \subset R$ generated by an unusually short principal generator $g \in R$. In this case this short generator g can be recovered in quantum polynomial time [Cra+16].

While these quantum algorithms on general ideal lattices do not reach an approximation ratio that would threaten the security of basic cryptographic primitives, the existence of this quantum advantage has mostly stopped the adoption of lattice-based cryptography based on ideal lattices. For example, ML-KEM uses module lattices with rank at least 4 to fall out of the scope of this line of attacks.

5.4.2 Class and (S-)Unit Groups

The quantum algorithms for ideal-SVP rely on the class and (S-)unit group of the underlying number ring R. Indeed, while computing these groups classically takes sub-exponential time [BF14; Bia+17], they can quantumly be computed in polynomial time [Eis+14; BS15; BDF20; BF25]. All these quantum algorithms rely on the same *continuous hidden subgroup* problem framework. For a group G of the form $G = \mathbb{R}^s \times \mathbb{Z}^t$, a subgroup $H \subset G$ and query access to a (sufficiently non-flat) H-periodic function $f : G \to \mathbb{C}$ satisfying $G(g+h) = G(x)$ for all $g \in G, h \in H$, the continuous hidden subgroup problem asks to recover (generators of) H. The problem of computing the class group, (S-)unit group and the recovery of a principal generator can all be reduced to an appropriate instance of this period-finding problem, which can be solved efficiently by a quantum computer. For more technical details about the continuous hidden subgroup problem we refer to [BF25].

5.4.3 Module-LLL

Recall that the lattice reduction algorithms like LLL and BKZ reduce the problem of computing approximately short vectors to several exact SVP instances in a lower dimension $\beta \geqslant 2$. *Module lattice reduction* algorithms proceed similarly, but they make use of the module structure, for example such that the lower-dimensional exact SVP instances are still acting on module lattices [Lee+19; MS20; DEP25], or to use the algebraic structure to improve the efficiency [KEF20].

Module-LLL or Module-BKZ [Lee+19; MS20; DEP25] indeed reduce the problem of lattice reduction of a rank $r > 2$ R-module lattice to that of a rank $2 \leqslant \beta < r$ R-module lattice. One obtains a trade-off between r, β, the discriminant of R, and the reached approximation radius. If one solves the rank β R-module SVP problem using a classical SVP algorithm in $\mathbb{Z}$-dimension βn, then this trade-off is worse than the classical BKZ algorithm for the power-of-two cyclotomic field used by ML-KEM. However, for fields with smaller discriminants the trade-off can lead to subexponential speed-ups [DEP25].

Note that module-LLL and module-BKZ only reduce to the module-SVP problem with rank at least 2, so the earlier mentioned ideal-SVP attacks are not relevant. This seems to indicate a hardness gap between rank 1 and rank at least 2 module lattice problems.

Another approach by [KEF20] is to use the algebraic structure to improve the runtime of the classical LLL (and in principle BKZ) algorithms. This can reduce the cost by polynomial factors, which especially for the already polynomial-time LLL algorithm can be beneficial. While such improvements can give small practical speed-ups for attacks on ML-KEM, they are insignificant compared to the large cost of the SVP calls.

5.5 Concrete Security Estimates

From these attack baselines we can estimate numerically the security given by the different parameter sets. To generate these figures, we relied on the state-of-the-art estimator from Albrecht et al. [APS15], available at https://github.com/malb/lattice-estimator.

5.5.1 Most Conservative Estimates via CORE-SVP Costing

To be as conservative as possible, we first rely on the so-called *CORE-SVP* costing model for lattice reduction, which forgets all polynomial factors beyond the main sieving cost subroutine in dimension β, which is thus costed as 0.292β bits following the asymptotically fastest sieve [Bec+16]. These costs are reported in Table 5. For consistency with the literature—and in particular with the original KYBER submission document [Ava+21], we used the *reasonable error modelling* as described in Section 5.1, but we also showcase the effect of the *no bit dropping* model (labeled as ML-KEM-512$^{\text{no drop}}$ in the table) as it provides the most conservative estimates. As mentioned, for ML-KEM-768 and ML-KEM-1024, there is no difference in these models.

Table 5: ML-KEM concrete security estimates in bits under CORE-SVP cost model. Best attack cost is in bold and corresponds to the dual attack, hybridized with some enumeration as described in Section 5.3.4.

Param set	NIST cat.	Primal attack	Dual attack (hybrid)	Combinatorial attacks
ML-KEM-512$^{\text{no drop}}$	1	115	120 (**112**)	167 (coded BKW)
ML-KEM-512	1	119	124 (**115**)	179 (coded BKW)
ML-KEM-768	3	182	189 (**174**)	238 (coded BKW)
ML-KEM-1024	5	255	264 (**242**)	310 (coded BKW)

5.5.2 Refined Estimates

We also consider a more refined cost model following [MAT22]. This model tries to estimate the costs of the progressive BKZ algorithms, the concrete cost of the BDGL sieve [Bec+16], and the probabilistic nature of the error distribution. Additionally, for a more refined estimate, matching the cost model, one should also use a BKZ simulator instead of the Geometric Series Assumption; however, this is currently not supported for the (hybrid) dual attacks in the estimator. These more refined bit-cost estimates are shown in Table 6. Here too, we use the reasonable error modeling as the base case, but also show the no bit dropping model for conservative costing.

Comparing the estimator to the literature, [Car+25] reports 0.2 to 2.5 bits lower for the dual hybrid attack after some additional improvements under the same cost model. Furthermore, while

in the current estimator the primal hybrid attacks are not shown to be effective, the works [Ber23; Kar+25a] indicate that a small speed-up is possible over the regular primal attack.

While the security estimates reported in Table 6 are slightly below the NIST bit-security levels of 143, 207 and 272 bits respectively, it is important to note that these more refined numbers should be seen as lower bounds for the current best known attacks. For example, they do not account for the basis quality loss coming from progressive sieving (+2.5 bits [Duc22a]), or for the overheads in the BDGL sieve [Duc22b] (+5 at security level 1), and some of the constants in the cost model are chosen optimistically. Additionally, they do not account for the cost of memory storage or access, which can turn out to be the largest cost in practice.

Table 6: ML-KEM concrete security estimates in bits under GSA and MATZOV cost model. These estimates should be interpreted as more refined lower-bounds. Best attack cost is in bold, corresponding to a dual attack hybridized with enumeration Section 5.3.4.

Param set	NIST cat.	Primal attack	Dual attack (hybrid)
ML-KEM-512$^{\text{no drop}}$	1	136.8	146.2 (**136.0**)
ML-KEM-512	1	140.2	149.9 (**139.7**)
ML-KEM-768	3	201.0	214.3 (**196.4**)
ML-KEM-1024	5	270.7	288.5 (**262.3**)

Taking this into account, all standardized ML-KEM parameter sets remain well within their target security categories even when the most optimistic attack parameters are assumed.

5.6 Decryption-Failure Attacks and Weak Keys in ML-KEM

A *decryption-failure (DF) attack* exploits the rare event that Decaps produces a key that does not match the encapsulator's key. Concretely, letting $\mathsf{ct} = (\mathbf{u}, v)$ be a ciphertext and $\mathbf{s}$ the secret, decapsulation essentially forms (modulo the technicalities yielded by the decompression) $w = v - \mathbf{s}^\top \cdot \mathbf{u} \bmod q$ and recovers bits by rounding the coefficients of w; a failure occurs only if noise pushes some coefficient across the decision boundary. If there is an observable distinction between such "success" and "failure" (different return values, timing, logs, or protocol branching, for instance), the adversary obtains a *binary oracle* [D'A+19]. The idea of decryption failure attacks is then to *boost* the failure rate by crafting ciphertexts whose induced error is near the boundary, and use the oracle's bit to run statistical tests on linear forms in $\mathbf{s}$; "directional" variants iteratively nudge (u, v) along locally more failure-prone directions to accelerate information extraction [D'A+19; DB22]. In a multi-target setting, *weak keys* are those secrets for which a fixed family of crafted ciphertexts happens to sit closer to the boundary—yielding a slightly higher failure probability and allowing an attacker to select weak users (or to amortize precomputation across many public keys) [DB22]. DF attacks can also be *amplified* operationally if the decapsulator uses a public key that is not the one matching its secret (e.g. due to storage/PK-substitution issues), which perturbs the effective noise and raises the failure rate [Flu+25]. In ML-KEM, however, properly designed systems are resilient: (i) parameters make the baseline failure probability δ negligible, so harvesting enough genuine failures is computationally infeasible [Nat24b][1]; (ii) the Fujisaki–Okamoto (FO) transform

[1]Remark that the standardized API uses implicit rejection so that a correct and secure implementation exposes no success/failure bit at all. However, as mentioned in Remark 3.1, in a more involved protocol, an implicit failure

binds the coins to a hash of the public key, which blocks multi-target precomputation and mitigates weak-key leverage [Ava+21; Sch22b; Sch22a]. With ML-KEM' s parameters and FO binding even accidental leakage leaves little room for practical failure exploitation without an oracle specifically *boosting* the failure probability; the remaining burden is mostly implementational: constant-time Decaps, identical code paths and logging across outcomes, storing/verifying the public key alongside the secret, and conservative query/rate limits [Flu+25]. We provide more details in Section 6.2.3.

5.7 Summary

Across all examined attack classes—lattice, hybrid, structural, and decryption-failure—no practically exploitable weakness is known for ML-KEM at any parameter level. Concrete estimates confirm that the schemes maintain security margins exceeding NIST's target categories even under aggressive classical and quantum cost assumptions.

might abort the protocol, giving de facto access to this bit, making the need for δ to be negligible in any case.

6 Implementation Details: Performance and Security

This section discusses subjects related to the implementation of ML-KEM in practice. In particular, Section 6.1 provides details about the algorithms used to implement ML-KEM. This includes low level functions implementing polynomial arithmetic, binomial sampling and the symmetric primitives that are used for cryptographic hashing and deterministic randomness generation.

In Section 6.2 the side channel security of ML-KEM implementations is studied. It provides background on side channel analysis and explains how side channel attacks on ML-KEM are performed. The focus is on the decapsulation, which is the operation in ML-KEM that is most vulnerable to physical attacks. Several countermeasures to protect against side channel attacks are also discussed.

Section 6.4 contains a comparison of results of hardware implementations of ML-KEM. The goal of this section is to provide some intuition regarding the computation time and area usage of practical implementations. It is also shown that the cost of protecting against side channel analysis can be considerable.

6.1 A Closer Look at the Algorithms of ML-KEM

This subsection provides background on the computation of the operations in ML-KEM that are particularly relevant to implementation security.

6.1.1 Building Blocks

Message encoding and decoding. The random 32-byte message generated during Encrypt in Encaps determines the shared secret. Since the confidentiality of the shared secret must be guaranteed, the operations that process the message are therefore of particular interest.

During encapsulation, a random message of 32 bytes is encoded to be hidden in the plaintext. The encoding process first converts the 32 bytes to 256 bits using the algorithms in Figure 4. Bits equal to 0 have to be mapped to 0 in $\mathbb{Z}_q$ and bits equal to 1 are mapped to $\frac{q}{2}$ in $\mathbb{Z}_q$. This is done using the $\mathsf{Decompress}_d$ function described in Section 2.3.5, with $d = 1$. The result is a message polynomial μ with coefficients in $\{0, \frac{q}{2}\}$ that is hidden in ciphertext part v.

During Decaps the original message bytes must be recovered from the ciphertext. This is done by applying the $\mathsf{Compress}_1$ function (explained in Section 2.3.5) to the intermediate result w in line 5 of Decrypt in Figure 8. This function maps elements from $\mathbb{Z}_q$ that are closer to 0 mod q than to $\frac{q}{2}$ to 0, and elements that are closer to $\frac{q}{2}$ are mapped to 1. The resulting bits are converted back to bytes using the BitsToBytes function from Figure 4.

Number Theoretic Transform (NTT). The NTT is a variant of the Fast Fourier Transform over the finite field $\mathbb{Z}_q$. It takes as input a polynomial in R_q and evaluates it in the powers of a primitive 512-th root of unity in $\mathbb{Z}_q$. The result is a vector of length 256 with coefficients in $\mathbb{Z}_q$, in the *NTT domain*. Polynomial multiplication between two polynomials a and b in the NTT domain is computed by multiplying their coefficients point-wise:

$$a \cdot b = (a_0 \cdot b_0 \mod q, a_1 \cdot b_1 \mod q, \ldots, a_{n-1} \cdot b_{n-1} \mod q). \tag{3}$$

This means that polynomial multiplication in the NTT domain can be computed in only $n = 256$ multiplications in $\mathbb{Z}_q$, whereas the Schoolbook polynomial multiplication method would require n^2

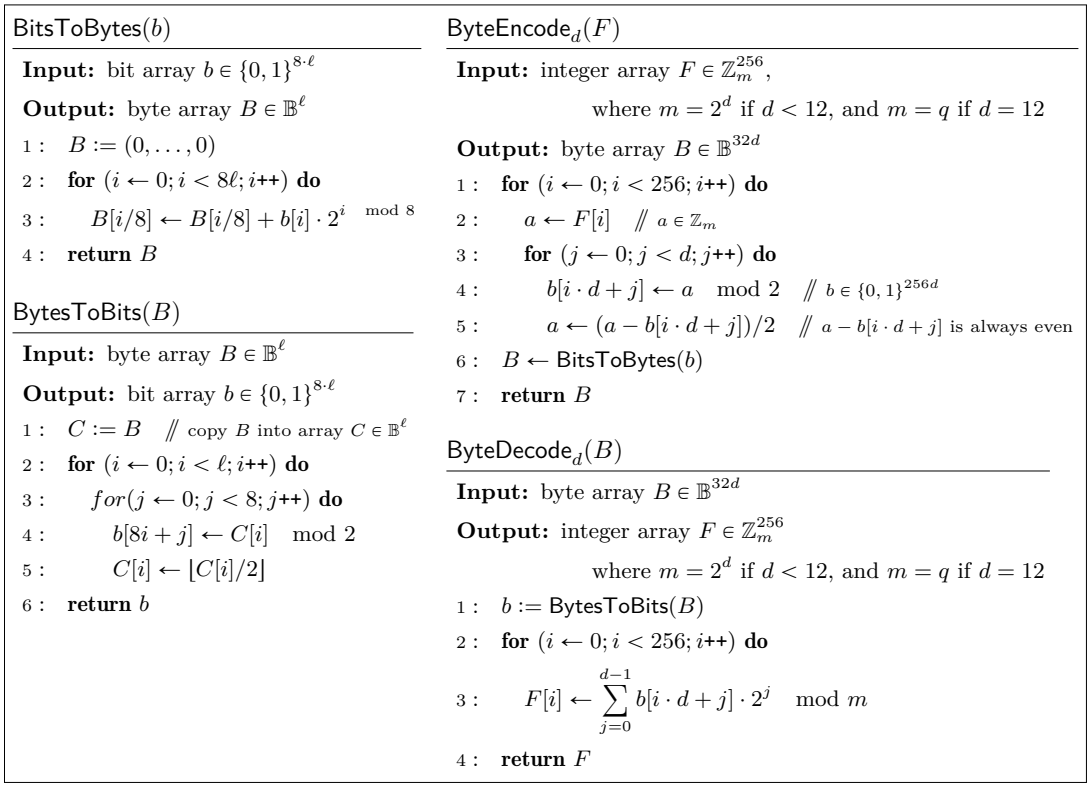


Figure 4: Algorithms BitsToBytes, BytesToBits, ByteEncode, and ByteDecode.

multiplications in $\mathbb{Z}_q$. Since the NTT itself is a linear operation, polynomial addition can simply be computed in the same way as in the *time* domain:

$$a + b = (a_0 + b_0 \mod q, a_1 + b_1 \mod q, \ldots, a_{n-1} + b_{n-1} \mod q). \tag{4}$$

Therefore polynomial multiplication, which can be a major performance bottleneck, can be sped up using the NTT. Once all polynomial arithmetic has been computed, the results must be transformed back into the time domain before applying non-linear operations. The inverse transform follows a similar structure and is also shown in Figure 5.

In ML-KEM, an *incomplete* NTT is used. That is, a polynomial in the time domain is evaluated by substituting x^2 by the powers of the 256-th root of unity ζ. The result is a vector of 128 polynomials of degree 1. The algorithm that computes the incomplete NTT is shown in Figure 5. In order to multiply two polynomials that are transformed by the incomplete NTT, the MultiplyNTTs algorithm from Figure 6 is computed. The product of two polynomials in the incomplete NTT domain can be computed by pair-wise multiplying the degree 1 polynomials of the two vectors. The pair-wise multiplication is defined by BaseCaseMultiply in Figure 6.

$\mathsf{NTT}(f)$

Input: array $f \in \mathbb{Z}_q^{256}$ // the coefficients of the input polynomial

Output: array $\hat{f} \in \mathbb{Z}_q^{256}$ // the coefficients of the NTT of the input polynomial

1 : $\hat{f} := f$ // will compute in place on a copy of input array
2 : $i := 1$
3 : **for** $(\mathsf{len} \leftarrow 128; \mathsf{len} \geqslant 2; \mathsf{len} \leftarrow \mathsf{len}/2)$ **do**
4 : **for** $(\mathsf{start} \leftarrow 0; \mathsf{start} < 256; \mathsf{start} \leftarrow \mathsf{start} + 2 \cdot \mathsf{len})$ **do**
5 : $\mathsf{zeta} \leftarrow \zeta^{\mathsf{BitRev}_7(i)} \mod q$
6 : $i \leftarrow i + 1$
7 : **for** $(j \leftarrow \mathsf{start}; j < \mathsf{start} + \mathsf{len}; j\mathsf{++})$ **do**
8 : $t \leftarrow \mathsf{zeta} \cdot \hat{f}[j + \mathsf{len}]$ // steps 8-10 done modulo q
9 : $\hat{f}[j + \mathsf{len}] \leftarrow \hat{f}[j] - t$
10 : $\hat{f}[j] \leftarrow \hat{f}[j] + t$
11 : **return** $\hat{f}$

$\mathsf{NTT}^{-1}(\hat{f})$

Input: array $\hat{f} \in \mathbb{Z}_q^{256}$ // the coefficients of input NTT representation

Output: array $f \in \mathbb{Z}_q^{256}$ // the coefficients of the inverse NTT of the input polynomial

1 : $f := \hat{f}$ // will compute in place on a copy of input array
2 : $i \leftarrow 127$
3 : **for** $(\mathsf{len} \leftarrow 2; \mathsf{len} \leqslant 128; \mathsf{len} \leftarrow \mathsf{len} \cdot 2)$ **do**
4 : **for** $(\mathsf{start} \leftarrow 0; \mathsf{start} < 256; \mathsf{start} \leftarrow \mathsf{start} + 2 \cdot \mathsf{len})$ **do**
5 : $\mathsf{zeta} \leftarrow \zeta_{2 \cdot \mathsf{len}}^{-\mathsf{BitRev}_7(i)} \mod q$
6 : $i \leftarrow i - 1$
7 : **for** $(j \leftarrow \mathsf{start}; j < \mathsf{start} + \mathsf{len}; j\mathsf{++})$ **do**
8 : $t \leftarrow f[j]$
9 : $f[j] \leftarrow t + f[j + \mathsf{len}]$ // steps 9-10 done modulo q
10 : $f[j + \mathsf{len}] \leftarrow \mathsf{zeta} \cdot (t - f[j + \mathsf{len}])$
11 : $f \leftarrow f \cdot 3303 \mod q$ // multiply every entry by $3303 = 128^{-1} \mod q$
12 : **return** f

Figure 5: Algorithms NTT and NTT^{-1}.

Binomial sampling. Both key generation and $\mathsf{Encrypt}$ require the sampling of polynomials whose coefficients follow the binomial distribution for a fixed parameter η. To sample one single coefficient from the binomial distribution, two uniformly random bit vectors $x = (x_0, \ldots, x_{\eta-1})$ and $y = (y_0, \ldots, y_{\eta-1})$ in $\mathbb{Z}_2^\eta$ are generated, and the sample

MultiplyNTTs$(\hat{f}, \hat{g})$

Input: Two arrays $\hat{f} \in \mathbb{Z}_q^{256}$ and $\hat{g} \in \mathbb{Z}_q^{256}$ // the coefficients of two NTT representations

Output: An array $\hat{h} \in \mathbb{Z}_q^{256}$ // the coefficients of the product of the inputs

1 : **for** $(i \leftarrow 0; i < 128; i$++$)$ **do**

2 : $(\hat{h}[2i], \hat{h}[2i+1]) \leftarrow \mathsf{BaseCaseMultiply}(\hat{f}[2i], \hat{f}[2i+1], \hat{g}[2i], \hat{g}[2i+1], \zeta^{2\mathsf{BitRev}_7(i)+1})$

3 : **return** $\hat{h}$

BaseCaseMultiply$(a_0, a_1, b_0, b_1, \gamma)$

Input: $a_0, a_1, b_0, b_1 \in \mathbb{Z}_q$ // the coefficients of $a_0 + a_1 X$ and $b_0 + b_1 X$

Input: $\gamma \in \mathbb{Z}_q$ // the modulus is $X^2 - \gamma$

Output: $c_0, c_1 \in \mathbb{Z}_q$ // the coefficients of the product of the two polynomials

1 : $c_0 := a_0 \cdot b_0 + a_1 \cdot b_1 \cdot \gamma$ // steps 1-2 done modulo q

2 : $c_1 := a_0 \cdot b_1 + a_1 \cdot b_0$

3 : **return** (c_0, c_1)

Figure 6: Algorithms MultiplyNTTs and BaseCaseMultiply.

$$(x_0 + \cdots + x_{\eta-1}) - (y_0 + \cdots + y_{\eta-1}) \mod q \tag{5}$$

is computed. This is done for each of the 256 coefficients in order to obtain the polynomial, as shown in SamplePolyCBD in Figure 7. In Encrypt, the uniformly random bits must be generated in a deterministic way to ensure that the encryption of a fixed message always returns the same ciphertext. Therefore the SHAKE128 algorithm is used to generate random bits. Encrypt is used in both Encaps and Decaps, so the total time spent for the generation of binomial samples is considerable.

Primitives using the Keccak permutation. Several cryptographic hash and extendible output functions are used in the ML-KEM description. In Section 3.2.1 they are called G, J, H, $\mathsf{H}_{\mathbf{A}}$, $\mathsf{H}_{\mathbf{s},\mathbf{e}}$, and $\mathsf{H}_{\mathbf{y},\mathbf{e}_1,e_2}$. They are all instantiated with primitives based on the Keccak permutation. SHA3-512 is used for function G, SHA3-256 is used for functions H and J, SHAKE256 is used as PRF for $\mathsf{H}_{\mathbf{s},\mathbf{e}}$ and $\mathsf{H}_{\mathbf{y},\mathbf{e}_1,e_2}$, and SHAKE128 is used as XOF for $\mathsf{H}_{\mathbf{A}}$.

Given the total number of Keccak permutations in ML-KEM, the computation of this permutation constitutes the other main performance bottleneck.

6.1.2 Building K-PKE and ML-KEM

Putting the building blocks together, the K-PKE is described in details in Figure 8. The following paragraphs specify how the building block functions from the previous paragraphs are used to implement the K-PKE scheme.

Key generation. The KeyGen takes as input a random seed which is used to derive seeds for the sampling of the secret key part $\mathbf{A}$ and the secret parts $\mathbf{s}$ and $\mathbf{e}$ respectively. The pseudorandom

SampleNTT(B)

Input: byte array $B \in \mathbb{B}^{32}$ // a 32-byte seed along with two indices

Output: array $\hat{a} \in \mathbb{Z}_q^{256}$ // the coefficients of the NTT of a polynomial

1 : ctx := XOF.Init()
2 : ctx ← XOF.Absorb(ctx, B) // input the given byte array into XOF
3 : $j := 0$
4 : **while** $(j < 256)$ **do**
5 : $(\text{ctx}, C) \leftarrow \text{XOF.Squeeze}(\text{ctx}, 3)$ // get a fresh 3-byte array C from XOF
6 : $d_1 \leftarrow C[0] + 256 \cdot (C[1] \mod 16)$ // $0 \leqslant d_1 < 2^{12}$
7 : $d_2 \leftarrow \lfloor C[1]/16 \rfloor + 16 \cdot C[2]$ // $0 \leqslant d_2 < 2^{12}$
8 : **if** $(d_1 < q)$ **then**
9 : $\hat{a}[j] \leftarrow d_1$ // $\hat{a} \in \mathbb{Z}_q^{256}$
10 : $j \leftarrow j + 1$
11 : **if** $(d_2 < q) \wedge (j < 256)$ **then**
12 : $\hat{a}[j] \leftarrow d_2$
13 : $j \leftarrow j + 1$
14 : **return** $\hat{a}$

SamplePolyCBD(B)

Input: byte array $B \in \mathbb{B}^{64\eta}$

Output: array $f \in \mathbb{Z}^{256}$ // the coefficients of the sampled polynomial

1 : $b := \text{BytesToBits}(B)$
2 : **for** $(i \leftarrow 0; i < 256; i\text{++})$ **do**
3 : $x \leftarrow \sum_{j=0}^{\eta-1} b[2i\eta + j]$ // $0 \leqslant x \leqslant \eta$
4 : $y \leftarrow \sum_{j=0}^{\eta-1} b[2i\eta + \eta + j]$ // $0 \leqslant y \leqslant \eta$
5 : $f[i] \leftarrow x - y \mod q$ // $0 \leqslant f[i] \leqslant \eta$ or $q - \eta \leqslant f[i] \leqslant q - 1$
6 : **return** f

Figure 7: Algorithm SampleNTT and SamplePolyCBD.

coefficients for matrix $\mathbf{A}$ are directly sampled in the NTT domain as shown in SampleNTT in Figure 7. The binomial distributed secret vectors are sampled using SamplePolyCBD from Figure 7. The input randomness required for binomial sampling is generated by the PRF. The pseudorandom matrix and the secret vectors are used to compute k MLWE samples in a vector $\mathbf{t}$. These MLWE samples constitute the public key, while the secret vector $\mathbf{s}$ is kept as secret key. Arithmetic computations are sped up using the NTT. The polynomial parts of the keys can be stored in the NTT domain, such that they can be multiplied directly when they are used during Encrypt or Decrypt. The

K-PKE.KeyGen$(;d)$

Input: randomness $d \in \mathbb{B}^{32}$

Output: encryption key $\mathsf{ek_{PKE}} \in \mathbb{B}^{384k+32}$

Output: decryption key $\mathsf{dk_{PKE}} \in \mathbb{B}^{384k}$

$(\rho, \sigma) := \mathsf{G}(d\|k)$

// expand 32+1 bytes to two pseudorandom 32-byte seeds

$N := 0$

for $(i \leftarrow 0; i < k; i\text{++})$ **do** // generate $\hat{\mathbf{A}} \in (\mathbb{Z}_q^{256})^{k \times k}$

  for $(j \leftarrow 0; i < k; j\text{++})$ **do**

    $\hat{\mathbf{A}}[i, j] := \mathsf{SampleNTT}(\rho\|j\|i)$

for $(i \leftarrow 0; i < k; i\text{++})$ **do** // generate $\mathbf{s} \in (\mathbb{Z}_q^{256})^k$

  $\mathbf{s}[i] := \mathsf{SamplePolyCBD}_{\eta_1}(\mathsf{PRF}_{\eta_1}(\sigma, N))$

  $N \leftarrow N + 1$

for $(i \leftarrow 0; i < k; i\text{++})$ **do** // generate $\mathbf{e} \in (\mathbb{Z}_q^{256})^k$

  $\mathbf{e}[i] := \mathsf{SamplePolyCBD}_{\eta_1}(\mathsf{PRF}_{\eta_1}(\sigma, N))$

  $N \leftarrow N + 1$

$\hat{\mathbf{s}} := \mathsf{NTT}(\mathbf{s})$

$\hat{\mathbf{e}} := \mathsf{NTT}(\mathbf{e})$

$\hat{\mathbf{t}} := \hat{\mathbf{A}} \circ \hat{\mathbf{s}} + \hat{\mathbf{e}}$

$\mathsf{ek_{PKE}} := (\mathsf{ByteEncode}_{12}(\hat{\mathbf{t}})\|\rho)$ // append $\hat{\mathbf{A}}$ seed

$\mathsf{dk_{PKE}} := \mathsf{ByteEncode}_{12}(\hat{\mathbf{s}})$

return $(\mathsf{ek_{PKE}}, \mathsf{dk_{PKE}})$

K-PKE.Decrypt$(\mathsf{dk_{PKE}}, \mathsf{ct})$

Input: decryption key $\mathsf{dk_{PKE}} \in \mathbb{B}^{384k}$

Input: ciphertext $\mathsf{ct} \in \mathbb{B}^{32(d_u k + d_v)}$

Output: message $m \in \mathbb{B}^{32}$

$\mathsf{ct}_1 := \mathsf{ct}[0 : 32d_u k]$

$\mathbf{u}' := \mathsf{Decompress}_{d_u}(\mathsf{ByteDecode}_{d_u}(\mathsf{ct}_1))$

$v' := \mathsf{Decompress}_{d_v}(\mathsf{ByteDecode}_{d_v}(\mathsf{ct}_2))$

$\hat{\mathbf{s}} := \mathsf{ByteDecode}_{12}(\mathsf{dk_{PKE}})$

$w := v' - \mathsf{NTT}^{-1}(\hat{\mathbf{s}}^\top \circ \mathsf{NTT}(\mathbf{u}'))$

$m := \mathsf{ByteEncode}_1(\mathsf{Compress}_1(w))$

// decode plaintext m from polynomial w

return m

K-PKE.Encrypt$(\mathsf{ek_{PKE}}, m; r)$

Input: encryption key $\mathsf{ek_{PKE}} \in \mathbb{B}^{384k+32}$

Input: message $m \in \mathbb{B}^{32}$

Input: randomness $r \in \mathbb{B}^{32}$

Output: ciphertext $\mathsf{ct} \in \mathbb{B}^{32(d_u k + d_v)}$

$N := 0$

$\hat{\mathbf{t}} := \mathsf{ByteDecode}_{12}(\mathsf{ek_{PKE}}[0 : 384k])$

$\rho := \mathsf{ek_{PKE}}[384k : 384k + 32]$

// extract 32-byte seed from $\mathsf{ek_{PKE}}$

for $(i \leftarrow 0; i < k; i\text{++})$ **do** // regenerate $\hat{\mathbf{A}} \in (\mathbb{Z}_q^{256})^{k \times k}$

  for $(j \leftarrow 0; i < k; j\text{++})$ **do**

    $\hat{\mathbf{A}}[i, j] \xleftarrow{\$} \mathsf{SampleNTT}(\rho\|j\|i)$

for $(i \leftarrow 0; i < k; i\text{++})$ **do** // generate $\mathbf{y} \in (\mathbb{Z}_q^{256})^k$

  $\mathbf{y}[i] \xleftarrow{\$} \mathsf{SamplePolyCBD}_{\eta_1}(\mathsf{PRF}_{\eta_1}(r, N))$

  $N \leftarrow N + 1$

for $(i \leftarrow 0; i < k; i\text{++})$ **do** // generate $\mathbf{e}_1 \in (\mathbb{Z}_q^{256})^k$

  $\mathbf{e}_1[i] \xleftarrow{\$} \mathsf{SamplePolyCBD}_{\eta_2}(\mathsf{PRF}_{\eta_2}(r, N))$

  $N \leftarrow N + 1$

$e_2 \xleftarrow{\$} \mathsf{SamplePolyCBD}_{\eta_2}(\mathsf{PRF}_{\eta_2}(r, N))$

$\hat{\mathbf{y}} := \mathsf{NTT}(\mathbf{y})$

$\mathbf{u} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}}^\top \circ \hat{\mathbf{y}}) + \mathbf{e}_1$

$\mu := \mathsf{Decompress}_1(\mathsf{ByteDecode}_1(m))$

$v := \mathsf{NTT}^{-1}(\hat{\mathbf{t}}^\top \circ \hat{\mathbf{y}}) + e_2 + \mu$

// encode plaintext m into polynomial v

$\mathsf{ct}_1 := \mathsf{ByteEncode}_{d_u}(\mathsf{Compress}_{d_u}(\mathbf{u}))$

$\mathsf{ct}_2 := \mathsf{ByteEncode}_{d_v}(\mathsf{Compress}_{d_v}(v))$

return $\mathsf{ct} := (\mathsf{ct}_1\|\mathsf{ct}_2)$

Figure 8: Specification of algorithms K-PKE.KeyGen, K-PKE.Encrypt, and K-PKE.Decrypt. See Figure 2 for the high level pseudocode.

pseudorandom matrix is not stored as a key, instead the seed that allows to re-compute this matrix is stored.

Encryption. In Encrypt the function SampleNTT is used to expand the public key seed, and SamplePolyCBD to sample from the binomial distribution. Similarly to KeyGen, k MLWE samples are computed for the pseudorandom matrix, using polynomial arithmetic in the NTT domain. A separate MLWE sample is computed for part $\mathbf{t}$ of the public key. The random 32-byte message is encoded using BytesToBits and Decompress as described in the paragraphs above. The encoded message is then added to the MLWE sample. All MLWE samples are then compressed using Compress in order to reduce the size, before being returned as ciphertext.

Decryption. Since the ciphertext was compressed during Encrypt, the first step of Decrypt is decompressing the two ciphertexts parts using Decompress. They are in the normal domain, so the polynomials of ciphertext part $\mathbf{u}'$ must be mapped to the NTT domain before multiplying them with the secret key polynomials which are already in the NTT domain. The product is mapped back to the time domain using the inverse NTT, before subtracting the result from ciphertext part v'. As described in Section 3.1, the result is a polynomial whose coefficients are close to 0 or close to $\frac{q}{2}$. The $\mathsf{Compress}_d$ function with $d = 1$ is applied to obtain a bit vector, which is then encoded to bytes using ByteEncode from Figure 4.

ML-KEM. The functions described above, including Encrypt, Decrypt, KeyGen, G, H and J, together define the Key Generation, Encaps and Decaps from the IND-CCA scheme. Figure 9 shows in detail how these functions are used. The design principle of ML-KEM was discussed in Section 3.1.

6.2 Side Channel Attacks on ML-KEM

6.2.1 Introduction

Side Channel Attacks (SCA) exploit leakage of information through *side channels* during the execution of a cryptographic algorithm on a device in order to recover secret information. Any physical observable can be used as side channel, and the most widely used side channels are the computation time, power consumption and electromagnetic emanation (EM) of the device that performs the cryptographic computations. The background provided here uses power analysis as example, but EM measurements can be used in an equivalent manner.

Simple power analysis. Power analysis on the RSA signature generation scheme is a well known example of a side channel attack. The exponentiation in RSA is computed using the square-and-multiply algorithm, where squaring operations and multiplications are computed in an order depending on the bits of the secret exponent. In unprotected devices, the power consumption of the device observed during a squaring operation might be slightly different from the power consumption during a multiplication. Therefore, by measuring and analyzing the power consumption of the device during the exponentiation, a SCA attacker may recover the operations sequence of squarings and multiplications. This sequence is uniquely determined by the value of the secret exponent, which can thus be recovered by a SCA attacker. This attack is an example of *Simple Power Analysis* (SPA), where the power consumption is measured during one single execution of the cryptographic algorithm. The power *trace* (measurement) is analyzed in order to recover the secret key directly.

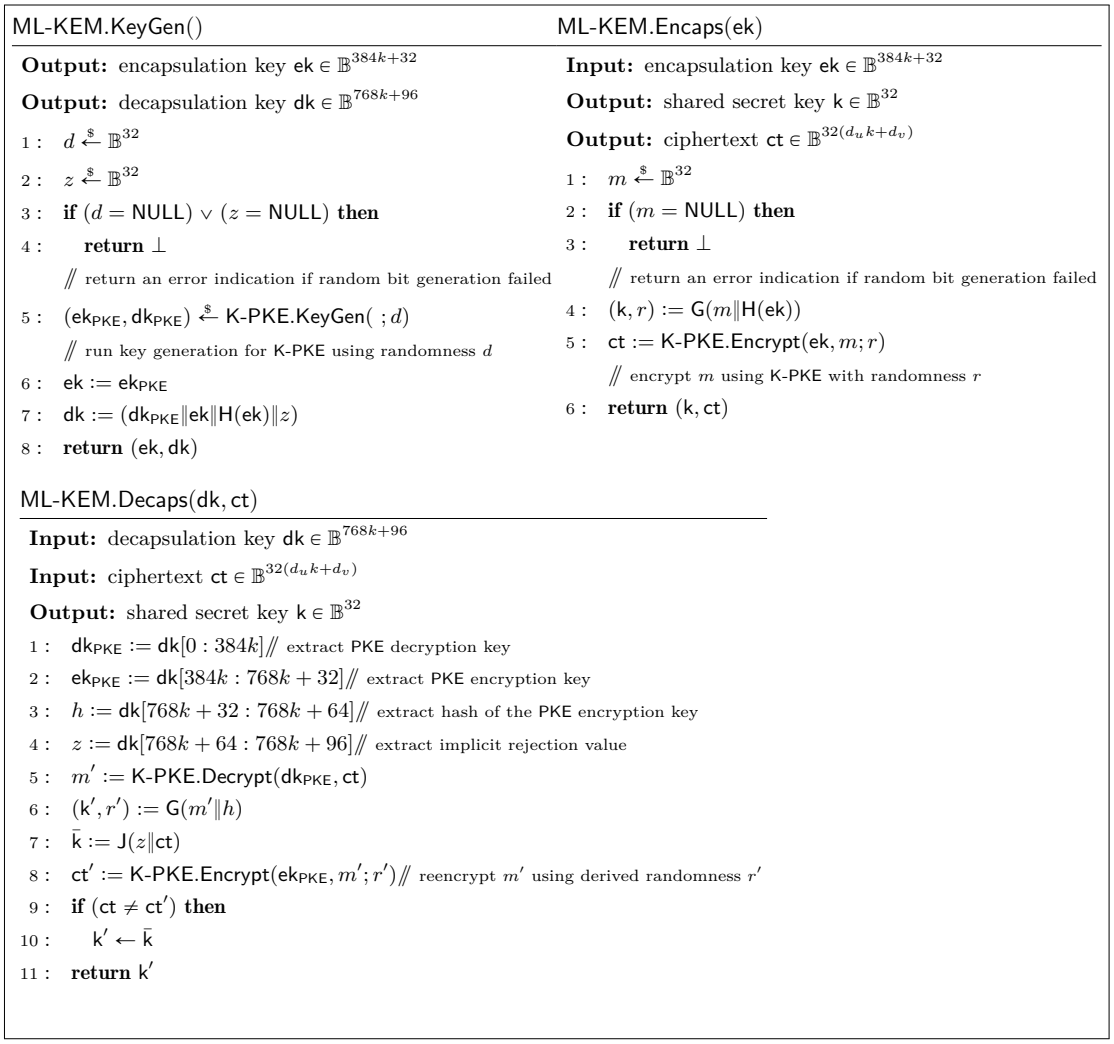


Figure 9: Specification of algorithms ML-KEM.KeyGen, ML-KEM.Encaps, and ML-KEM.Decaps running K-PKEas a sub-routine. See Figure 3 for the high level pseudocode.

Differential power analysis. Depending on the target device, the information leakage from a single power measurement might not be sufficient for full key recovery. In the Hamming weight model, it is assumed that an attacker can obtain (an approximation of) the Hamming weight of the bytes (or words) that are processed by the device. A power trace of the device computing an AES [Aes] may for example leak the Hamming weights of the bytes of the key. This is not sufficient to directly recover the value of the secret key.

In this case the attacker may try a divide-and-conquer strategy using multiple power measurements and targeting the key bytes one by one. Suppose that the attacker can trigger encryptions

for known plaintext on the target device which uses a fixed unknown key. By combining knowledge of the plaintext and the recovered Hamming weight of the intermediate state bytes during the first round, information about the key bytes can be obtained. The Hamming weight of the intermediate state bytes right after the AES' SubBytes operation (see [Aes]), which computes the S-box permutation on the bytes of the state of the first round provides valuable information. For each of the 256 possible guesses of the first byte of the first round key, the first byte of the intermediate state can be computed by XOR'ing the plaintext byte with the key guess and applying the S-Box permutation. Only a subset of those 256 possible intermediate state bytes will have the same Hamming weight as was observed by analyzing the power trace. Those key guesses that result in intermediate state bytes with a different Hamming weight are eliminated, so that the number of possible guesses for the first byte of the secret key is reduced. Repeating the same procedure for a different input plaintext will, with high probability, even further reduce the number of remaining possibilities for the first key byte. After analyzing a certain number of traces, only one single guess will remain for the first key byte. The same can be done to recover all other 15 byte positions so that the first round key is completely recovered. This divide-and-conquer strategy in which the key bytes are targeted separately by exploiting new information from each trace is referred to as *Differential Power Analysis* (DPA).

In practice, the power measurements may be noisy such that only a rough approximation of the Hamming weight of the processed bytes can be obtained from trace analysis. In that case, the attacker will have to increase the number of traces to be measured. For each trace and for each of the 256 possible key byte guesses, the Hamming weight of the intermediate state bytes is computed in the same way as described above. In *Correlation Power Analysis* (CPA), the attacker computes the correlation between the power measurements and the computed intermediate state bytes for each of the 256 key byte guesses. If a sufficient amount of traces is used (or if the noise level is sufficiently low), the highest correlation is obtained for the correct key guess.

Template attacks. In some cases, an attacker might be able to acquire a device similar to the target device. On this *clone* device, the attacker has full control over the keys, and has therefore knowledge of all the intermediate values during the computations on the clone device. This ability can be used to create a model of the clone device, which consists of templates that describe the power consumption as a function of the intermediate values during computation. If the power consumption of the clone device is sufficiently similar to the power consumption of the target device, then the power model can be used to predict the values of intermediates given power traces of the target device. Templates derived from a clone device can be used to enhance SPA, DPA or CPA attacks.

6.2.2 Side Channel Attacks on Decapsulation

During the Decrypt part of Decaps, part $\mathbf{s} = (\mathbf{s}_0, \ldots, \mathbf{s}_{k-1})$ of the decapsulation key dk is multiplied by part $\mathbf{u} = (\mathbf{u}_0, \ldots, \mathbf{u}_{k-1})$ of the input ciphertext. This section will focus on the first polynomial multiplication between $s := \mathbf{s}_0$ and $u := \mathbf{u}_0$ only, as the others are treated the same way and can therefore be attacked using the same methods.

Leakage of intermediate products during the multiplication can be exploited by a CPA attacker in order to recover s. The multiplication is computed in the NTT domain, such that the polynomial multiplication $s \cdot u$ consists of 128 multiplications of degree 1 polynomials, as can be seen in the MultiplyNTTs description in Figure 6.

Multiplications between degree 1 polynomials $u_0 + u_1 \cdot x$ and $s_0 + s_1 \cdot x$ are computed using the BaseCaseMultiply algorithm from Figure 6 and include the computation of the product $u_0 \cdot s_0 \mod q$. If the target device leaks the Hamming weight of $u_0 \cdot s_0 \mod q$ through side channels during this computation, then a DPA attacker can exploit this to recover s_0 by following the steps of this subroutine:

1. Generate a list of N_{traces} random ciphertexts and save the u_0 of each ciphertext in a list $U := u_0^{(0)}, \ldots, u_0^{(N_{\text{traces}}-1)}$.
2. Execute the decapsulation on the target device for each of the ciphertexts in the list and measure the power traces $T^{(0)}, \ldots, T^{(N_{\text{traces}}-1)}$, where each trace $T^{(i)}$ consists of N_{samples} measured samples: $T^{(i)} = T_0^{(i)}, \ldots, T_{N_{\text{samples}}-1}^{(i)}$.
3. For each candidate $\hat{s} \in \mathbb{Z}_q$ compute the *prediction vector* $P_{\hat{s}}$, which is computed as the Hamming weight $\text{HW}(\cdot)$ of the products:

$$P_{\hat{s}} = \text{HW}(U \cdot \hat{s}) = \text{HW}(u_0^{(0)} \cdot \hat{s} \mod q), \ldots, \text{HW}(u_0^{(N_{\text{traces}}-1)} \cdot \hat{s} \mod q) \tag{6}$$

4. If the traces leak the Hamming weight of the $u_0 \cdot s_0 \mod q$, then P_{s_0} must have high correlation with the vector $T_i^{(0)}, \ldots, T_i^{(N_{\text{traces}}-1)}$ for some trace sample i and the correct key guess s_0. Therefore the attacker computes the correlation $\rho_{\hat{s},j}$ between $P_{\hat{s}}$ and $T_j^{(0)}, \ldots, T_j^{(N_{\text{traces}}-1)}$ for each candidate $\hat{s} \in \mathbb{Z}_q$ and each trace sample $j = 0, 1, \ldots, N_{\text{samples}} - 1$.
5. The $\hat{s}$ for which $\rho_{\hat{s},j}$ is maximum is the correct key guess.

If the device leaks the Hamming weight of $u_i \cdot s_i \mod q$ for all $0 \leqslant i < 256$, then the complete secret key polynomial s can be recovered by repeating the process of each coefficient. The subroutines that recover the 256 coefficients are independent from one another and can therefore be performed in parallel.

Related work. The first CPA on lattice-based schemes was presented by [Rep+15]. The same framework was adapted by [Muj+24] for various polynomial multiplication algorithms, and also applied to the first version of KYBER, a predecessor of ML-KEM. The use of an incomplete NTT in ML-KEM makes the CPA slightly more complicated [Alp+24] because coefficients at odd indices are treated differently from those at even indices.

6.2.3 Side Channel Assisted Chosen Ciphertext Attacks

The re-encryption and ciphertext comparison steps in lines 5 and 6 of the Decaps algorithm in Figure 3 make sure that the decapsulation output does not reveal any information about the decrypted message m in line 6 of the Decrypt algorithm in Figure 2. This is necessary because a chosen ciphertext attacker may craft specific ciphertexts for which the bits of the decrypted message m reveal information about the secret key. However, by exploiting side channel leakage during the decapsulation, it may still be possible to perform chosen ciphertext attacks. The following paragraphs discuss the various side channel assisted chosen ciphertext attacks that are applicable to ML-KEM.

Implementing a plaintext-checking oracle with SCA. A Plaintext-Checking (PC) oracle tells whether the decrypted message is equal to the zero message $m = 000\ldots00$ or to the non-zero message $m = 100\ldots00$. In order to use a PC oracle for secret key recovery, specific input ciphertexts must be crafted. Let CT be the ciphertext for which $\mathbf{u}'_0 = k_u$, $v' = k_v$ for some constants $k_u, k_v \in \mathbb{Z}_q$, and $\mathbf{u}'_i = 0$ for $1 \leqslant i < k$. Then line 5 in Decrypt computes

$$\begin{aligned} w &= v' - \mathbf{s}^\top \cdot \mathbf{u}' \\ &= k_v - k_u \cdot \mathbf{s}_0. \end{aligned}$$

Writing $s := \mathbf{s}_0$, the first coefficient of w is equal to $w_0 = k_v - k_u \cdot s_0$ while the other coefficients $w_i = -k_u \cdot s_i$ for $i = 1, \ldots, 255$.

The next step in Decrypt is $\mathsf{Compress}_1$ which maps elements from $\mathbb{Z}_q$ to 0 or 1 depending on whether they are closer to $0 \mod q$ or to $\frac{q}{2}$. Since all the coefficients of s are within the small interval of $\{-\eta, \ldots, \eta\}$, it is possible to choose constants k_u and k_v such that Decrypt computes $\mathsf{Compress}_1(w_i) = 0$ for all $i > 0$, and $\mathsf{Compress}_1(w_0)$ is equal to 0 or 1 depending on the value of s_0 only. For instance, choosing k_v close to $\frac{q}{4}$ and k_u a positive constant slightly smaller than $\frac{q}{4\eta}$, then for $i > 0$, $|w_i| = |k_u \cdot s_i| < \frac{q}{4}$ for any $s_i \in \{-\eta, \ldots, \eta\}$, such that it compresses to 0 independently of s_i. For the first coefficient, if $s_0 > 0$ then it holds that $w_0 = k_v - k_u \cdot s_0 > \frac{q}{4}$ such that it compresses to 1, and if $s_0 < 0$ then $w_0 < \frac{q}{4}$ such that it compresses to 0. Therefore knowing whether $m = 000\ldots00$ or $m = 100\ldots00$ can enable an attacker to recover 1 bit of information about s_0.

The whole coefficient s_0 can be recovered by using the PC oracle for various different pairs of constants k_u, k_v. Other coefficients of s can be targeted by exploiting the cyclic nature of multiplication by x in R_q and repeating the same procedure with $\mathbf{u}'_0 = k_u \cdot x^i$ for $i = 1, \ldots, 255$.

SCA can be used in order to instantiate a PC oracle. As can be seen in line 5 of Decaps in Figure 3, the inputs to Encrypt depend only on the public key and the decrypted message m'. When using a constant public key, the ciphertext computed by Encrypt is completely determined by m'. If $m = 000\ldots00$ then the output of Encrypt and almost all of the intermediate values computed during Encrypt are completely different from those that are computed in the case where $m = 100\ldots00$. In other words, many intermediate values computed by Encrypt depend on a single bit of information, which is the same bit that a PC oracle should return. The PC oracle can be instantiated in the following way:

1. Craft two ciphertexts CT_0 and CT_1, where CT_0 decrypts to $000\ldots00$ and CT_1 decrypts to $100\ldots00$. This can be done using the Encrypt method, and knowledge of the secret key used during decryption is not required as long as the correct public key is used.

2. Perform a number of decapsulations using inputs CT_0 and CT_1, and record the power traces during the computation, such that two sets S_0 and S_1 of power traces are obtained. Each set S_i for $i \in \{0, 1\}$ contains power traces of decapsulations of CT_i only.

3. Use S_i to create models M_i for $i \in \{0, 1\}$ that describe the power consumption of the device during decapsulation of CT_i. One method could be for instance to let M_i be the average power trace in set S_i. The models can be used as a PC oracle by taking as input a power trace and using models M_i to check whether the power trace is more likely to correspond to a decapsulation of $000\ldots00$ or $100\ldots00$.

During the attack phase of a PC oracle SCA, a ciphertext of a special form is crafted using constants k_u and k_v as described earlier. A power trace is measured during the decapsulation of the ciphertext, and given to the PC oracle which returns one bit of information about the secret key. This process is repeated until all coefficients of the secret key are recovered.

The re-encryption is not the only part of Decaps that can be targeted by PC oracle SCA. Lines 3 and 6 of Decaps in Figure 3 are also uniquely determined by the decrypted message. Therefore, all the intermediates during the computation of hash function G depend on the same secret key dependent bit of information. The re-computed ciphertext is compared to the input ciphertext in line 6, which means that the comparison operation is also a potential target.

Related work. One of the first PC oracle attacks on ML-KEM was presented by [Rav+20b], in which they show that the attack is generic and applies to many lattice-based KEMs that use the F-O transform. It was shown by [Uen+22] that a similar SCA framework is also applicable to several code-based KEMs.

Even if the leakage is insufficient to instantiate a perfect PC oracle, [She+23] showed that it is still possible to recover the ML-KEM secret key using probabilistic PC oracles. One way to use a PC oracle that is correct with some probability < 1 is to repeatedly query it by using multiple power traces.

The work by [Raj+23] reduces the number of traces required for key recovery by attacking multiple secret key coefficients simultaneously. Instead of a binary distinguisher, they use leakage throughout the decapsulation to create a multiple-bit distinguisher.

Implementing a full decryption oracle with SCA. A Full Decryption (FD) oracle returns all the bits of the decrypted message that is computed during Decrypt. A successful a message recovery attack allows to re-compute the shared secret by concatenating it with the public key hash digest h and computing the hash function G, as shown in line 3 in Decaps. Moreover, message recovery can also be used to recover the secret key. Given a specifically crafted input ciphertext similar to those used in PC oracle attacks, the bits of the decrypted message contain information about the secret key. While in PC oracle attacks both ciphertext parts $\mathbf{u}_0$ and v are set to constants (degree 0 polynomials), in FD oracle attacks the input ciphertext part v is set to $k_v \sum_{i=0}^{n-1} x^i$. All 256 coefficients of intermediate $w = v' - \mathbf{s}^\top \cdot \mathbf{u}'$ of the decrypted ciphertext then have the same behaviour as decribed in the previous section for the first coefficient in the context of PC oracle attacks. Each coefficient is compressed to either 0 or 1 depending on the value of the corresponding secret key coefficient. Therefore, for all 256 bits of the decrypted message, the i-th bit reveals one bit of information about the i-th secret key coefficient.

Only the operations that process all bits of the decrypted message can be targeted by FD oracle SCA. These operations are ByteEncode and Compress in line 6 of Decrypt in Figure 9 and ByteDecode and Decompress in line 16 of Encrypt during the re-encryption in Decaps.

Related work. In [Xu+22] a Kyber-512 secret key is recovered using only 8 power traces, by recovering the complete decrypted message for each trace.

Subsequent works have focused on the same target operations in implementations that use countermeasures to protect against SCA. A method to defeat the shuffling countermeasure was presented by [Rav+22]. An implementation that used both shuffling and masking was shown to be still vulnerable against FD oracle attacks by [Bac+23; Jen+23]. An attack on the same target operation protected by higher order masking was presented by [Dub+23].

Implementing a decryption failure oracle with SCA. A decryption failure occurs if during the decapsulation of some ciphertext, the decrypted m' in line 2 of Figure 3 is different from the m that was encrypted in line 3 of Encaps when the ciphertext was generated. In case of Decryption Failure, the ciphertext that is re-computed during Decaps will be different from the input ciphertext. Then Decaps returns a bogus shared secret that is different from the one obtained during Encaps.

Decryption Failure (DF) oracles tell whether or not a decryption failure occurred during Decaps in line 2 of Figure 3.

The occurance of a decryption failure for a valid ciphertext can leak information about the secret key. The decryption computes

$$\begin{aligned} w &= v' - \mathbf{s}^\top \cdot \mathbf{u}' + \delta_{\text{compress}} \\ &= \mathbf{t}^\top \cdot \mathbf{y} + e_2 + \mu - \mathbf{s}^\top \cdot (\mathbf{A}^\top \cdot \mathbf{y} + \mathbf{e}_1) + \delta_{\text{compress}} \\ &= (\mathbf{A} \cdot \mathbf{s} + \mathbf{e})^\top \cdot \mathbf{y} + e_2 + \mu - \mathbf{s}^\top \cdot (\mathbf{A}^\top \cdot \mathbf{y} + \mathbf{e}_1) + \delta_{\text{compress}} \\ &= \mathbf{e}^\top \cdot \mathbf{y} + e_2 + \mu - \mathbf{s}^\top \cdot \mathbf{e}_1 + \delta_{\text{compress}}, \end{aligned}$$

where δ_{compress} denotes the small errors introduced by the compression and decompression, which can be computed by the attacker who created the ciphertext using the encapsulation algorithm. The error terms $\mathbf{e}_1$, $\mathbf{y}$ and e_2, and the message μ are also part of the encapsulation and therefore known to the attacker. If the decryption succeeds, then the sum of all the error terms must be smaller than $\frac{q}{4}$ in absolute value, such that $\mathsf{Compress}_1(w)$ computes exactly the same message bits as those used during encapsulation. If, however, the decryption fails, then that means that the absolute value of the sum of all error terms exceeds $\frac{q}{4}$ in at least one coefficient index i. The attacker obtains a linear inequality for unknown variables $\mathbf{s}$ and $\mathbf{e}$:

$$|(\mathbf{e}^\top \cdot \mathbf{y})_i + e_{2,i} - (\mathbf{s}^\top \cdot \mathbf{e}_1)_i - \delta_{\text{compress},i}| > \frac{q}{4} \tag{7}$$

This provides some information about secret key parts $\mathbf{s}$ and $\mathbf{e}$. Given many inequalities of this form, a system of inequalities is obtained. This system may be solved for $\mathbf{s}$ and $\mathbf{e}$, so that the secret key can be obtained.

Since the probability of such a decryption failure occurring for a valid ciphertext is negligible, the attacker may add $\frac{q}{4}$ to one of the coefficients of v', such that a decryption failure is triggered with a non-negligible probability.

Related work. The impact of decryption failures on the security of lattice-based crypto schemes was studied by [DVV18]. Decryption failures occur with negligible probability for valid ciphertexts, but this can be improved by crafting (invalid) chosen ciphertexts and using SCA leakage to detect decryption failures. It was shown by [GJN20] that side channel leakage could be used to instantiate a DF oracle for lattice-based schemes. Multiple works [Bha+21; Uen+22] have shown that practical implementations of ML-KEM can be successfully targeted by SCA-based DF oracle attacks. The number of traces required for key recovery in the case of imperfect SCA-based DF oracles was studied by [Her+23].

6.3 Countermeasures Against Side Channel Attacks

In order to protect against side channel attacks, countermeasures must be implemented. Algorithmic countermeasures aim to reduce or remove the statistical dependencies between processed data and

static secret keys.

Masking. *Masking* is a particularly effective technique in which each secret key dependent variable x is decomposed in random *shares* before computing the cryptographic algorithm. To protect a multiplication $x \cdot y \mod q$, where x is a secret key and y is public, first a random mask $r \xleftarrow{\$} \mathbb{Z}_q$ is generated, and the decomposition $x = x_0 + x_1 \mod q$ is computed as $x_0 = r$ and $x_1 = x - x_0 \mod q$. Note that the mask must be a fresh uniformly random value for each execution of the algorithm. The multiplication is computed on the two shares separately $x_0 \cdot y$ and $x_1 \cdot y$. All the operands during the masked multiplication are statistically independent of the secret key x. Therefore, side channel leakage of the operands cannot be used to recover information about the secret key. After computing the complete cryptographic algorithm in multiple shares, the correct output can be obtained by combining the two output shares, in this particular example by addition of the shares in $\mathbb{Z}_q$. The example uses only two shares, but the masking order can be increased to protect against higher order side channel attacks.

Depending on the type of operation to protect, either arithmetic masking (as in the example above) can be used or boolean masking. Boolean masking is suitable for protecting boolean operations. ML-KEM uses both arithmetic and boolean operations. In order to protect both types of operations, *A2B* (Arithmetic to Boolean) and *B2A* (Boolean to Arithmetic) mask conversion algorithms must be used to switch between boolean and arithmetic masking without unmasking any intermediate values. The first masked implementations of predecessors of ML-KEM were created by [Ode+18] and [Rep+15]. Masking gadgets for all operations in ML-KEM can be found in [Bos+21] and [Hei+22].

The inconvenience of masking is that the computation time of the cryptographic algorithm is multiplied by the number of shares. In practice the performance overhead is even greater due to mask conversions and other non-linear operations, which are harder to protect.

Multiplicative masking is a technique that randomizes inputs or secret keys by multiplying them with a random scalar at the start of the computation. After processing all arithmetic operations, the randomization is undone by multiplying the result with its multiplicative inverse in $\mathbb{Z}_q$. This technique is sometimes referred to as *blinding*, and was first described for ML-KEM's predecessors by [Saa18]. In [Rav+20a] the NTT in ML-KEM is protected by using blinding. The advantage of multiplicative masking is the limited computation overhead: all arithmetic operations are still only computed once, unlike for additive masking where the number of computations to be performed is doubled. A drawback of this technique is that is only protects operations that are linear over $\mathbb{Z}_q$. Many non-linear operations such as the Keccak-based primitives or the binomial sampling cannot be protected in this manner and must rely on boolean masking.

Hiding. The goal of hiding countermeasures is to dissimulate the true location of the target operation in the power trace. There are many different ways to obtain such an effect. For instance, the *shuffling* countermeasure consists of processing several independent operations in random order. The 128 calls to the BaseCaseMultiply routine for example, are all independent from one another. It does not matter if the first call processes the first degree 1 polynomials or any of the 127 others. The polynomial multiplication remains correct as long as each of the degree 1 polynomials is processed exactly once during MultiplyNTTs. By randomizing the processing order, the SCA attacker is unable to locate with certainty the exact samples of the power trace that should be targeted, thus complicating the attack. Shuffling was used in [Rav+20a] to randomize the computation order of operations

inside the NTT. The hardware implementation by [XWT25] also increased SCA resistance by using shuffling.

Other hiding countermeasures aim to degrade the quality of the power trace, by for instance causing misalignment between different power traces, or increasing the noise level of the measurements. Trace misalignment complicates SCA because the exact location of the trace samples that must be targeted varies from one trace to another. This complicates the detection of statistical dependencies between power traces and processed data.

6.4 Performance Results in Hardware

This section contains a selection of performance results reported in publications in the state of the art. Table 7 shows the cycle counts, computation time and area usage for several works. The comparison focuses on pure hardware implementations only, therefore benchmark results on CPUs are not included. NIST asked the submitters to include two benchmarks using Intel Haswell CPUs and ARM Cortex-M4 CPUs, which can be found in the round 3 KYBER documentation [Ava+21]. They are not included in the comparison of Table 7 because memory usage in a CPU is not comparable to physical chip area on FPGA. The the number of cycles on a FPGA depends on the level of parellelization, while CPUs compute instruction sequentially.

The performance numbers, both in terms of speed and area, depend strongly on the type of FPGA that is used. An implementation synthesized for higher end FPGAs such as the Virtex-7 will have lower area usage and higher speed than when synthesized for a lower end FPGA such as the Artix-7. This is because not all FPGAs implement look-up tables (LUT), flipflops (FF), BRAM (block RAM) and DSP (digital signal processors) in the same way. Even implementations for the same FPGA family are difficult to compare. In [DMG23] the XC7A200 device is used, while [XL21] used the XA7A12 variant, which are slightly different. The fastest implementation was made by [DMG23], while the most compact implementation, i.e. with the smallest area footprint, is the one by [XL21].

When adding SCA countermeasures there is a performance impact to be expected, both on speed and area usage. The two entries in the table for the work by [Kam+22] show that the impact is indeed considerable, even though they used a high end FPGA.

Table 7: Comparison of FPGA Implementations of ML-KEM (Kyber)

Reference	Parameter	Cycles (×1000) (K / E / D)	Freq (MHz)	Time (μs) (K / E / D)	Area				FPGA
					LUT (×1000)	FF (×1000)	DSP	BRAM	
[XL21]	Kyber-512 Kyber-768 Kyber-1024	3.8 / 5.1 / 6.7 6.3 / 7.9 / 10.0 9.4 / 11.3 / 13.9	161	23.4 / 31.5 / 41.4 39.2 / 49.2 / 62.4 58.3 / 70.3 / 86.4	7.4	4.6	2	3	Artix-7
[Hua+20]	Kyber-512 Kyber-768 Kyber-1024	- / 48.0 / 68.8 - / 77.5 / 102.1 - / 107.1 / 135.6	155 192	- / 366 / 444 - / 564 / 686 - / 802 / 975	88.9 110.2 132.9	152.9 167.3 172.5	354 292 548	202 202 202	Artix-7
[DMG23]	Kyber-512 Kyber-768 Kyber-1024	2.2 / 3.2 / 4.5 2.6 / 3.7 / 4.9 3.6 / 4.8 / 5.8	220	10.0 / 14.7 / 20.5 12.0 / 17.0 / 22.2 16.2 / 21.7 / 26.4	9.5 10.5 11.6	8.5 9.8 11.1	4 6 8	4.5 6.5 8.5	Artix-7
(Hiding-only) [Kam+22]	Kyber-512	- / 882 / 1266	100	- / 88.2 / 126.6	153.9	-	60	294	Virtex-7
(Hiding+Masking) [Kam+22]	Kyber-512	- / 881 / 1377	100	- / 88.1 / 137.7	163.6	-	76	489.5	Virtex-7

Note: K = KeyGen, E = Encapsulation, D = Decapsulation. Kamucheka et al. figures are for Kyber-512 on Virtex-7 (VC707).

7 Application on ML-KEM: Transport Layer Security

In this section, we will discuss the Transport Layer Security (TLS) protocol, how we can integrate ML-KEM to provide post-quantum confidentiality, and what the performance impact of this change is. We will first explain TLS, before discussing how ML-KEM is integrated into TLS and the performance impacts. We will also discuss post-quantum/traditional (PQ/T) "hybrids" of classic and post-quantum cryptography, and discuss the ongoing deployment of TLS in the web. Note that although we will briefly mention post-quantum authentication using post-quantum digital signatures, we will not go in detail. Parts of this section are based on [Wig24].

7.1 Transport Layer Security Version 1.3

The Transport Layer Security protocol, the current version of which is TLS 1.3, is defined by the Internet Engineering Task Force (IETF) standard RFC 8446 [Res18]. TLS, also known as SSL (which is the name of its original versions developed by Netscape), is well known for being the 'S' component in 'HTTPS', used in secure web browsing [Res00]. However, TLS is widely used in many contexts that require a secure channel, including secure email [New99; Hof02], file transfer [FH05], and VPN connections [Opeb].

TLS consists of two sub-protocols. For the actual encrypted transmission of application data, the *record layer* protocol uses symmetric-key authenticated encryption algorithms. Because these are quantum-secure, we will not further discuss the record layer. The keys that the record layer uses are computed in the *handshake* protocol. This is an authenticated key exchange algorithm which performs an ephemeral key exchange to compute encryption keys, and authenticates the server using digital signatures. TLS 1.3 optionally supports authentication of the client, but this is not particularly relevant for this document as we are chiefly concerned with ML-KEM, a key exchange algorithm.

A high-level overview of TLS 1.3 is shown in Figure 10. In the initial message by the client, it samples a new Elliptic Curve Diffie–Hellman (ECDH) private key x and sends the corresponding public value xG to the server. The server uses this value with its sampled ECDH private key y to compute a shared secret key ss. Traffic encryption keys are derived from ss using a Key Derivation Function (KDF). The server responds to the client's message by sending its ECDH public value yG. This allows the client to also compute ss and the traffic encryption keys. The server also sends its identity and signature public key in a *certificate*, plus a signature over the exchanged messages (i.e., the *transcript*). This signature proves that the server owns the private key corresponding to the authenticated public key in the certificate. Finally, it sends a key confirmation message, after which the server can start sending encrypted application data. The client will confirm its view on the handshake by also sending a key confirmation message, after which the handshake is completed.

7.2 TLS with Post-Quantum Confidentiality

To provide security against quantum adversaries in TLS 1.3, we can, in principle, straightforwardly replace all pre-quantum algorithms by post-quantum primitives.

We can replace the pre-quantum elliptic-curve Diffie–Hellman (DH) key exchange algorithms by ML-KEM, to provide post-quantum confidentiality. This is particularly important when considering "harvest-now-decrypt-later" attacks.

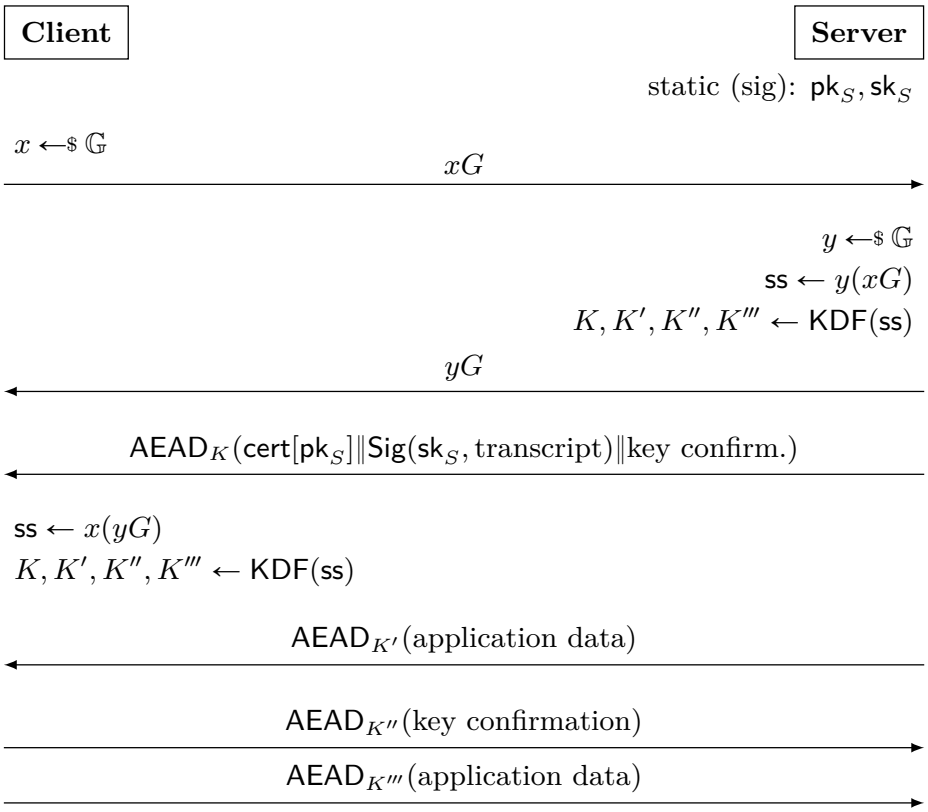


Figure 10: High-level overview of the TLS 1.3 handshake.

The way we replace DH by ML-KEM in TLS 1.3 is straightforward.[2] The client still sends an encapsulation key to the server, though now it is generated using ML-KEM's key generation functionality. The server's behavior changes a bit more: instead of generating an encapsulation key, it calls the KEM.Encapsulate function of ML-KEM and transmits the generated ciphertext.

The signature algorithms used in the handshake and certificates, which currently are based on RSA or elliptic-curve signatures, can simply be replaced by post-quantum signature algorithms, as they provide the same functionality.[3] As this document focuses on ML-KEM, a post-quantum *key exchange* algorithm, we will not further treat this issue, and in all measurements we are assuming *classical authentication* algorithms.

A high-level overview of TLS 1.3 using post-quantum ML-KEM in place of DH is shown in Figure 11.

[2]Though many uses of DH key exchange can be replaced by ML-KEM, this is not true in general. In particular, when DH is used for authentication, ML-KEM may not be suitable. There are unfortunately no practical, truly drop-in post-quantum replacements for Diffie–Hellman, so each protocol will need to be evaluated for compatibility.

[3]Practically, there are significant concerns on the transition to post-quantum authentication due to the large number of signatures involved, combined with the large sizes of post-quantum signature schemes.

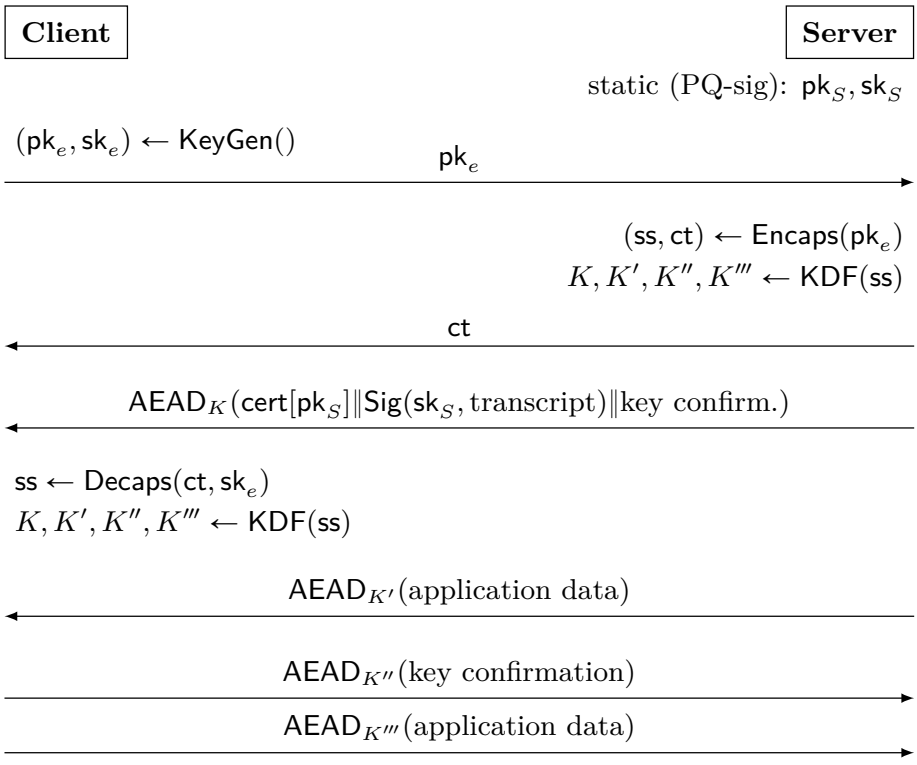


Figure 11: A high-level overview of TLS 1.3 with post-quantum ML-KEM. The ECDH ephemeral key exchange is replaced by ML-KEM operations KeyGen, Encaps, and Decaps. Note that it is possible to use post-quantum KEM without using post-quantum signatures, which provides post-quantum confidentiality against harvest-now-decrypt-later attacks. If post-quantum signatures are used in place of the classical signatures, this protocol is fully post-quantum secure against active quantum attackers.

7.3 Comparing ML-KEM to ECDH Key Exchange Algorithms

Although there are no theoretical obstacles to integrating ML-KEM into TLS 1.3, there are other considerations that affect practical use, notably overhead. In particular, almost all use cases are concerned with runtime performance and bandwidth usage.[4] In this section, we compare these characteristics for key exchange algorithms as used in TLS. This includes the currently-used classic or "traditional" algorithms and "pure" post-quantum key exchange methods (i.e., ML-KEM), but also combinations of traditional and post-quantum algorithms (also called hybrid algorithms, see Section 7.4).

[4]Some platforms may also be concerned with implementation size, but it is difficult to make general statements about the implementation size of any algorithm, including ML-KEM, without making assumptions on the target platform and technology. However, some representative benchmarks can be found at the pqm4 project [Kan+].

Table 8: Performance of cryptographic primitive operations (in milliseconds) for ECDH and ML-KEM.

Primitive	**PQ security level**	KeyGen	Encaps	Decaps
Classic key exchange algorithms				
EC X25519	1†	0.027	0.058	0.029
EC P-256	1†	0.008	0.058	0.047
EC P-384	3†	0.088	0.327	0.229
EC P-521	5†	0.098	0.341	0.226
Post-Quantum key exchange algorithms				
ML-KEM-512	1	0.020	0.014	0.023
ML-KEM-768	3	0.031	0.020	0.032
ML-KEM-1024	5	0.047	0.028	0.043
Post-Quantum/Traditional "hybrid" key exchange algorithms				
X25519 + ML-KEM-768	1† + 3	0.061	0.076	0.060
P-256 + ML-KEM-768	1† + 3	0.044	0.076	0.076
P-384 + ML-KEM-1024	3† + 5	0.143	0.344	0.256

†: (approximate) security level versus classic adversaries, no post-quantum security.
Benchmarks obtained using `openssl speed`, using OpenSSL 3.6.0 on a Macbook Pro with M2 Pro, running MacOS 26.0.

7.3.1 Computation Time

The time it takes to compute the KeyGen, Encaps, and Decaps operations is very relevant to TLS, as these operations are directly contributing to the time it takes to set up a connection. Fortunately, ML-KEM computation time is generally very good. In Table 8, we compare the performance of ML-KEM to ECDH algorithms currently used in TLS. For the ECDH algorithms, "Encapsulate" measures the generation of a new ECDH key, plus a group operation; "Decapsulate" is a single group operation. ML-KEM-512 is about as fast or much faster than all algorithms in the list, except for P-256's keygen operation. At higher security levels, ML-KEM is much faster than the NIST P-384 and P-521 elliptic curves.

It is worth pointing out, however, that we generally measure transit time on the network in (tens of) milliseconds, so all of these algorithms contribute only fractionally to the time it takes to set up a single connection: they are "fast enough". Runtime may still be relevant to applications that handle many connections at the same time, such as web servers or firewalls that do TLS connection inspection, though such applications may also want to consider offloading public-key computations from the main CPU to dedicated hardware implementations.

7.3.2 Bandwidth Usage

Another factor that determines the practicality of TLS is bandwidth usage. This both concerns the time that is needed to transmit messages, but also the direct cost of bandwidth for e.g. users on metered connections. Bandwidth is significantly affected by integrating ML-KEM. This is in

Table 9: Comparing encapsulation key and ciphertext sizes (in bytes) for ECDH and ML-KEM.

Primitive	PQ security level	Encapsulation key	Ciphertext
Classic key exchange algorithms			
EC X25519	$1^\dagger$	32	32
EC P-256	$1^\dagger$	65	65
EC P-384	$3^\dagger$	97	97
EC P-521	$5^\dagger$	123	123
Post-Quantum key exchange algorithms			
ML-KEM-512	1	800	768
ML-KEM-768	3	1184	1088
ML-KEM-1024	5	1568	1568
Post-Quantum/Traditional "hybrid" key exchange algorithms			
X25519 + ML-KEM-768	$1^\dagger + 3$	1216	1120
P-256 + ML-KEM-768	$1^\dagger + 3$	1249	1153
P-384 + ML-KEM-1024	$3^\dagger + 5$	1665	1617

†: (approximate) security level versus classic adversaries, no post-quantum security.

large part due to the fact that ML-KEM encapsulation keys and ciphertexts are much larger. In Table 9, we compare the encapsulation key and ciphertext sizes for ECDH algorithm with ML-KEM. It is clear that ML-KEM encapsulation keys and ciphertexts are a factor 12–25× larger than the ECDH equivalents. In particular, X25519 is currently the most popular key exchange algorithm on the web, and ML-KEM-512 encapsulation keys are 25× larger. In the benchmark results and in the deployment of ML-KEM on the web, both of which we will discuss below, the increase in size has proven manageable and do not hinder the usage of ML-KEM on the internet. However, in Section 7.5.1 we will discuss how this increase in sizes has uncovered some bugs in implementations.

7.3.3 Laboratory Experiments with ML-KEM in TLS 1.3

To explore the end-to-end effects of the integration of ML-KEM-based key exchange algorithms in TLS on connection setup times, we integrated ML-KEM-512, ML-KEM-768, and ML-KEM-1024, as well as the PQ/T hybrids X25519 + ML-KEM-768, P-256 + ML-KEM-768, and P-384 + ML-KEM-1024, into TLS 1.3. In particular, we have integrated the ML-KEM implementations from `liboqs` [Ope25] 0.14.0 into Rustls [BP]. The benchmark setup is based on the work by Wiggers; a more detailed description can be found in [Wig24, Ch. 10].

Note that we are only measuring the impact of changing the authentication algorithms. For all reported measurements, we have fixed the signature algorithms for the server's identity certificate, the intermediate CA certificate and the root CA certificate to RSA2048.

We report performance in two emulated network environments. In the first, the network latency is 31ms and the network bandwidth is 1000mbps. This represents a high-bandwidth, relatively low-latency connection; for example, between two servers across a continent. The second network environment sets the network latency to 195ms, and the network bandwidth to 10 mbps. This

represents a low-bandwidth with a latency representing a transatlantic connection. Experiments were run on a `m8a.24xlarge` virtual machine from Amazon AWS, using 96 cores on an AMD EPYC 9R45 CPU, running Amazon Linux 2023 with Linux 6.12. The reported numbers are the average of 24 000 runs. The benchmarking software and collected results are archived with DOI 10.5281/zenodo.17607131.

Table 10: Comparing TLS 1.3 connection setup times.

	Client handshake time	
Key exchange algorithm	1000mbps, 31.4ms latency	10mbps, 196.1ms latency
X25519	62.6 ms	394.8 ms
P-256	62.6 ms	394.9 ms
P-384	63.2 ms	396.1 ms
ML-KEM-512	62.6 ms	395.8 ms
ML-KEM-768	62.6 ms	396.5 ms
ML-KEM-1024	62.6 ms	397.2 ms
X25519 + ML-KEM-768	62.7 ms	396.6 ms
P-256 + ML-KEM-768	62.6 ms	396.7 ms
P-384 + ML-KEM-1024	63.3 ms	398.1 ms

RSA-2048 was used for all server signatures.

The results of the experiments are shown in Table 10. The client handshake times are dominated by the round-trips necessary for the handshake: one to establish the TCP connection over which the TLS protocol is run (the TCP `SYN`/`ACK`), and one to do the message exchange for TLS 1.3. Only on the low-bandwidth connection there appears to be any effect of using ML-KEM, but the incurred additional handshake latency is very minimal.

7.4 Post-Quantum/Traditional "Hybrid" Algorithms

The currently-used classic key exchange algorithms in TLS, and their software or hardware implementations, are well-understood and well-tested. Some consider replacing these implementations by "new" ML-KEM implementations risky. This is one of the reasons that many choose to use Post-Quantum/Traditional (PQ/T) KEM algorithms, also known as "hybrid" KEMs. These algorithms combine a quantum-vulnerable traditional (typically elliptic-curve) algorithm with ML-KEM. They are executed in parallel in such a way that both algorithms would need to be compromised in order to compromise the combination. This provides strong assurances against algorithm or implementation flaws, though at the cost of (a small amount) of additional bandwidth used and having to compute both key exchanges. Using a PQ/T algorithm instead of "pure", stand-alone ML-KEM also implies more code size or area (as both algorithms need to be implemented), which may be prohibitive in embedded settings. Finally, using PQ/T algorithms implies migrating twice: once to the PQ/T scheme, and once quantum computers are available or trust in ML-KEM and its implementation is sufficient, away from the hybrid to the "pure" ML-KEM. Choosing between PQ/T algorithms or "pure" ML-KEM requires balancing these concerns.

In Tables 8 and 9, we already listed the characteristics of currently popular PQ/T KEMs built from ML-KEM. It is notable that they use ML-KEM-768, which targets a 192-bit (PQ) security level,

together with X25519 and P-256, which provide 128 bits of classical security. The choice for a higher security level of ML-KEM hedges against advances in cryptanalysis of ML-KEM. The combination of P-384 and ML-KEM-1024 is a more conservative option.

7.4.1 Regulator Positions on Hybrids

In Europe, the German federal regulator BSI [ISB25], and French regulator ANSSI [ANS23] recommend or require (depending on context) the use of hybrid schemes. This position is mirrored in a European Commission recommendation to EU member states [Gro25]. In, for example, Australia, Canada, the United Kingdom, and the United States of America, national cybersecurity bodies are taking a more neutral position, allowing the use of hybrids as a stepping stone, but acknowledging that they add complexity along the way to a "fully post-quantum end-state" [Aus25; Mat25; Nat24a; Moo+24]. In a set of requirements (CNSA 2.0) for the intelligence systems overseen by the United States National Security Agency, meanwhile, hybrids are generally not permitted, as they seemingly are trying to avoid the complexity of a diversified cryptographic algorithm landscape and managing multiple migrations (in general, CNSA 2.0 only permits a very limited set of algorithms) [Nat24c].

7.5 Experiments with TLS with Post-Quantum Confidentiality

Because of many practical, large-scale experiments going back to 2016, we are able to provide an accurate view on the practicality of TLS 1.3 with ML-KEM on the public internet. The Google Chrome browser started the CECPQ1 ("combined elliptic-curve and post-quantum 1") experiment in 2016 [Lan16], combining X25519 [Ber06] with the NewHope lattice-based KEM key exchange [Alk+16] (a predecessor to ML-KEM) in the TLS 1.2 handshake. The follow-up CECPQ2 experiment based on TLS 1.3 was announced in late 2018 [Lan18; KV19], using a combination of X25519 and the lattice-based scheme NTRU-HRSS [Hül+17; Sch+17],[5] and X25519 with the isogeny-based scheme SIKE [Jao+22]. The first results from this experiment are presented in [Kwi+19].

The academic Open Quantum Safe (OQS) initiative [SM16] provides prototype integrations of post-quantum and hybrid key exchange in TLS 1.2 and TLS 1.3 to the OpenSSL library [Opea]. First results in terms of feasibility of migration and performance using OQS were presented in [CPS19]; more detailed benchmarks are presented in [PST20]. Schwabe, Stebila, and Wiggers [SSW20; SSW21], Celi et al. [Cel+21], and Wiggers [Wig24] present results using post-quantum TLS with different key exchange and authentication algorithms through experiments with Rustls [BP], though the focus of their work is on comparing post-quantum signature-based authentication in TLS 1.3 to KEMTLS, an alternative TLS protocol which uses authentication based on KEMs. On embedded platforms, experimentation with TLS was done by [GW22; BS+20; Tas+22].

Meanwhile, draft specifications for PQ/T key exchange had started the discussion on standardization of post-quantum cryptography for TLS [SWZ17; CC21; KK18; Why+17; SS17; SFG25; HW20]. This has progressed with three drafts that are almost finalized to RFC, and that are rapidly getting implemented and deployed [Con25; Kwi+25; SFG25]

Based on the results from academic, CECPQ1 and CECPQ2 experiments, Google Chrome and Cloudflare were able to start the deployment of the `X25519MLKEM768` key exchange algorithm that combines X25519 with ML-KEM-768.[6] This started with in August 2023 with a small percentage rollout on desktop [O'B23], but has now been fully enabled in Google Chrome (100% since

[5] NTRU-HRSS was merged into the NTRU [Che+20] submission which was eventually not selected.
[6] Though initially using draft standards with minor differences to the final ML-KEM-768 in FIPS 203.

April 2024 [Adr+24]), and in Chrome-derived browers like Microsoft Edge [Mic25]. Apple has enabled PQC hybrid key exchange in Safari since September 2025 [App25].

As of November 2025, Cloudflare reports that around 50% of their HTTPS traffic is using `X25519MLKEM768` [Clo]

7.5.1 Compatibility Problems Found During the Deployment of `X25519MLKEM768`

As Google was rolling out the PQ/T algorithm `X25519MLKEM768`, they found that this was causing some connections to fail. This was caused by so-called "middleboxes". These are network appliances like firewalls, typically found in enterprise networks, that may inspect or intercept TLS traffic. They were found to sometimes have implementations that were not prepared to handle initial TLS messages that span more than a single network packet. This was not encountered before, as all key exchange methods (plus protocol overhead) typically easily fit in single network packets. As using `X25519MLKEM768` pushes the initial message over the common approximately 1200 byte limit for network packets, the packet needs to be fragmented. On the website `https://tldr.fail`, Google further explains this issue and tracks the status of fixes.

What this issue indicates is that implementations may make (invalid) assumptions on what a typical TLS connection should look like. These assumptions may be stricter than or even go against protocol specifications. Making a large change to the "shape" of the connection, including adding larger-than-before key exchange methods such as ML-KEM, may invalidate those assumptions. This issue is sometimes referred to as "protocol ossification". Similar issues plagued the deployment of TLS 1.3 [Sul17].

When deploying ML-KEM, one should be mindful of protocol ossification problems.

7.6 Availability of ML-KEM Support in Popular TLS Libraries

In this section, we survey some popular TLS libraries and note the availability of ML-KEM support. Table 11 show for each library if algorithms are not supported (✗); supported, but currently disabled by default (✓); supported, and enabled by default, but not the preferred algorithm in notation (✓✓); and if an algorithm is supported, enabled, and preferred in algorithm negotiation (★). It also lists the version number in which (any) ML-KEM support first became available,[7] though the supported features are for the most recent version as of mid-November 2025.

As Table 11 shows, TLS libraries are rapidly adopting support for ML-KEM, in particular the X25519 + ML-KEM-768 PQ/T algorithm. Note that Table 11 only refers to functionality for TLS; many libraries also offer access to cryptographic primitives directly, and they may support using additional parameter sets of ML-KEM that way even if they are not supported in TLS.

Lastly, note that most libraries come with defaults, but these may be overridden by the users of the library or, e.g., configurations shipped in Linux distributions. Only checking library versions is not sufficient to verify if an application uses ML-KEM and provides post-quantum confidentiality.

7.7 Discussion

The results discussed in this section show that ML-KEM is suitable for use in replacing elliptic-curve based key exchange in TLS for "normal" web browsing settings. Even when using "hybrid" PQ/T algorithms in which the cost of ML-KEM is added "on top" of the current connection overhead

[7] Though we do not consider support for draft versions of ML-KEM.

Table 11: Support for ML-KEM and ML-KEM-based PQ/T algorithms in popular TLS libraries.

		First support		Support in TLS					
Library	**Website**	**Version**	**Date**	**ML-KEM-512**	**ML-KEM-768**	**ML-KEM-1024**	**X25519 + ML-KEM-768**	**P-256 + ML-KEM-768**	**P-384 + ML-KEM-1024**
OpenSSL	[Opea]	3.5.0	2025-04-08	✓	✓	✓	★	✓✓	✓✓
mbed TLS	[ARM]	–	–	✗	✗	✗	✗	✗	✗
Rustls	[BP]	0.23.22	2025-01-31	✗	✓	✗	★	✓	✗
BoringSSL	[Goo]	N/A	2024-08-27	✗	✗	✓	★	✗	✗
Bouncy Castle Java	[Theb]	1.81	2025-06-04	✓	✓	✓	✓	✓	✓
Bouncy Castle C#	[Theb]	2.6.1	2025-06-04	✓	✓	✓	✓	✓	✓
NSS	[Moz]	0.105.0	2024-09-26	✗	✗	✗	★	✓✓	✓✓
Go	[Thea]	1.24.0	2025-02-11	✗	✗	✗	★	✗	✗
Microsoft SChannel	[Mic]	–	–	✗	✗	✗	✗	✗	✗
Apple Network.framework	[App]	26.0	2025-09-15	✗	✗	✗	★	✗	✗

Support legend: ✗: Not supported; ✓: Supported, but not enabled by default; ✓✓: Enabled by default; ★: Enabled and preferred algorithm in negotiation. Version listed indicates first availability of any support; checkmarks indicate support as of writing (November 2025).
Note: applications may override the default preferences of the libraries they use.

posed by cryptography, ML-KEM adds negligible additional latency to TLS connection setup. For embedded settings, the additional bandwidth costs may be more prohibitive, see e.g., [GW22; BS+20; Tas+22]. Additionally, when integrating ML-KEM into protocols that are currently used, protocol ossification risks may exist that can hinder deployment. Such concerns will need to be addressed on a case-by-case basis. Finally, many TLS libraries are already updated for ML-KEM support, indicating broad support and availability.

References

[Abd+05] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. "Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions". In: *CRYPTO 2005*. Ed. by Victor Shoup. Vol. 3621. LNCS. Springer, Berlin, Heidelberg, Aug. 2005, pp. 205–222. DOI: 10.1007/11535218_13.

[ABD16] Martin R. Albrecht, Shi Bai, and Léo Ducas. "A Subfield Lattice Attack on Overstretched NTRU Assumptions - Cryptanalysis of Some FHE and Graded Encoding Schemes". In: *CRYPTO 2016, Part I*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9814. LNCS. Springer, Berlin, Heidelberg, Aug. 2016, pp. 153–178. DOI: 10.1007/978-3-662-53018-4_6.

[ABN10] Michel Abdalla, Mihir Bellare, and Gregory Neven. "Robust Encryption". In: *TCC 2010*. Ed. by Daniele Micciancio. Vol. 5978. LNCS. Springer, Berlin, Heidelberg, Feb. 2010, pp. 480–497. DOI: 10.1007/978-3-642-11799-2_28.

[ACW19] Martin R. Albrecht, Benjamin R. Curtis, and Thomas Wunderer. "Exploring Trade-offs in Batch Bounded Distance Decoding". In: *SAC 2019*. Ed. by Kenneth G. Paterson and Douglas Stebila. Vol. 11959. LNCS. Springer, Cham, Aug. 2019, pp. 467–491. DOI: 10.1007/978-3-030-38471-5_19.

[AD21] Martin Albrecht and Léo Ducas. *Lattice Attacks on NTRU and LWE: A History of Refinements*. Cryptology ePrint Archive, Report 2021/799. 2021. URL: https://eprint.iacr.org/2021/799.

[Adr+24] David Adrian, Bob Beck, David Benjamin, and Devon O'Brien. *Advancing Our Amazing Bet on Asymmetric Cryptography*. May 23, 2024. URL: https://blog.chromium.org/2024/05/advancing-our-amazing-bet-on-asymmetric.html.

[Aes] *Advanced Encryption Standard (AES)*. National Institute of Standards and Technology, NIST FIPS PUB 197, U.S. Department of Commerce. Nov. 2001.

[AFG14] Martin R. Albrecht, Robert Fitzpatrick, and Florian Göpfert. "On the Efficacy of Solving LWE by Reduction to Unique-SVP". In: *ICISC 13*. Ed. by Hyang-Sook Lee and Dong-Guk Han. Vol. 8565. LNCS. Springer, Cham, Nov. 2014, pp. 293–310. DOI: 10.1007/978-3-319-12160-4_18.

[AG11] Sanjeev Arora and Rong Ge. "New Algorithms for Learning in Presence of Errors". In: *ICALP 2011, Part I*. Ed. by Luca Aceto, Monika Henzinger, and Jiri Sgall. Vol. 6755. LNCS. Springer, Berlin, Heidelberg, July 2011, pp. 403–415. DOI: 10.1007/978-3-642-22006-7_34.

[Agg+20] Divesh Aggarwal, Jianwei Li, Phong Q. Nguyen, and Noah Stephens-Davidowitz. "Slide Reduction, Revisited - Filling the Gaps in SVP Approximation". In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Cham, Aug. 2020, pp. 274–295. DOI: 10.1007/978-3-030-56880-1_10.

[Alb+14] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, and Ludovic Perret. *Algebraic Algorithms for LWE*. Cryptology ePrint Archive, Report 2014/1018. 2014. URL: https://eprint.iacr.org/2014/1018.

[Alb17] Martin R. Albrecht. "On Dual Lattice Attacks Against Small-Secret LWE and Parameter Choices in HElib and SEAL". In: *EUROCRYPT 2017, Part II*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10211. LNCS. Springer, Cham, 2017, pp. 103–129. DOI: 10.1007/978-3-319-56614-6_4.

[Alb+19] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. "The General Sieve Kernel and New Records in Lattice Reduction". In: *EUROCRYPT 2019, Part II*. Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11477. LNCS. Springer, Cham, May 2019, pp. 717–746. DOI: 10.1007/978-3-030-17656-3_25.

[Alb+20a] Martin R. Albrecht, Shi Bai, Pierre-Alain Fouque, Paul Kirchner, Damien Stehlé, and Weiqiang Wen. "Faster Enumeration-Based Lattice Reduction: Root Hermite Factor $k^{1/(2k)}$ Time $k^{k/8+o(k)}$". In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Cham, Aug. 2020, pp. 186–212. DOI: 10.1007/978-3-030-56880-1_7.

[Alb+20b] Martin R. Albrecht, Vlad Gheorghiu, Eamonn W. Postlethwaite, and John M. Schanck. "Estimating Quantum Speedups for Lattice Sieves". In: *ASIACRYPT 2020, Part II*. Ed. by Shiho Moriai and Huaxiong Wang. Vol. 12492. LNCS. Springer, Cham, Dec. 2020, pp. 583–613. DOI: 10.1007/978-3-030-64834-3_20.

[Alb+21] Martin R. Albrecht, Shi Bai, Jianwei Li, and Joe Rowell. "Lattice Reduction with Approximate Enumeration Oracles - Practical Algorithms and Concrete Performance". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Cham, Aug. 2021, pp. 732–759. DOI: 10.1007/978-3-030-84245-1_25.

[Alk+16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. "Post-quantum Key Exchange - A New Hope". In: *USENIX Security 2016*. Ed. by Thorsten Holz and Stefan Savage. USENIX Association, Aug. 2016, pp. 327–343. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim.

[Alm+24] José Bacelar Almeida, Santiago Arranz Olmos, Manuel Barbosa, Gilles Barthe, François Dupressoir, Benjamin Grégoire, Vincent Laporte, Jean-Christophe Léchenet, Cameron Low, Tiago Oliveira, Hugo Pacheco, Miguel Quaresma, Peter Schwabe, and Pierre-Yves Strub. "Formally Verifying Kyber - Episode V: Machine-Checked IND-CCA Security and Correctness of ML-KEM in EasyCrypt". In: *CRYPTO 2024, Part II*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14921. LNCS. Springer, Cham, Aug. 2024, pp. 384–421. DOI: 10.1007/978-3-031-68379-4_12.

[Alp+24] Estuardo Alpirez Bock, Gustavo Banegas, Chris Brzuska, Lukasz Chmielewski, Kirthivaasan Puniamurthy, and Milan Sorf. "Breaking DPA-Protected Kyber via the Pair-Pointwise Multiplication". In: *ACNS 2024, Part II*. Ed. by Christina Pöpper and Lejla Batina. Vol. 14584. LNCS. Springer, Cham, Mar. 2024, pp. 101–130. DOI: 10.1007/978-3-031-54773-7_5.

[ANS23] ANSSI. *ANSSI views on the Post-Quantum Cryptography transition (2023 follow up)*. Dec. 21, 2023. URL: https://messervices.cyber.gouv.fr/documents-guides/follow_up_position_paper_on_post_quantum_cryptography.pdf.

[Aon+16] Yoshinori Aono, Yuntao Wang, Takuya Hayashi, and Tsuyoshi Takagi. "Improved Progressive BKZ Algorithms and Their Precise Cost Estimation by Sharp Simulator". In: *EUROCRYPT 2016, Part I*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9665. LNCS. Springer, Berlin, Heidelberg, May 2016, pp. 789–819. DOI: 10.1007/978-3-662-49890-3_30.

[App] Apple. *Network.framework*. URL: https://developer.apple.com/documentation/network/.

[App+09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Berlin, Heidelberg, Aug. 2009, pp. 595–618. DOI: 10.1007/978-3-642-03356-8_35.

[App25] Apple. *Prepare your network for quantum-secure encryption in TLS*. July 23, 2025. URL: https://support.apple.com/en-me/122756.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. "On the concrete hardness of Learning with Errors". In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203. DOI: doi:10.1515/jmc-2015-0016. URL: https://doi.org/10.1515/jmc-2015-0016.

[ARM] ARM Limited. *mbed TLS*. URL: https://tls.mbed.org/.

[Aus25] Australian Signals Directorate. *Guidelines for Cryptography*. Information Security Manual (ISM), first published 4 December 2025, last updated 4 December 2025. Dec. 2025. URL: https://www.cyber.gov.au/business-government/asds-cyber-security-frameworks/ism/cyber-security-guidelines/guidelines-for-cryptography (visited on 01/13/2026).

[Ava+21] Roberto Avanzi, Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. *CRYSTALS-Kyber: Algorithm Specifications and Supporting Documentation (version 3.02)*. Submission to NIST PQC Round 3. Version 3.02. Aug. 2021. URL: https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf.

[Bac+23] Linus Backlund, Kalle Ngo, Joel Gärtner, and Elena Dubrova. "Secret Key Recovery Attack on Masked and Shuffled Implementations of CRYSTALS-Kyber and Saber". In: *Applied Cryptography and Network Security Workshops: ACNS 2023 Satellite Workshops, ADSC, AIBlock, AIHWS, AIoTS, CIMSS, Cloud S&P, SCI, SecMT, SiMLA, Kyoto, Japan, June 19–22, 2023, Proceedings*. Kyoto, Japan: Springer-Verlag, 2023, pp. 159–177. DOI: 10.1007/978-3-031-41181-6_9.

[Bar+25] Manuel Barbosa, Matthias J Kannwischer, Thing-han Lim, Peter Schwabe, and Pierre-Yves Strub. *Formally Verified Correctness Bounds for Lattice-Based Cryptography*. To Appear in *ACM CCS 2025*. 2025. URL: https://eprint.iacr.org/2025/1562.

[BDF20] Koen de Boer, Léo Ducas, and Serge Fehr. "On the Quantum Complexity of the Continuous Hidden Subgroup Problem". In: *EUROCRYPT 2020, Part II*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12106. LNCS. Springer, Cham, May 2020, pp. 341–370. DOI: 10.1007/978-3-030-45724-2_12.

[Bec+16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. "New directions in nearest neighbor searching with applications to lattice sieving". In: *27th SODA*. Ed. by Robert Krauthgamer. ACM-SIAM, Jan. 2016, pp. 10–24. DOI: 10.1137/1.9781611974331.ch2.

[Bel+01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. "Key-Privacy in Public-Key Encryption". In: *ASIACRYPT 2001*. Ed. by Colin Boyd. Vol. 2248. LNCS. Springer, Berlin, Heidelberg, Dec. 2001, pp. 566–582. DOI: 10.1007/3-540-45682-1_33.

[Ber06] Daniel J. Bernstein. "Curve25519: New Diffie-Hellman Speed Records". In: *PKC 2006*. Ed. by Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin. Vol. 3958. LNCS. Springer, Berlin, Heidelberg, Apr. 2006, pp. 207–228. DOI: 10.1007/11745853_14.

[Ber22] Daniel J. Bernstein. *Multi-ciphertext security degradation for lattices*. Cryptology ePrint Archive, Report 2022/1580. 2022. URL: https://eprint.iacr.org/2022/1580.

[Ber23] Daniel J. Bernstein. *Asymptotics of hybrid primal lattice attacks*. Cryptology ePrint Archive, Report 2023/1892. 2023. URL: https://eprint.iacr.org/2023/1892.

[BF14] Jean-François Biasse and Claus Fieker. "Subexponential class group and unit group computation in large degree number fields". In: *LMS Journal of Computation and Mathematics* 17.A (2014), pp. 385–403. DOI: 10.1112/S1461157014000345.

[BF25] Koen de Boer and Joël Felderhoff. "Quantumly Computing S-unit Groups in Quantified Polynomial Time and Space". In: *Cryptology ePrint Archive* (2025).

[BGJ15] Anja Becker, Nicolas Gama, and Antoine Joux. *Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search*. Cryptology ePrint Archive, Report 2015/522. 2015. URL: https://eprint.iacr.org/2015/522.

[Bha+21] Shivam Bhasin, Jan-Pieter D'Anvers, Daniel Heinz, Thomas Pöppelmann, and Michiel Van Beirendonck. "Attacking and Defending Masked Polynomial Comparison". In: *IACR TCHES* 2021.3 (2021), pp. 334–359. ISSN: 2569-2925. DOI: 10.46586/tches.v2021.i3.334-359. URL: https://tches.iacr.org/index.php/TCHES/article/view/8977.

[Bia+17] Jean-François Biasse, Thomas Espitau, Pierre-Alain Fouque, Alexandre Gélin, and Paul Kirchner. "Computing Generator in Cyclotomic Integer Rings - A Subfield Algorithm for the Principal Ideal Problem in $L_{|\Delta_{\mathbb{K}}|}(\frac{1}{2})$ and Application to the Cryptanalysis of a FHE Scheme". In: *EUROCRYPT 2017, Part I*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. LNCS. Springer, Cham, 2017, pp. 60–88. DOI: 10.1007/978-3-319-56620-7_3.

[Bin+19] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. "Tighter Proofs of CCA Security in the Quantum Random Oracle Model". In: *TCC 2019, Part II*. Ed. by Dennis Hofheinz and Alon Rosen. Vol. 11892. LNCS. Springer, Cham, Dec. 2019, pp. 61–90. DOI: 10.1007/978-3-030-36033-7_3.

[BKW00] Avrim Blum, Adam Kalai, and Hal Wasserman. "Noise-tolerant learning, the parity problem, and the statistical query model". In: *32nd ACM STOC*. ACM Press, May 2000, pp. 435–440. DOI: 10.1145/335305.335355.

[BL16] Anja Becker and Thijs Laarhoven. "Efficient (Ideal) Lattice Sieving Using Cross-Polytope LSH". In: *AFRICACRYPT 16*. Ed. by David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi. Vol. 9646. LNCS. Springer, Cham, Apr. 2016, pp. 3–23. DOI: 10.1007/978-3-319-31517-1_1.

[BLS16] Shi Bai, Thijs Laarhoven, and Damien Stehlé. "Tuple lattice sieving". In: *LMS Journal of Computation and Mathematics* 19.A (2016), pp. 146–162. DOI: 10.1112/S1461157016000292.

[Bon+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random Oracles in a Quantum World". In: *ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. LNCS. Springer, Berlin, Heidelberg, Dec. 2011, pp. 41–69. DOI: 10.1007/978-3-642-25385-0_3.

[Bon+23] Xavier Bonnetain, André Chailloux, André Schrottenloher, and Yixin Shen. "Finding Many Collisions via Reusable Quantum Walks: Application to Lattice Sieving". In: *EUROCRYPT 2023, Part V*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14008. LNCS. Springer, Cham, Apr. 2023, pp. 221–251. DOI: 10.1007/978-3-031-30589-4_8.

[Bos+15] Joppe W. Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. "Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem". In: *2015 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2015, pp. 553–570. DOI: 10.1109/SP.2015.40.

[Bos+16] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. "Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE". In: *ACM CCS 2016*. Ed. by Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi. ACM Press, Oct. 2016, pp. 1006–1018. DOI: 10.1145/2976749.2978425.

[Bos+18] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM". In: *2018 IEEE European Symposium on Security and Privacy*. IEEE Computer Society Press, Apr. 2018, pp. 353–367. DOI: 10.1109/EuroSP.2018.00032.

[Bos+21] Joppe W. Bos, Marc Gourjon, Joost Renes, Tobias Schneider, and Christine van Vredendaal. "Masking Kyber: First- and Higher-Order Implementations". In: *IACR TCHES* 2021.4 (2021), pp. 173–214. ISSN: 2569-2925. DOI: 10.46586/tches.v2021.i4.173-214. URL: https://tches.iacr.org/index.php/TCHES/article/view/9064.

[Bou+21] Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. "On the Hardness of Module-LWE with Binary Secret". In: *CT-RSA 2021*. Ed. by Kenneth G. Paterson. Vol. 12704. LNCS. Springer, Cham, May 2021, pp. 503–526. DOI: 10.1007/978-3-030-75539-3_21.

[Bou+22] Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. "Entropic Hardness of Module-LWE from Module-NTRU". In: *INDOCRYPT 2022*. Ed. by Takanori Isobe and Santanu Sarkar. Vol. 13774. LNCS. Springer, Cham, Dec. 2022, pp. 78–99. DOI: 10.1007/978-3-031-22912-1_4.

[Boy+09] Colin Boyd, Yvonne Cliff, Juan M. Gonzalez Nieto, and Kenneth G. Paterson. "One-round key exchange in the standard model". In: *International Journal of Applied Cryptography* 1.3 (Feb. 2009), pp. 181–199. DOI: 10.1504/IJACT.2009.023466.

[BP] Joseph Birr-Pixton. *Rustls: A modern TLS library in Rust*. URL: https://github.com/rustls/rustls (visited on 12/22/2022).

[BR93] Mihir Bellare and Phillip Rogaway. "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols". In: *ACM CCS 93*. Ed. by Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby. ACM Press, Nov. 1993, pp. 62–73. DOI: 10.1145/168588.168596.

[Bra+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. "Classical hardness of learning with errors". In: *45th ACM STOC*. Ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM Press, June 2013, pp. 575–584. DOI: 10.1145/2488608.2488680.

[Bru+16] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. "Flush, Gauss, and Reload - A Cache Attack on the BLISS Lattice-Based Signature Scheme". In: *CHES 2016*. Ed. by Benedikt Gierlichs and Axel Y. Poschmann. Vol. 9813. LNCS. Springer, Berlin, Heidelberg, Aug. 2016, pp. 323–345. DOI: 10.1007/978-3-662-53140-2_16.

[BS15] J.-F. Biasse and F. Song. *A note on the quantum attacks against schemes relying on the hardness of finding a short generator of an ideal in* $\mathbb{Q}(\zeta_{2^n})$. Tech. rep. 2015-12. Revision of September 28th 2015. The University of Waterloo, 2015.

[BS+20] Kevin Bürstinghaus-Steinbach, Christoph Krauß, Ruben Niederhagen, and Michael Schneider. "Post-Quantum TLS on Embedded Systems: Integrating and Evaluating Kyber and SPHINCS+ with mbed TLS". In: *ASIACCS 20*. Ed. by Hung-Min Sun, Shiuh-Pyng Shieh, Guofei Gu, and Giuseppe Ateniese. ACM Press, Oct. 2020, pp. 841–852. DOI: 10.1145/3320269.3384725.

[BSW18] Shi Bai, Damien Stehlé, and Weiqiang Wen. "Measuring, Simulating and Exploiting the Head Concavity Phenomenon in BKZ". In: *ASIACRYPT 2018, Part I*. Ed. by Thomas Peyrin and Steven Galbraith. Vol. 11272. LNCS. Springer, Cham, Dec. 2018, pp. 369–404. DOI: 10.1007/978-3-030-03326-2_13.

[BW25] Kaveh Bashiri and Andreas Wiemers. "On the independence heuristic in the dual attack". In: *Journal of Mathematical Cryptology* 19.1 (2025), p. 20240028. DOI: doi:10.1515/jmc-2024-0028. URL: https://doi.org/10.1515/jmc-2024-0028.

[Car+25] Kevin Carrier, Charles Meyer-Hilfiger, Yixin Shen, and Jean-Pierre Tillich. "Assessing the Impact of a Variant of MATZOV's Dual Attack on Kyber". In: *Advances in Cryptology – CRYPTO 2025*. Ed. by Yael Tauman Kalai and Seny F. Kamara. Cham: Springer Nature Switzerland, 2025, pp. 444–476. ISBN: 978-3-032-01855-7.

[CC21] Matt Campagna and Eric Crockett. *Hybrid Post-Quantum Key Encapsulation Methods (PQ KEM) for Transport Layer Security 1.2 (TLS)*. Internet-Draft draft-campagna-tls-bike-sike-hybrid-07. Work in Progress. Internet Engineering Task Force, Sept. 2, 2021. 17 pp. URL: https://datatracker.ietf.org/doc/html/draft-campagna-tls-bike-sike-hybrid-07.

[CDM24] Cas Cremers, Alexander Dax, and Niklas Medinger. "Keeping Up with the KEMs: Stronger Security Notions for KEMs and Automated Analysis of KEM-based Protocols". In: *ACM CCS 2024*. Ed. by Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie. ACM Press, Oct. 2024, pp. 1046–1060. DOI: 10.1145/3658644.3670283.

[CDW17] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. "Short Stickelberger Class Relations and Application to Ideal-SVP". In: *EUROCRYPT 2017, Part I*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. LNCS. Springer, Cham, 2017, pp. 324–348. DOI: 10.1007/978-3-319-56620-7_12.

[CDW21] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. "Mildly Short Vectors in Cyclotomic Ideal Lattices in Quantum Polynomial Time". In: *J. ACM* 68.2 (Jan. 2021). DOI: 10.1145/3431725.

[Cel+21] Sofía Celi, Armando Faz-Hernández, Nick Sullivan, Goutam Tamvada, Luke Valenta, Thom Wiggers, Bas Westerbaan, and Christopher A. Wood. "Implementing and Measuring KEMTLS". In: *LATINCRYPT 2021*. Ed. by Patrick Longa and Carla Ràfols. Vol. 12912. LNCS. Springer, Cham, Oct. 2021, pp. 88–107. DOI: 10.1007/978-3-030-88238-9_5.

[Che+20] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa. *NTRU*. Tech. rep. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions. National Institute of Standards and Technology, 2020.

[Che+23] Zhao Chen, Xianhui Lu, Dingding Jia, and Bao Li. "IND-CCA Security of Kyber in the Quantum Random Oracle Model, Revisited". In: *Information Security and Cryptology*. Springer Nature Switzerland, 2023, pp. 148–166. DOI: 10.1007/978-3-031-26553-2_8.

[CL01] Jan Camenisch and Anna Lysyanskaya. "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation". In: *EUROCRYPT 2001*. Ed. by Birgit Pfitzmann. Vol. 2045. LNCS. Springer, Berlin, Heidelberg, May 2001, pp. 93–118. DOI: 10.1007/3-540-44987-6_7.

[CL21] André Chailloux and Johanna Loyer. "Lattice Sieving via Quantum Random Walks". In: *ASIACRYPT 2021, Part IV*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13093. LNCS. Springer, Cham, Dec. 2021, pp. 63–91. DOI: 10.1007/978-3-030-92068-5_3.

[CL23] André Chailloux and Johanna Loyer. "Classical and Quantum 3 and 4-Sieves to Solve SVP with Low Memory". In: *Post-Quantum Cryptography - 14th International Workshop, PQCrypto 2023*. Ed. by Thomas Johansson and Daniel Smith-Tone. Springer, Cham, Aug. 2023, pp. 225–255. DOI: 10.1007/978-3-031-40003-2_9.

[Clo] Cloudflare. *Cloudflare Radar – Adoption & Usage – Post-quantum encryption*. URL: https://radar.cloudflare.com/adoption-and-usagexpost-quantum-encryption (visited on 11/11/2025).

[CN11] Yuanmi Chen and Phong Q. Nguyen. "BKZ 2.0: Better Lattice Security Estimates". In: *ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. LNCS. Springer, Berlin, Heidelberg, Dec. 2011, pp. 1–20. DOI: 10.1007/978-3-642-25385-0_1.

[Con25] Deirdre Connolly. *ML-KEM Post-Quantum Key Agreement for TLS 1.3*. Internet-Draft draft-ietf-tls-mlkem-05. Work in Progress. Internet Engineering Task Force, Nov. 2025. 10 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-tls-mlkem/05/.

[CPS19] Eric Crockett, Christian Paquin, and Douglas Stebila. *Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH*. Workshop Record of the Second PQC Standardization Conference. 2019. iacr: 2019/858. URL: https://csrc.nist.gov/CSRC/media/Events/Second-PQC-Standardization-Conference/documents/accepted-papers/stebila-prototyping-post-quantum.pdf.

[Cra+16] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. "Recovering Short Generators of Principal Ideals in Cyclotomic Rings". In: *EUROCRYPT 2016, Part II*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. LNCS. Springer, Berlin, Heidelberg, May 2016, pp. 559–585. DOI: 10.1007/978-3-662-49896-5_20.

[CS03] Ronald Cramer and Victor Shoup. "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack". In: *SIAM Journal on Computing* 33.1 (2003), pp. 167–226. DOI: 10.1137/S0097539702403773.

[D'A+19] Jan-Pieter D'Anvers, Qian Guo, Thomas Johansson, Alexander Nilsson, Frederik Vercauteren, and Ingrid Verbauwhede. "Decryption Failure Attacks on IND-CCA Secure Lattice-Based Schemes". In: *PKC 2019, Part II*. Ed. by Dongdai Lin and Kazue Sako. Vol. 11443. LNCS. Springer, Cham, Apr. 2019, pp. 565–598. DOI: 10.1007/978-3-030-17259-6_19.

[Dac+20] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. "LWE with Side Information: Attacks and Concrete Security Estimation". In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Cham, Aug. 2020, pp. 329–358. DOI: 10.1007/978-3-030-56880-1_12.

[DB22] Jan-Pieter D'Anvers and Senne Batsleer. "Multitarget Decryption Failure Attacks and Their Application to Saber and Kyber". In: *PKC 2022, Part I*. Ed. by Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe. Vol. 13177. LNCS. Springer, Cham, Mar. 2022, pp. 3–33. DOI: 10.1007/978-3-030-97121-2_1.

[DEP23] Léo Ducas, Thomas Espitau, and Eamonn W. Postlethwaite. "Finding Short Integer Solutions When the Modulus Is Small". In: *CRYPTO 2023, Part III*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14083. LNCS. Springer, Cham, Aug. 2023, pp. 150–176. DOI: 10.1007/978-3-031-38548-3_6.

[DEP25] Léo Ducas, Lynn Engelberts, and Paola de Perthuis. "Predicting Module-Lattice Reduction". In: *Cryptology ePrint Archive* (2025).

[Din+22] Xiaohui Ding, Muhammed F. Esgin, Amin Sakzad, and Ron Steinfeld. "An Injectivity Analysis of Crystals-Kyber and Implications on Quantum Security". In: *ACISP 22*. Ed. by Khoa Nguyen, Guomin Yang, Fuchun Guo, and Willy Susilo. Vol. 13494. LNCS. Springer, Cham, Nov. 2022, pp. 332–351. DOI: 10.1007/978-3-031-22301-3_17.

[DLW20] Léo Ducas, Thijs Laarhoven, and Wessel P. J. van Woerden. "The Randomized Slicer for CVPP: Sharper, Faster, Smaller, Batchier". In: *PKC 2020, Part II*. Ed. by Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas. Vol. 12111. LNCS. Springer, Cham, May 2020, pp. 3–36. DOI: 10.1007/978-3-030-45388-6_1.

[DM13] Nico Döttling and Jörn Müller-Quade. "Lossy Codes and a New Variant of the Learning-With-Errors Problem". In: *EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Vol. 7881. LNCS. Springer, Berlin, Heidelberg, May 2013, pp. 18–34. DOI: 10.1007/978-3-642-38348-9_2.

[DMG23] Viet Ba Dang, Kamyar Mohajerani, and Kris Gaj. "High-Speed Hardware Architectures and FPGA Benchmarking of CRYSTALS-Kyber, NTRU, and Saber". In: *IEEE Transactions on Computers* 72.2 (2023), pp. 306–320. DOI: 10.1109/TC.2022.3222954.

[Don+22] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Online-Extractability in the Quantum Random-Oracle Model". In: *EUROCRYPT 2022, Part III*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13277. LNCS. Springer, Cham, 2022, pp. 677–706. DOI: 10.1007/978-3-031-07082-2_24.

[Dor+24] Joao F. Doriguello, George Giapitzakis, Alessandro Luongo, and Aditya Morolia. *On the practicality of quantum sieving algorithms for the shortest vector problem*. Cryptology ePrint Archive, Report 2024/1692. 2024. URL: https://eprint.iacr.org/2024/1692.

[DP23a] Léo Ducas and Ludo N. Pulles. *Accurate Score Prediction for Dual-Sieve Attacks*. Cryptology ePrint Archive, Report 2023/1850. 2023. URL: https://eprint.iacr.org/2023/1850.

[DP23b] Léo Ducas and Ludo N. Pulles. "Does the Dual-Sieve Attack on Learning with Errors Even Work?" In: *CRYPTO 2023, Part III*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14083. LNCS. Springer, Cham, Aug. 2023, pp. 37–69. DOI: 10.1007/978-3-031-38548-3_2.

[DPW19] Léo Ducas, Maxime Plançon, and Benjamin Wesolowski. "On the Shortness of Vectors to Be Found by the Ideal-SVP Quantum Algorithm". In: *CRYPTO 2019, Part I*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11692. LNCS. Springer, Cham, Aug. 2019, pp. 322–351. DOI: 10.1007/978-3-030-26948-7_12.

[DSW21] Léo Ducas, Marc Stevens, and Wessel P. J. van Woerden. "Advanced Lattice Sieving on GPUs, with Tensor Cores". In: *EUROCRYPT 2021, Part II*. Ed. by Anne Canteaut and François-Xavier Standaert. Vol. 12697. LNCS. Springer, Cham, Oct. 2021, pp. 249–279. DOI: 10.1007/978-3-030-77886-6_9.

[Dub+23] Elena Dubrova, Kalle Ngo, Joel Gärtner, and Ruize Wang. "Breaking a Fifth-Order Masked Implementation of CRYSTALS-Kyber by Copy-Paste". In: *Proceedings of the 10th ACM Asia Public-Key Cryptography Workshop*. APKC '23. Melbourne, VIC, Australia: Association for Computing Machinery, 2023, pp. 10–20. ISBN: 9798400701832. DOI: 10.1145/3591866.3593072. URL: https://doi.org/10.1145/3591866.3593072.

[Duc18] Léo Ducas. "Shortest Vector from Lattice Sieving: A Few Dimensions for Free". In: *EUROCRYPT 2018, Part I*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10820. LNCS. Springer, Cham, 2018, pp. 125–145. DOI: 10.1007/978-3-319-78381-9_5.

[Duc22a] Léo Ducas. *Comment on "Improved Dual Lattice Attack"*. NIST PQC Forum. May 2022. URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/Fm4cDfsx65s/m/BZFRC8hiAAAJ.

[Duc22b] Léo Ducas. "Estimating the Hidden Overheads in the BDGL Lattice Sieving Algorithm". In: *Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022*. Ed. by Jung Hee Cheon and Thomas Johansson. Springer, Cham, Sept. 2022, pp. 480–497. DOI: 10.1007/978-3-031-17234-2_22.

[dv25] Koen de Boer and Wessel P. J. van Woerden. *Lattice-based Cryptography: A survey on the security of the lattice-based NIST finalists*. Cryptology ePrint Archive, Report 2025/304. 2025. URL: https://eprint.iacr.org/2025/304.

[DVV18] Jan-Pieter D'Anvers, Frederik Vercauteren, and Ingrid Verbauwhede. *On the impact of decryption failures on the security of LWE/LWR based schemes*. Cryptology ePrint Archive, Report 2018/1089. 2018. URL: https://eprint.iacr.org/2018/1089.

[DW21] Léo Ducas and Wessel P. J. van Woerden. "NTRU Fatigue: How Stretched is Overstretched?" In: *ASIACRYPT 2021, Part IV*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13093. LNCS. Springer, Cham, Dec. 2021, pp. 3–32. DOI: 10.1007/978-3-030-92068-5_1.

[Eis+14] Kirsten Eisenträger, Sean Hallgren, Alexei Kitaev, and Fang Song. "A quantum algorithm for computing the unit group of an arbitrary degree number field". In: *46th ACM STOC*. Ed. by David B. Shmoys. ACM Press, 2014, pp. 293–302. DOI: 10.1145/2591796.2591860.

[EJK20] Thomas Espitau, Antoine Joux, and Natalia Kharchenko. "On a Dual/Hybrid Approach to Small Secret LWE - A Dual/Enumeration Technique for Learning with Errors and Application to Security Estimates of FHE Schemes". In: *INDOCRYPT 2020*. Ed. by Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran. Vol. 12578. LNCS. Springer, Cham, Dec. 2020, pp. 440–462. DOI: 10.1007/978-3-030-65277-7_20.

[EK20] Thomas Espitau and Paul Kirchner. *The nearest-colattice algorithm*. Cryptology ePrint Archive, Report 2020/694. 2020. URL: https://eprint.iacr.org/2020/694.

[Esp+17] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. "Side-Channel Attacks on BLISS Lattice-Based Signatures: Exploiting Branch Tracing against strongSwan and Electromagnetic Emanations in Microcontrollers". In: *ACM CCS 2017*. Ed. by Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu. ACM Press, 2017, pp. 1857–1874. DOI: 10.1145/3133956.3134028.

[FH05] Paul Ford-Hutchinson. *Securing FTP with TLS*. RFC 4217. Oct. 2005. DOI: 10.17487/RFC4217.

[Flu+25] Scott Fluhrer, Quynh Dang, John Preuß Mattsson, Kevin Milner, and Daniel Shiu. *ML-KEM Security Considerations*. Internet-Draft draft-sfluhrer-cfrg-ml-kem-security-considerations-03. Work in Progress. IETF, May 2025. URL: https://datatracker.ietf.org/doc/draft-sfluhrer-cfrg-ml-kem-security-considerations/.

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. "Secure Integration of Asymmetric and Symmetric Encryption Schemes". In: *Journal of Cryptology* 26.1 (Jan. 2013), pp. 80–101. DOI: 10.1007/s00145-011-9114-1.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. "Secure Integration of Asymmetric and Symmetric Encryption Schemes". In: *CRYPTO'99*. Ed. by Michael J. Wiener. Vol. 1666. LNCS. Springer, Berlin, Heidelberg, Aug. 1999, pp. 537–554. DOI: 10.1007/3-540-48405-1_34.

[Fuj+13] Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. "Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism". In: *ASIACCS 13*. Ed. by Kefei Chen, Qi Xie, Weidong Qiu, Ninghui Li, and Wen-Guey Tzeng. ACM Press, May 2013, pp. 83–94. DOI: 10.1145/2484313.2484323.

[Fuj+15] Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. "Strongly secure authenticated key exchange from factoring, codes, and lattices". In: *DCC* 76.3 (2015), pp. 469–504. DOI: 10.1007/s10623-014-9972-2.

[Gen09] Craig Gentry. "Fully homomorphic encryption using ideal lattices". In: *41st ACM STOC*. Ed. by Michael Mitzenmacher. ACM Press, 2009, pp. 169–178. DOI: 10.1145/1536414.1536440.

[GJ21] Qian Guo and Thomas Johansson. "Faster Dual Lattice Attacks for Solving LWE with Applications to CRYSTALS". In: *ASIACRYPT 2021, Part IV*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13093. LNCS. Springer, Cham, Dec. 2021, pp. 33–62. DOI: 10.1007/978-3-030-92068-5_2.

[GJN20] Qian Guo, Thomas Johansson, and Alexander Nilsson. "A Key-Recovery Timing Attack on Post-quantum Primitives Using the Fujisaki-Okamoto Transformation and Its Application on FrodoKEM". In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Cham, Aug. 2020, pp. 359–386. DOI: 10.1007/978-3-030-56880-1_13.

[GKV10] S. Dov Gordon, Jonathan Katz, and Vinod Vaikuntanathan. "A Group Signature Scheme from Lattice Assumptions". In: *ASIACRYPT 2010*. Ed. by Masayuki Abe. Vol. 6477. LNCS. Springer, Berlin, Heidelberg, Dec. 2010, pp. 395–412. DOI: 10.1007/978-3-642-17373-8_23.

[GLX24] Jiangxia Ge, Heming Liao, and Rui Xue. *Measure-Rewind-Extract: Tighter Proofs of One-Way to Hiding and CCA Security in the Quantum Random Oracle Model*. Cryptology ePrint Archive, Report 2024/777. 2024. URL: https://eprint.iacr.org/2024/777.

[GMP22] Paul Grubbs, Varun Maram, and Kenneth G. Paterson. "Anonymous, Robust Post-quantum Public Key Encryption". In: *EUROCRYPT 2022, Part III*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13277. LNCS. Springer, Cham, 2022, pp. 402–432. DOI: 10.1007/978-3-031-07082-2_15.

[GN08] Nicolas Gama and Phong Q. Nguyen. "Predicting Lattice Reduction". In: *EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. LNCS. Springer, Berlin, Heidelberg, Apr. 2008, pp. 31–51. DOI: 10.1007/978-3-540-78967-3_3.

[GNR10] Nicolas Gama, Phong Q. Nguyen, and Oded Regev. "Lattice Enumeration Using Extreme Pruning". In: *EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. LNCS. Springer, Berlin, Heidelberg, 2010, pp. 257–278. DOI: 10.1007/978-3-642-13190-5_13.

[Goo] Google Inc. *BoringSSL*. URL: https://boringssl.googlesource.com/boringssl/.

[Gro25] NIS Cooperation Group. *Recommendation on a Coordinated Implementation Roadmap for the transition to Post-Quantum Cryptography*. June 11, 2025. URL: https://digital-strategy.ec.europa.eu/en/library/recommendation-coordinated-implementation-roadmap-transition-post-quantum-cryptography.

[GW22] Ruben Gonzalez and Thom Wiggers. "KEMTLS vs. Post-quantum TLS: Performance on Embedded Systems". In: *Security, Privacy, and Applied Cryptography Engineering*. Ed. by Lejla Batina, Stjepan Picek, and Mainack Mondal. Cham: Springer Nature Switzerland, 2022, pp. 99–117. ISBN: 978-3-031-22829-2. DOI: 10.1007/978-3-031-22829-2. URL: https://thomwiggers.nl/publication/kemtls-embedded/.

[Hei21] Max Heiser. *Improved Quantum Hypercone Locality Sensitive Filtering in Lattice Sieving*. Cryptology ePrint Archive, Report 2021/1295. 2021. URL: https://eprint.iacr.org/2021/1295.

[Hei+22] Daniel Heinz, Matthias J. Kannwischer, Georg Land, Thomas Pöppelmann, Peter Schwabe, and Amber Sprenkels. *First-Order Masked Kyber on ARM Cortex-M4*. Cryptology ePrint Archive, Report 2022/058. 2022. URL: https://eprint.iacr.org/2022/058.

[Her+23] Julius Hermelink, Erik Mårtensson, Simona Samardjiska, Peter Pessl, and Gabi Dreo Rodosek. "Belief Propagation Meets Lattice Reduction: Security Estimates for Error-Tolerant Key Recovery from Decryption Errors". In: *IACR TCHES* 2023.4 (2023), pp. 287–317. DOI: 10.46586/tches.v2023.i4.287-317.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. "A Modular Analysis of the Fujisaki-Okamoto Transformation". In: *TCC 2017, Part I*. Ed. by Yael Kalai and Leonid Reyzin. Vol. 10677. LNCS. Springer, Cham, Nov. 2017, pp. 341–371. DOI: 10.1007/978-3-319-70500-2_12.

[HHM22] Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. "Failing Gracefully: Decryption Failures and the Fujisaki-Okamoto Transform". In: *ASIACRYPT 2022, Part IV*. Ed. by Shweta Agrawal and Dongdai Lin. Vol. 13794. LNCS. Springer, Cham, Dec. 2022, pp. 414–443. DOI: 10.1007/978-3-031-22972-5_15.

[Hir+09] Philip S. Hirschhorn, Jeffrey Hoffstein, Nick Howgrave-Graham, and William Whyte. "Choosing NTRUEncrypt Parameters in Light of Combined Lattice Reduction and MITM Approaches". In: *ACNS 2009*. Ed. by Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud. Vol. 5536. LNCS. Springer, Berlin, Heidelberg, June 2009, pp. 437–455. DOI: 10.1007/978-3-642-01957-9_27.

[HK25] Kathrin Hövelmanns and Mikhail A. Kudinov. "Treating Dishonest Ciphertexts in Post-quantum KEMs - Explicit vs. Implicit Rejection in the FO Transform". In: *Post-Quantum Cryptography - 16th International Workshop, PQCrypto 2025, Part II*. Ed. by Ruben Niederhagen and Markku-Juhani O. Saarinen. Springer, Cham, Apr. 2025, pp. 325–350. DOI: 10.1007/978-3-031-86602-9_12.

[HKL18] Gottfried Herold, Elena Kirshanova, and Thijs Laarhoven. "Speed-Ups and Time-Memory Trade-Offs for Tuple Lattice Sieving". In: *PKC 2018, Part I*. Ed. by Michel Abdalla and Ricardo Dahab. Vol. 10769. LNCS. Springer, Cham, Mar. 2018, pp. 407–436. DOI: 10.1007/978-3-319-76578-5_14.

[HM24] Kathrin Hövelmanns and Christian Majenz. "A Note on Failing Gracefully: Completing the Picture for Explicitly Rejecting Fujisaki-Okamoto Transforms Using Worst-Case Correctness". In: *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Part II*. Ed. by Markku-Juhani Saarinen and Daniel Smith-Tone. Springer, Cham, June 2024, pp. 245–265. DOI: 10.1007/978-3-031-62746-0_11.

[Hof02] Paul E. Hoffman. *SMTP Service Extension for Secure SMTP over Transport Layer Security*. RFC 3207. Feb. 2002. DOI: 10.17487/RFC3207.

[Höv+20] Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. "Generic Authenticated Key Exchange in the Quantum Random Oracle Model". In: *PKC 2020, Part II*. Ed. by Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas. Vol. 12111. LNCS. Springer, Cham, May 2020, pp. 389–422. DOI: 10.1007/978-3-030-45388-6_14.

[How07] Nick Howgrave-Graham. "A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU". In: *CRYPTO 2007*. Ed. by Alfred Menezes. Vol. 4622. LNCS. Springer, Berlin, Heidelberg, Aug. 2007, pp. 150–169. DOI: 10.1007/978-3-540-74143-5_9.

[HS07] Guillaume Hanrot and Damien Stehlé. "Improved Analysis of Kannan's Shortest Lattice Vector Algorithm". In: *CRYPTO 2007*. Ed. by Alfred Menezes. Vol. 4622. LNCS. Springer, Berlin, Heidelberg, Aug. 2007, pp. 170–186. DOI: 10.1007/978-3-540-74143-5_10.

[HS08] Guillaume Hanrot and Damien Stehlé. "Worst-case Hermite-Korkine-Zolotarev reduced lattice bases". In: *arXiv preprint arXiv:0801.3331* (2008).

[HS10] Guillaume Hanrot and Damien Stehlé. "A complete worst-case analysis of Kannan' s shortest lattice vector algorithm". In: *Manuscript* (2010). Full version of [HS07; HS08], pp. 1–34.

[Hua+20] Yiming Huang, Miaoqing Huang, Zhongkui Lei, and Jiaxuan Wu. "A Pure Hardware Implementation of CRYSTALS-KYBER PQC Algorithm through Resource Reuse". In: *IEICE Electronics Express* advpub (2020), p. 17.20200234. DOI: 10.1587/elex.17.20200234.

[Hül+17] Andreas Hülsing, Joost Rijneveld, John M. Schanck, and Peter Schwabe. "High-Speed Key Encapsulation from NTRU". In: *CHES 2017*. Ed. by Wieland Fischer and Naofumi Homma. Vol. 10529. LNCS. Springer, Cham, Sept. 2017, pp. 232–252. DOI: 10.1007/978-3-319-66787-4_12.

[HW20] Jonathan Hoyland and Christopher Wood. *TLS 1.3 Extended Key Schedule*. Internet-Draft draft-jhoyla-tls-extended-key-schedule-03. Work in Progress. Internet Engineering Task Force, Dec. 2020. 1–7. URL: https://datatracker.ietf.org/doc/html/draft-jhoyla-tls-extended-key-schedule-03.

[Hö20] Kathrin Hövelmanns. "Generic constructions of quantum-resistant cryptosystems". PhD Thesis. PhD thesis. Bochum: Ruhr-Universität Bochum, 2020. URL: https://research.tue.nl/en/publications/generic-constructions-of-quantum-resistant-cryptosystems.

[ISB25] German Federal Office for Information Security (BSI). *Cryptographic Mechanisms: Recommendations and Key Lengths*. Tech. rep. BSI TR-02102-1. BSI, Jan. 2025. URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.html.

[Jao+22] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson. *SIKE*. Tech. rep. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions. National Institute of Standards and Technology, 2022.

[Jen+23] Sönke Jendral, Kalle Ngo, Ruize Wang, and Elena Dubrova. *A Single-Trace Message Recovery Attack on a Masked and Shuffled Implementation of CRYSTALS-Kyber*. Cryptology ePrint Archive, Report 2023/1587. 2023. URL: https://eprint.iacr.org/2023/1587.

[Jia+18] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. "IND-CCA-Secure Key Encapsulation Mechanism in the Quantum Random Oracle Model, Revisited". In: *CRYPTO 2018, Part III*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10993. LNCS. Springer, Cham, Aug. 2018, pp. 96–125. DOI: 10.1007/978-3-319-96878-0_4.

[JZM19a] Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. "Key Encapsulation Mechanism with Explicit Rejection in the Quantum Random Oracle Model". In: *PKC 2019, Part II*. Ed. by Dongdai Lin and Kazue Sako. Vol. 11443. LNCS. Springer, Cham, Apr. 2019, pp. 618–645. DOI: 10.1007/978-3-030-17259-6_21.

[JZM19b] Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. "Tighter Security Proofs for Generic Key Encapsulation Mechanism in the Quantum Random Oracle Model". In: *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*. Ed. by Jintai Ding and Rainer Steinwandt. Springer, Cham, 2019, pp. 227–248. DOI: 10.1007/978-3-030-25510-7_13.

[Kam+22] Tendayi Kamucheka, Alexander Nelson, David Andrews, and Miaoqing Huang. "A Masked Pure-Hardware Implementation of Kyber Cryptographic Algorithm". In: *2022 International Conference on Field-Programmable Technology (ICFPT)*. 2022, pp. 1–1. DOI: 10.1109/ICFPT56656.2022.9974404.

[Kan+] Matthias J. Kannwischer, Richard Petri, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. *pqm4: Post-quantum crypto library for the ARM Cortex-M4*. URL: https://github.com/mupq/pqm4.

[Kan83] Ravi Kannan. "Improved Algorithms for Integer Programming and Related Lattice Problems". In: *15th ACM STOC*. ACM Press, Apr. 1983, pp. 193–206. DOI: 10.1145/800061.808749.

[Kar+25a] Alexander Karenin, Elena Kirshanova, Alexander May, and Julian Nowakowski. "Fast Slicer for Batch-CVP: Making Lattice Hybrid Attacks Practical". In: *Cryptology ePrint Archive* (2025).

[Kar+25b] Alexandr Karenin, Elena Kirshanova, Julian Nowakowski, Eamonn W Postlethwaite, and Fernando Virdia. "Cool+Cruel=Dual". In: *Cryptology ePrint Archive* (2025).

[KEF20] Paul Kirchner, Thomas Espitau, and Pierre-Alain Fouque. "Fast Reduction of Algebraic Lattices over Cyclotomic Fields". In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Cham, Aug. 2020, pp. 155–185. DOI: 10.1007/978-3-030-56880-1_6.

[KF15] Paul Kirchner and Pierre-Alain Fouque. "An Improved BKW Algorithm for LWE with Applications to Cryptography and Lattices". In: *CRYPTO 2015, Part I*. Ed. by Rosario Gennaro and Matthew J. B. Robshaw. Vol. 9215. LNCS. Springer, Berlin, Heidelberg, Aug. 2015, pp. 43–62. DOI: 10.1007/978-3-662-47989-6_3.

[KF17] Paul Kirchner and Pierre-Alain Fouque. "Revisiting Lattice Attacks on Overstretched NTRU Parameters". In: *EUROCRYPT 2017, Part I*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. LNCS. Springer, Cham, 2017, pp. 3–26. DOI: 10.1007/978-3-319-56620-7_1.

[KK18] Franziskus Kiefer and Krzysztof Kwiatkowski. *Hybrid ECDHE-SIDH Key Exchange for TLS*. Internet-Draft draft-kiefer-tls-ecdhe-sidh-00. Work in Progress. Internet Engineering Task Force, Nov. 2018. 13 pp. URL: https://datatracker.ietf.org/doc/html/draft-kiefer-tls-ecdhe-sidh-00.

[KL21] Elena Kirshanova and Thijs Laarhoven. "Lower Bounds on Lattice Sieving and Information Set Decoding". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Cham, Aug. 2021, pp. 791–820. DOI: 10.1007/978-3-030-84245-1_27.

[Kre24] Katharina Kreuzer. "Verification of Correctness and Security Properties for CRYSTALS-KYBER". In: *CSF 2024 Computer Security Foundations Symposium*. IEEE Computer Society Press, July 2024, pp. 511–526. DOI: 10.1109/CSF61375.2024.00016.

[KS23] Ehren Kret and Rolfe Schmidt. *The PQXDH Key Agreement Protocol*. Protocol documentation. Oct. 18, 2023. URL: https://signal.org/docs/specifications/pqxdh/.

[Kuc+20] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. "Measure-Rewind-Measure: Tighter Quantum Random Oracle Model Proofs for One-Way to Hiding and CCA Security". In: *EUROCRYPT 2020, Part III*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12107. LNCS. Springer, Cham, May 2020, pp. 703–728. DOI: 10.1007/978-3-030-45727-3_24.

[KV19] Kris Kwiatkowski and Luke Valenta. *The TLS Post-Quantum Experiment*. Post on the Cloudflare blog. Cloudflare, 2019. URL: https://blog.cloudflare.com/the-tls-post-quantum-experiment/ (visited on 12/22/2022).

[Kwi+19] Krzysztof Kwiatkowski, Nick Sullivan, Adam Langley, Dave Levin, and Alan Mislove. *Measuring TLS key exchange with post-quantum KEM*. Workshop Record of the Second PQC Standardization Conference. 2019. URL: https://csrc.nist.gov/CSRC/media/Events/Second-PQC-Standardization-Conference/documents/accepted-papers/kwiatkowski-measuring-tls.pdf.

[Kwi+25] Kris Kwiatkowski, Panos Kampanakis, Bas Westerbaan, and Douglas Stebila. *Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLSv1.3*. Internet-Draft draft-ietf-tls-ecdhe-mlkem-01. Work in Progress. Internet Engineering Task Force, Sept. 2025. 10 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-tls-ecdhe-mlkem/01/.

[Laa15] Thijs Laarhoven. "Sieving for Shortest Vectors in Lattices Using Angular Locality-Sensitive Hashing". In: *CRYPTO 2015, Part I*. Ed. by Rosario Gennaro and Matthew J. B. Robshaw. Vol. 9215. LNCS. Springer, Berlin, Heidelberg, Aug. 2015, pp. 3–22. DOI: 10.1007/978-3-662-47989-6_1.

[Laa16] Thijs Laarhoven. "Search problems in cryptography: from fingerprinting to lattice sieving". English. PhD Thesis. PhD thesis. Mathematics and Computer Science, Feb. 2016. ISBN: 978-90-386-4021-1.

[Lan16] Adam Langley. *CECPQ1 results*. Blog post. 2016. URL: https://www.imperialviolet.org/2016/11/28/cecpq1.html.

[Lan18] Adam Langley. *CECPQ2*. Blog post. 2018. URL: https://www.imperialviolet.org/2018/12/12/cecpq2.html.

[Lee+19] Changmin Lee, Alice Pellet-Mary, Damien Stehlé, and Alexandre Wallet. "An LLL Algorithm for Module Lattices". In: *ASIACRYPT 2019, Part II*. Ed. by Steven D. Galbraith and Shiho Moriai. Vol. 11922. LNCS. Springer, Cham, Dec. 2019, pp. 59–90. DOI: 10.1007/978-3-030-34621-8_3.

[Li+21] Shuaigang Li, Xianhui Lu, Jiang Zhang, Bao Li, and Lei Bi. "Predicting the Concrete Security of LWE Against the Dual Attack Using Binary Search". In: *ICICS 21, Part I*. Ed. by Debin Gao, Qi Li, Xiaohong Guan, and Xiaofeng Liao. Vol. 12919. LNCS. Springer, Cham, Nov. 2021, pp. 265–282. DOI: 10.1007/978-3-030-88052-1_16.

[LM18] Thijs Laarhoven and Artur Mariano. "Progressive Lattice Sieving". In: *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*. Ed. by Tanja Lange and Rainer Steinwandt. Springer, Cham, 2018, pp. 292–311. DOI: 10.1007/978-3-319-79063-3_14.

[LMP15] Thijs Laarhoven, Michele Mosca, and Joop van de Pol. "Finding shortest lattice vectors faster using quantum search". In: *DCC* 77.2-3 (2015), pp. 375–400. DOI: 10.1007/s10623-015-0067-5.

[LN25] Jianwei Li and Phong Q Nguyen. "A complete analysis of the BKZ lattice reduction algorithm". In: *Journal of Cryptology* 38.1 (2025), p. 12. DOI: 10.1007/s00145-024-09527-0.

[LP11] Richard Lindner and Chris Peikert. "Better Key Sizes (and Attacks) for LWE-Based Encryption". In: *CT-RSA 2011*. Ed. by Aggelos Kiayias. Vol. 6558. LNCS. Springer, Berlin, Heidelberg, Feb. 2011, pp. 319–339. DOI: 10.1007/978-3-642-19074-2_21.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. LNCS. Springer, Berlin, Heidelberg, 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1.

[LS15] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *DCC* 75.3 (2015), pp. 565–599. DOI: 10.1007/s10623-014-9938-4.

[MAT22] MATZOV. *Report on the Security of LWE: Improved Dual Lattice Attack*. Tech. rep. 2022. DOI: https://doi.org/10.5281/zenodo.6493704.

[Mat25] John Preuss Mattsson. *PQC Dialogue with Government Stakeholders*. Transcript of an IETF side-meeting posted on a mailing list. May 13, 2025. URL: https://mailarchive.ietf.org/arch/msg/pqc/14f3hjUlpiwSbusornAjRN98QLc/.

[Mic] Microsoft. *Schannel (Security Support Provider)*. URL: https://learn.microsoft.com/en-us/windows/win32/secauthn/schannel.

[Mic18] Daniele Micciancio. "On the Hardness of Learning With Errors with Binary Secrets". In: *Theory of Computing* 14.13 (2018), pp. 1–17. DOI: 10.4086/toc.2018.v014a013.

[Mic25] Microsoft. *Microsoft Edge browser policy PostQuantumKeyAgreementEnabled*. Sept. 18, 2025. URL: https://learn.microsoft.com/en-us/deployedge/microsoft-edge-browser-policies/postquantumkeyagreementenabled.

[MM11] Daniele Micciancio and Petros Mol. "Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions". In: *CRYPTO 2011*. Ed. by Phillip Rogaway. Vol. 6841. LNCS. Springer, Berlin, Heidelberg, Aug. 2011, pp. 465–484. DOI: 10.1007/978-3-642-22792-9_26.

[Moh10] Payman Mohassel. "A Closer Look at Anonymity and Robustness in Encryption Schemes". In: *ASIACRYPT 2010*. Ed. by Masayuki Abe. Vol. 6477. LNCS. Springer, Berlin, Heidelberg, Dec. 2010, pp. 501–518. DOI: 10.1007/978-3-642-17373-8_29.

[Moo+24] Dustin Moody, Ray Perlner, Andrew Regenscheid, Angela Robinson, and David Cooper. *Transition to Post-Quantum Cryptography Standards*. NIST Interagency Report 8547. Initial Public Draft. National Institute of Standards and Technology, Nov. 2024. DOI: 10.6028/NIST.IR.8547.ipd. URL: https://csrc.nist.gov/pubs/ir/8547/ipd.

[Moz] Mozilla Foundation. *Network Security Services (NSS)*. URL: https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS.

[MP12] Daniele Micciancio and Chris Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: *EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. LNCS. Springer, Berlin, Heidelberg, Apr. 2012, pp. 700–718. DOI: 10.1007/978-3-642-29011-4_41.

[MP13] Daniele Micciancio and Chris Peikert. "Hardness of SIS and LWE with Small Parameters". In: *CRYPTO 2013, Part I*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8042. LNCS. Springer, Berlin, Heidelberg, Aug. 2013, pp. 21–39. DOI: 10.1007/978-3-642-40041-4_2.

[MR09] Daniele Micciancio and Oded Regev. "Lattice-based Cryptography". In: *Post-Quantum Cryptography*. Ed. by Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 147–191. DOI: 10.1007/978-3-540-88702-7_5.

[MS20] Tamalika Mukherjee and Noah Stephens-Davidowitz. "Lattice Reduction for Modules, or How to Reduce ModuleSVP to ModuleSVP". In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Cham, Aug. 2020, pp. 213–242. DOI: 10.1007/978-3-030-56880-1_8.

[Muj+24] Catinca Mujdei, Lennert Wouters, Angshuman Karmakar, Arthur Beckers, Jose Maria Bermudo Mera, and Ingrid Verbauwhede. "Side-channel Analysis of Lattice-based Post-quantum Cryptography: Exploiting Polynomial Multiplication". In: *ACM Transactions on Embedded Computing Syststems* 23.2 (Mar. 2024). DOI: 10.1145/3569420. URL: https://doi.org/10.1145/3569420.

[MX23] Varun Maram and Keita Xagawa. "Post-quantum Anonymity of Kyber". In: *PKC 2023, Part I*. Ed. by Alexandra Boldyreva and Vladimir Kolesnikov. Vol. 13940. LNCS. Springer, Cham, May 2023, pp. 3–35. DOI: 10.1007/978-3-031-31368-4_1.

[Nat16] National Institute of Standards and Technology. *Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process*. Dec. 2016. URL: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf (visited on 05/09/2023).

[Nat24a] National Cyber Security Centre. *Next steps in preparing for post-quantum cryptography. How system owners can begin planning for the migration to post-quantum cryptography (PQC)*. Originally published November 2023; updated August 2024. National Cyber Security Centre (NCSC). Aug. 14, 2024. URL: https://www.ncsc.gov.uk/whitepaper/next-steps-preparing-for-post-quantum-cryptography (visited on 01/13/2026).

[Nat24b] National Institute of Standards and Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. Federal Information Processing Standards Publication (FIPS) FIPS 203. NIST has assigned NIST FIPS 203 as the publication identifier for this FIPS. Department of Commerce, Washington, D.C., 2024.

[Nat24c] National Security Agency. *The Commercial National Security Algorithm Suite 2.0 and Quantum Computing FAQ*. Dec. 2024. URL: https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI_CNSA_2.0_FAQ_.PDF.

[New99] Chris Newman. *Using TLS with IMAP, POP3 and ACAP*. RFC 2595. June 1999. DOI: 10.17487/RFC2595. URL: https://www.rfc-editor.org/info/rfc2595.

[Ngu10] Phong Q. Nguyen. "Hermite's Constant and Lattice Algorithms". In: ed. by Phong Q. Nguyen and Brigitte Vallée. ISC. Springer, 2010, pp. 19–69. ISBN: 978-3-642-02294-4. DOI: 10.1007/978-3-642-02295-1.

[NV08] Phong Q Nguyen and Thomas Vidick. "Sieve algorithms for the shortest vector problem are practical". In: *Journal of Mathematical Cryptology* 2.2 (2008), pp. 181–207.

[O'B23] Devon O'Brien. *Protecting Chrome Traffic with Hybrid Kyber KEM*. Aug. 10, 2023. URL: https://blog.chromium.org/2023/08/protecting-chrome-traffic-with-hybrid.html.

[Ode+18] Tobias Oder, Tobias Schneider, Thomas Pöppelmann, and Tim Güneysu. "Practical CCA2-Secure Masked Ring-LWE Implementations". In: *IACR TCHES* 2018.1 (2018), pp. 142–174. ISSN: 2569-2925. DOI: 10.13154/tches.v2018.i1.142-174. URL: https://tches.iacr.org/index.php/TCHES/article/view/836.

[Opea] *OpenSSL: The Open Source toolkit for SSL/TLS*. OpenSSL project. URL: https://www.openssl.org/ (visited on 05/09/2023).

[Opeb] *OpenVPN Protocol*. URL: https://openvpn.net/community-resources/openvpn-protocol/ (visited on 11/10/2025).

[Ope25] Open Quantum Safe. *liboqs: An open-source C library for quantum-safe cryptography*. Version 0.14.0. July 11, 2025. URL: https://github.com/open-quantum-safe/liboqs.

[PBY17] Peter Pessl, Leon Groot Bruinderink, and Yuval Yarom. "To BLISS-B or not to be: Attacking strongSwan's Implementation of Post-Quantum Signatures". In: *ACM CCS 2017*. Ed. by Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu. ACM Press, 2017, pp. 1843–1855. DOI: 10.1145/3133956.3134023.

[Pei09] Chris Peikert. "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract". In: *41st ACM STOC*. Ed. by Michael Mitzenmacher. ACM Press, 2009, pp. 333–342. DOI: 10.1145/1536414.1536461.

[PG14] Thomas Pöppelmann and Tim Güneysu. "Towards Practical Lattice-Based Public-Key Encryption on Reconfigurable Hardware". In: *SAC 2013*. Ed. by Tanja Lange, Kristin Lauter, and Petr Lisonek. Vol. 8282. LNCS. Springer, Berlin, Heidelberg, Aug. 2014, pp. 68–85. DOI: 10.1007/978-3-662-43414-7_4.

[PHS19] Alice Pellet-Mary, Guillaume Hanrot, and Damien Stehlé. "Approx-SVP in Ideal Lattices with Pre-processing". In: *EUROCRYPT 2019, Part II*. Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11477. LNCS. Springer, Cham, May 2019, pp. 685–716. DOI: 10.1007/978-3-030-17656-3_24.

[PS24] Amaury Pouly and Yixin Shen. "Provable Dual Attacks on Learning with Errors". In: *EUROCRYPT 2024, Part VII*. Ed. by Marc Joye and Gregor Leander. Vol. 14657. LNCS. Springer, Cham, May 2024, pp. 256–285. DOI: 10.1007/978-3-031-58754-2_10.

[PST20] Christian Paquin, Douglas Stebila, and Goutam Tamvada. "Benchmarking Post-quantum Cryptography in TLS". In: *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*. Ed. by Jintai Ding and Jean-Pierre Tillich. Springer, Cham, 2020, pp. 72–91. DOI: 10.1007/978-3-030-44223-1_5.

[PV21] Eamonn W. Postlethwaite and Fernando Virdia. "On the Success Probability of Solving Unique SVP via BKZ". In: *PKC 2021, Part I*. Ed. by Juan Garay. Vol. 12710. LNCS. Springer, Cham, May 2021, pp. 68–98. DOI: 10.1007/978-3-030-75245-3_4.

[PV25] Ludo N Pulles and Paul Vié. "Accelerating the Primal Hybrid Attack against Sparse LWE using GPUs". In: *Cryptology ePrint Archive* (2025).

[Raj+23] Gokulnath Rajendran, Prasanna Ravi, Jan-Pieter D'Anvers, Shivam Bhasin, and Anupam Chattopadhyay. "Pushing the Limits of Generic Side-Channel Attacks on LWE-based KEMs - Parallel PC Oracle Attacks on Kyber KEM and Beyond". In: *IACR TCHES* 2023.2 (2023), pp. 418–446. DOI: 10.46586/tches.v2023.i2.418-446.

[Rav+20a] Prasanna Ravi, Romain Poussier, Shivam Bhasin, and Anupam Chattopadhyay. "On Configurable SCA Countermeasures Against Single Trace Attacks for the NTT: A Performance Evaluation Study over Kyber and Dilithium on the ARM Cortex-M4". In: *Security, Privacy, and Applied Cryptography Engineering: 10th International Conference, SPACE 2020, Kolkata, India, December 17–21, 2020, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2020, pp. 123–146. DOI: 10.1007/978-3-030-66626-2_7.

[Rav+20b] Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. "Generic Side-channel attacks on CCA-secure lattice-based PKE and KEMs". In: *IACR TCHES* 2020.3 (2020), pp. 307–335. ISSN: 2569-2925. DOI: 10.13154/tches.v2020.i3.307-335. URL: https://tches.iacr.org/index.php/TCHES/article/view/8592.

[Rav+22] Prasanna Ravi, Shivam Bhasin, Sujoy Sinha Roy, and Anupam Chattopadhyay. "On Exploiting Message Leakage in (Few) NIST PQC Candidates for Practical Message Recovery Attacks". In: *IEEE Transactions on Information Forensics and Security* 17 (2022), pp. 684–699. DOI: 10.1109/TIFS.2021.3139268.

[Reg05] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *37th ACM STOC*. Ed. by Harold N. Gabow and Ronald Fagin. ACM Press, May 2005, pp. 84–93. DOI: 10.1145/1060590.1060603.

[Rep+15] Oscar Reparaz, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. "A Masked Ring-LWE Implementation". In: *CHES 2015*. Ed. by Tim Güneysu and Helena Handschuh. Vol. 9293. LNCS. Springer, Berlin, Heidelberg, Sept. 2015, pp. 683–702. DOI: 10.1007/978-3-662-48324-4_34.

[Res00] Eric Rescorla. *HTTP Over TLS*. RFC 2818. May 2000. DOI: 10.17487/RFC2818.

[Res18] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018. DOI: 10.17487/RFC8446.

[Saa18] Markku-Juhani O. Saarinen. "Arithmetic coding and blinding countermeasures for lattice signatures - Engineering a side-channel resistant post-quantum signature scheme with compact signatures". In: *Journal of Cryptographic Engineering* 8.1 (Apr. 2018), pp. 71–84. DOI: 10.1007/s13389-017-0149-6.

[Sak00] Kazue Sako. "An Auction Protocol Which Hides Bids of Losers". In: *PKC 2000*. Ed. by Hideki Imai and Yuliang Zheng. Vol. 1751. LNCS. Springer, Berlin, Heidelberg, Jan. 2000, pp. 422–432. DOI: 10.1007/978-3-540-46588-1_28.

[Sch+17] John M. Schanck, Andreas Hulsing, Joost Rijneveld, and Peter Schwabe. *NTRU-HRSS-KEM*. Tech. rep. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions. National Institute of Standards and Technology, 2017.

[Sch22a] Peter Schwabe. *CRYSTALS-Kyber Update*. Talk at the Fourth NIST PQC Standardization Conference. Slides: https://csrc.nist.gov/csrc/media/Presentations/2022/crystals-kyber-update/images-media/session-1-schwabe-crystals-kyber-pqc2022.pdf. Nov. 2022. URL: https://csrc.nist.gov/Presentations/2022/crystals-kyber-update.

[Sch22b] Peter Schwabe. *Kyber decisions, part 2: FO transform*. Message to the `pqc-forum@list.nist.gov` mailing list. Explains hashing of (hash of) the public key into coins/shared key and its robustness rationale. Dec. 2022. URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/C0D3W1KoINY.

[Sch24] Sophie Schmieg. *Unbindable Kemmy Schmidt: ML-KEM is neither MAL-BIND-K-CT nor MAL-BIND-K-PK*. Cryptology ePrint Archive, Report 2024/523. 2024. URL: https://eprint.iacr.org/2024/523.

[Sch87] Claus-Peter Schnorr. "A hierarchy of polynomial time lattice basis reduction algorithms". In: *Theoretical computer science* 53.2-3 (1987), pp. 201–224. DOI: https://doi.org/10.1016/0304-3975(87)90064-8.

[SFG25] Douglas Stebila, Scott Fluhrer, and Shay Gueron. *Hybrid key exchange in TLS 1.3*. Internet-Draft draft-ietf-tls-hybrid-design-16. Work in Progress. Internet Engineering Task Force, Sept. 2025. 23 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/16/.

[SH95] Claus-Peter Schnorr and Horst Helmut Hörner. "Attacking the Chor-Rivest Cryptosystem by Improved Lattice Reduction". In: *EUROCRYPT'95*. Ed. by Louis C. Guillou and Jean-Jacques Quisquater. Vol. 921. LNCS. Springer, Berlin, Heidelberg, May 1995, pp. 1–12. DOI: 10.1007/3-540-49264-X_1.

[She+23] Muyan Shen, Chi Cheng, Xiaohan Zhang, Qian Guo, and Tao Jiang. "Find the Bad Apples: An efficient method for perfect key recovery under imperfect SCA oracles - A case study of Kyber". In: *IACR TCHES* 2023.1 (2023), pp. 89–112. DOI: 10.46586/tches.v2023.i1.89-112.

[SM16] Douglas Stebila and Michele Mosca. "Post-quantum Key Exchange for the Internet and the Open Quantum Safe Project". In: *SAC 2016*. Ed. by Roberto Avanzi and Howard M. Heys. Vol. 10532. LNCS. Springer, Cham, Aug. 2016, pp. 14–37. DOI: 10.1007/978-3-319-69453-5_2.

[SS17] John M. Schanck and Douglas Stebila. *A Transport Layer Security (TLS) Extension For Establishing An Additional Shared Secret*. Internet-Draft draft-schanck-tls-additional-keyshare-00. Work in Progress. Internet Engineering Task Force, Apr. 2017. 1–10. URL: https://datatracker.ietf.org/doc/html/draft-schanck-tls-additional-keyshare-00.

[SSW20] Peter Schwabe, Douglas Stebila, and Thom Wiggers. "Post-Quantum TLS Without Handshake Signatures". In: *ACM CCS 2020*. Ed. by Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna. ACM Press, Nov. 2020, pp. 1461–1480. DOI: 10.1145/3372297.3423350.

[SSW21] Peter Schwabe, Douglas Stebila, and Thom Wiggers. "More Efficient Post-quantum KEMTLS with Pre-distributed Public Keys". In: *ESORICS 2021, Part I*. Ed. by Elisa Bertino, Haya Shulman, and Michael Waidner. Vol. 12972. LNCS. Springer, Cham, Oct. 2021, pp. 3–22. DOI: 10.1007/978-3-030-88418-5_1.

[Ste24] Matthias Johann Steiner. "The Complexity of Algebraic Algorithms for LWE". In: *EUROCRYPT 2024, Part III*. Ed. by Marc Joye and Gregor Leander. Vol. 14653. LNCS. Springer, Cham, May 2024, pp. 375–403. DOI: 10.1007/978-3-031-58734-4_13.

[Sul17] Nick Sullivan. *Why TLS 1.3 isn't in browsers yet*. Cloudflare Blog. Dec. 26, 2017. URL: https://blog.cloudflare.com/why-tls-1-3-isnt-in-browsers-yet/.

[SWZ17] John M. Schanck, William Whyte, and Zhenfei Zhang. *Quantum-Safe Hybrid (QSH) Ciphersuite for Transport Layer Security (TLS) version 1.2*. Internet-Draft draft-whyte-qsh-tls12-02. Work in Progress. Internet Engineering Task Force, Jan. 2017. 1–19. URL: https://datatracker.ietf.org/doc/html/draft-whyte-qsh-tls12-02.

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. "Tightly-Secure Key-Encapsulation Mechanism in the Quantum Random Oracle Model". In: *EUROCRYPT 2018, Part III*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10822. LNCS. Springer, Cham, 2018, pp. 520–551. DOI: 10.1007/978-3-319-78372-7_17.

[Tas+22] George Tasopoulos, Jinhui Li, Apostolos P. Fournaris, Raymond K. Zhao, Amin Sakzad, and Ron Steinfeld. "Performance Evaluation of Post-Quantum TLS 1.3 on Resource-Constrained Embedded Systems". In: *Information Security Practice and Experience: 17th International Conference, ISPEC 2022, Taipei, Taiwan, November 23–25, 2022, Proceedings*. Taipei, Taiwan: Springer-Verlag, 2022, pp. 432–451. ISBN: 978-3-031-21279-6. DOI: 10.1007/978-3-031-21280-2_24. URL: https://doi.org/10.1007/978-3-031-21280-2_24.

[Thea] The Go Authors. *crypto/tls package*. URL: https://pkg.go.dev/crypto/tls.

[Theb] The Legion of the Bouncy Castle. *Bouncy Castle Crypto APIs*. URL: https://www.bouncycastle.org/.

[TU16] Ehsan Ebrahimi Targhi and Dominique Unruh. "Post-Quantum Security of the Fujisaki-Okamoto and OAEP Transforms". In: *TCC 2016-B, Part II*. Ed. by Martin Hirt and Adam D. Smith. Vol. 9986. LNCS. Springer, Berlin, Heidelberg, 2016, pp. 192–216. DOI: 10.1007/978-3-662-53644-5_8.

[Uen+22] Rei Ueno, Keita Xagawa, Yutaro Tanaka, Akira Ito, Junko Takahashi, and Naofumi Homma. "Curse of Re-encryption: A Generic Power/EM Analysis on Post-Quantum KEMs". In: *IACR TCHES* 2022.1 (2022), pp. 296–322. DOI: 10.46586/tches.v2022.i1.296-322.

[Why+17] William Whyte, Zhenfei Zhang, Scott Fluhrer, and Oscar Garcia-Morchon. *Quantum-Safe Hybrid (QSH) Key Exchange for Transport Layer Security (TLS) version 1.3*. Internet-Draft draft-whyte-qsh-tls13-06. Work in Progress. Internet Engineering Task Force, Oct. 2017. 19 pp. URL: https://datatracker.ietf.org/doc/html/draft-whyte-qsh-tls13-06.

[Wig24] Thom Wiggers. "Post-Quantum TLS". PhD thesis. Nijmegen, The Netherlands: Radboud University, Jan. 9, 2024. URL: https://thomwiggers.nl/publication/thesis/.

[Xag22] Keita Xagawa. "Anonymity of NIST PQC Round 3 KEMs". In: *EUROCRYPT 2022, Part III*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13277. LNCS. Springer, Cham, 2022, pp. 551–581. DOI: 10.1007/978-3-031-07082-2_20.

[Xia+22] Wenwen Xia, Leizhang Wang, GengWang, Dawu Gu, and Baocang Wang. *Improved Progressive BKZ with Lattice Sieving*. Cryptology ePrint Archive, Report 2022/1343. 2022. URL: https://eprint.iacr.org/2022/1343.

[Xia+24] Wenwen Xia, Leizhang Wang, Geng Wang, Dawu Gu, and Baocang Wang. "A Refined Hardness Estimation of LWE in Two-Step Mode". In: *PKC 2024, Part II*. Ed. by Qiang Tang and Vanessa Teague. Vol. 14603. LNCS. Springer, Cham, Apr. 2024, pp. 3–35. DOI: 10.1007/978-3-031-57725-3_1.

[XL21] Yufei Xing and Shuguo Li. "A Compact Hardware Implementation of CCA-Secure Key Exchange Mechanism CRYSTALS-KYBER on FPGA". In: *IACR TCHES* 2021.2 (2021), pp. 328–356. ISSN: 2569-2925. DOI: 10.46586/tches.v2021.i2.328-356. URL: https://tches.iacr.org/index.php/TCHES/article/view/8797.

[Xu+22] Zhuang Xu, Owen Pemberton, Sujoy Sinha Roy, David Oswald, Wang Yao, and Zhiming Zheng. "Magnifying Side-Channel Leakage of Lattice-Based Cryptosystems With Chosen Ciphertexts: The Case Study of Kyber". In: *IEEE Transactions on Computers* 71.9 (2022), pp. 2163–2176. DOI: 10.1109/TC.2021.3122997.

[XWT25] Dejun Xu, Kai Wang, and Jing Tian. "A Hardware-Friendly Shuffling Countermeasure Against Side-Channel Attacks for Kyber". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 72.3 (2025), pp. 504–508. DOI: 10.1109/TCSII.2025.3528751.

[YD17] Yang Yu and Léo Ducas. "Second Order Statistical Behavior of LLL and BKZ". In: *SAC 2017*. Ed. by Carlisle Adams and Jan Camenisch. Vol. 10719. LNCS. Springer, Cham, Aug. 2017, pp. 3–22. DOI: 10.1007/978-3-319-72565-9_1.

[YZ21] Takashi Yamakawa and Mark Zhandry. "Classical vs Quantum Random Oracles". In: *EUROCRYPT 2021, Part II*. Ed. by Anne Canteaut and François-Xavier Standaert. Vol. 12697. LNCS. Springer, Cham, Oct. 2021, pp. 568–597. DOI: 10.1007/978-3-030-77886-6_20.

[ZDY25] Ziyu Zhao, Jintai Ding, and Bo-Yin Yang. "Sieving with Streaming Memory Access". In: *IACR TCHES* 2025.2 (2025), pp. 362–384. DOI: 10.46586/tches.v2025.i2.362-384.

[Zha19] Mark Zhandry. "How to Record Quantum Queries, and Applications to Quantum Indifferentiability". In: *CRYPTO 2019, Part II*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11693. LNCS. Springer, Cham, Aug. 2019, pp. 239–268. DOI: 10.1007/978-3-030-26951-7_9.

[ZLJ25] Biming Zhou, Yiting Liu, and Haodong Jiang. "SoK: Post-Quantum Key Encapsulation Mechanisms—Security Definitions and Proof Techniques". In: *Security Standardisation Research*. Springer Nature Switzerland, 2025. DOI: 10.1007/978-3-031-87541-0_6.

資料３－４

外部評価：耐量子計算機暗号への移行に関する技術動向調査（案）

１　背景

（１）2022 年 7 月 5 日に NIST から耐量子計算機暗号（PQC）の標準化方式として、公開鍵暗号 1 方式と電子署名 3 方式が発表された。これら 4 方式のうち、格子に基づく公開鍵暗号方式 ML-KEM は FIPS 203 として、格子に基づく署名方式 ML-DSA は FIPS 204 として、ハッシュ関数に基づく署名方式 SLH-DSA は FIPS 205 として、それぞれ 2024 年 8 月 13 日に標準化された。

（２）2024 年度暗号技術検討会において、PQC への対応について議論が行われ、CRYPTREC 暗号リストへの掲載に向けた PQC の技術的検討と、PQC への移行方針の検討を両輪として並行に進めていくべきであるとの合意が得られた。

（３）2025 年度第 1 回暗号技術評価委員会において、PQC への移行に関する技術動向調査を外部評価により実施することが承認された。

２　実施概要

鈴木茜 様（日立製作所）に外部評価を依頼した。選出理由と依頼内容は次のとおり。

（１）選出理由

暗号の 2010 年問題における政府認証基盤の暗号移行に関する事業に携わるほか、近年は PQC 導入に向けた政府動向や標準技術仕様の調査を実施するとともに、今後の PQC 移行に向けて認証基盤システムへの影響を検討するなど、当該分野における知識・経験が豊富である。

（２）依頼内容

PQC への移行に関する技術動向を調査し、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。

３　外部評価報告書の概要【報告事項】

（１）PQC 移行時・導入時における課題（別紙第 2 章）

PQC への移行時および導入時に直面する技術的課題について、用途別（署名用途、守秘用途、鍵共有用途）に整理された。

（２）PQC 導入へのアプローチ（別紙第 3 章）

PQC 導入に伴う全体プロセス、移行計画策定の検討事項、用途別の導入アプローチ、並びに段階的な移行モデルであるハイブリッド構成の位置付けについて整理された。特に、ハイブリッド構成は、既存システムとの連続性を確保しつつ、新たな暗号方式を段階的に導入するための現実的な構成例として位置付けられていると整理された。

（３）ハイブリッド構成に関する解説（別紙第 4 章）

ハイブリッド構成は単一の技術要素ではなく、複数のレイヤーにまたがる設計課題を内包していることが確認された。この認識に基づき、ハイブリッド構成を目的（後方互換性の確保および安全性の維持）と運用主体（アルゴリズム、プロトコル、システム）の各レイヤーから体系的に整理された。

（４）ハイブリッド構成の安全性に関する解説（別紙第 5 章）

標準化文書に記載された内容を基に、ハイブリッド構成における安全性について整理された。特に、安全性が成立するための条件と、それらの条件が実際の運用においてどのような構成要素に依存して具体化されるのかという点に着目された。

（５）ハイブリッド構成の実装・運用に関する解説（別紙第 6 章）

PQC およびハイブリッド構成の実装に関する主要な OSS の整備状況と、実運用環境における実装・移行事例が整理された。特に、国際会議 PQC Conference で報告された以下の事例について紹介された。

① 実運用を想定した PQC 移行および性能評価の事例
② Web PKI における段階的 PQC 導入の事例
③ PKI 階層設計におけるハイブリッド構成の活用事例
④ S/MIME 電子メールにおけるハイブリッド構成の実装事例

（６）PQC 移行に関わる標準化動向の調査（別紙第 7 章）

PQC 移行期におけるハイブリッド構成の方式の取り扱いに着目し、国際的な標準化団体および関連組織における検討状況が整理された。

表 1. 調査対象組織の一覧

No.	組織名	URL
1	NIST	https:www.nist.gov/
2	IETF	https://www.ietf.org/
3	ITU	https://www.itu.int/
4	ETSI	https://www.etsi.org/
5	IEEE	https://www.ieee.org/
6	ISO	https://www.iso.org/
7	ASC X9	https://x9.org/
8	NSA	https://www.nsa.gov/
9	CSA	https://cloudsecurityalliance.org/
10	PQCRYPTO	https://pqcrypto.eu.org/
11	PQCC	https://pqcc.org/

（７）調査結果に関する考察（別紙第 8 章）

2020 年度外部評価報告書「ハイブリッドモードの技術動向調査」の公開時点[1]では、ハイブリッド構成は概念レベルにとどまり、標準化活動も議論が始まったばかりの段階

[1] 2020 年度の報告書では、ハイブリッド構成のことをハイブリッドモードと称していた。

であった。一方、2025 年以降は主要な標準化機関において本格的な標準化活動が進展しており、その動きが活発化していることが確認された。

特に、2026 年 1 月時点では、ハイブリッド構成は主要な標準化機関において技術仕様として確立されており、TLS、X.509、VPN などの各種プロトコルや PKI 基盤に関しても、具体的な実装指針が整備されていることが確認された。

これらの具体的な整備により、ハイブリッド構成は単なる概念段階を超えて、既存プロトコル・証明書・運用基盤の中で「移行期に採用可能な実装構成」として体系的に確立されたと評価できる。

4　審議事項

2025 年度外部評価報告書（別紙）は、耐量子計算機暗号への移行に関する技術動向調査として十分な内容を含んでいると考えられる。このため、本報告書を CRYPTREC の技術調査報告書として公開してよろしいかご審議いただきたい。

以上

耐量子計算機暗号への移行に関する
技術動向調査

株式会社日立製作所

2026 年 1 月

エグゼクティブサマリ

本報告書は、耐量子計算機暗号（PQC）への移行を、暗号アルゴリズムの置換にとどまらないシステム全体の取り組みとして整理する。移行は、業務や、業務で使用する機器が利用する暗号方式の把握（クリプト・インベントリ）、影響評価と優先度付け、方針・計画の策定、段階的な導入・運用という複数段階で進める。移行期には従来暗号と PQC の併用（ハイブリッド構成）が中心的な選択肢となる。

ハイブリッド構成は、アルゴリズム・プロトコル・システムの各レイヤーで設計される。鍵共有では複数の共有のための秘密値を鍵導出関数（KDF）に入力して統合する枠組みが用いられ、署名では複合署名による安全性維持や代替署名・複数証明書の活用により後方互換性を確保する。安全性に関する整理は、複数要素の組合せに依存することから、設計目標と仕様が保証する範囲の区別、実装・運用に起因するリスクへの留意が必要である。

実装・運用面では、OQS／liboqs、OQS-OpenSSL、Bouncy Castle、wolfSSL 等のオープンソースソフトウェア（OSS）やライブラリが検証基盤を提供しており、PKI／メール等では段階的導入の事例が提案されている。性能影響はユースケースに依存するが、クリプトグラフィック・アジリティ（暗号の俊敏性）の確保が移行コストを大きく左右する。

標準化動向として、NIST（ML-KEM／ML-DSA、SP 800-227、SP 800-56C Rev.2）、IETF（TLS ハイブリッド鍵共有、ハイブリッド KEM／署名、MLS コンバイナ、RFC 9794）、ETSI（TS 103 744）が挙げられ、産業連携では Post-Quantum Cryptography Coalition（PQCC）がロードマップ等と通じて、移行戦略と設計指針の枠組みを提供している。

本報告書では、移行課題、導入プロセス、ハイブリッド構成の具体化、安全性整理、実装・運用事例、標準化動向を体系的に記述する。最終的な完全移行を見据えつつ、移行期間のリスク低減と相互運用性の確保を重視する。

目　次

用語

略称	正式名称	日本語訳
OSS	Open Source Software	オープンソースソフトウェア
CMS	Cryptographic Message Syntax	暗号メッセージ構文
OID	Object IDentifier	オブジェクト識別子
PQC	Post-Quantum Cryptography	耐量子計算機暗号
ML-KEM	Module-Lattice-based Key Encapsulation Mechanism	モジュール格子ベース鍵カプセル化方式
ML-DSA	Module-Lattice-based Digital Signature Algorithm	モジュール格子ベースデジタル署名アルゴリズム
RSA	Rivest-Shamir-Adleman	RSA（公開鍵暗号）
DH	Diffie-Hellman	ディフィー・ヘルマン
ECDH	Elliptic Curve Diffie-Hellman	楕円曲線ディフィー・ヘルマン
ECDSA	Elliptic Curve Digital Signature Algorithm	楕円曲線電子署名
KEM	Key Encapsulation Mechanism	鍵カプセル化方式
HKDF	HMAC-based Key Derivation Function	HMAC ベース鍵導出関数
KDF	Key Derivation Function	鍵導出関数
KDM	Key Derivation Method	鍵導出方法
TLS	Transport Layer Security	トランスポート層セキュリティ
MLS	Messaging Layer Security	メッセージング層セキュリティ
PKI	Public Key Infrastructure	公開鍵暗号基盤
X.509	ITU-T X.509	ITU-T 勧告 X.509
PQ/T	Post-Quantum and Traditional	耐量子計算機暗号と既存の公開鍵暗号の（ハイブリッド）構成を指す

1. はじめに

1.1 目的

本報告書の目的は、CRYPTREC が 2025 年に公表した暗号技術ガイドライン [1]や研究動向調査報告書 [2]を踏まえ、耐量子計算機暗号（PQC）への移行に関する最新の技術動向を整理することである。移行期間においては、RSA や楕円曲線暗号などの既存の公開鍵暗号方式と PQC を併用するハイブリッド構成が現実的な選択肢となる。また、ハイブリッド構成については 2020 年のハイブリッドモード[1]の技術動向調査 [3]で技術動向が示されているが、その後の標準化や実装の進展を反映したアップデートが必要である。これらを背景に、移行プロセスの体系化、ハイブリッド構成の整理、安全性評価の枠組み、実装事例の収集、標準化動向の包括的調査を通じて、移行期間のリスク低減と相互運用性の確保を支援することを目的とする。

1.2 本書の構成

本書の構成は以下の通りである。

第 2 章：

移行時・導入時の技術的課題を用途横断で整理する。

第 3 章：

暗号資産の把握（クリプト・インベントリ）、優先度付け、計画策定、段階的導入といった PQC 導入アプローチを整理する。

第 4 章：

ハイブリッド構成をアルゴリズム／プロトコル／システムの各層で解説する。

第 5 章：

各標準化文書に記述されているハイブリッド構成の安全性概念を整理する。

第 6 章：

OSS（OQS/liboqs 等）や PKI／メール／ブラウザなどの 実装・運用事例を紹介する。

第 7 章：

NIST、IETF、ETSI、ITU、IEEE、ISO、X9、NSA、CSA、PQCRYPTO、PQCC の標準化動向を整理する。

第 8 章：

調査結果の考察を述べる。

第 9 章：

参考文献を掲載する。

[1] 2020 年の報告書では、本報告書の「ハイブリッド構成」に相当する用語として「ハイブリッドモード」が用いられていたため、当該箇所では当時の用語をそのまま記載している。

2. PQC 移行時・導入時における課題

本章では、用語の定義を含む PQC 移行における前提整理を行い、PQC への移行時および導入時に直面する技術的課題を整理する。

CRYPTREC の暗号技術ガイドライン [1]や研究動向調査報告書 [2]では、PQC 移行は暗号アルゴリズム単体の置換に留まらず、業務や業務で使用する機器が利用している暗号方式の棚卸し、鍵管理、通信方式、システム更新サイクル、相互運用性などを含むシステム全体の問題として捉える必要があることが指摘されている。

また、PQC 移行は短期間で完了するものではなく、既存方式との共存を含む移行期間を前提として段階的に進められることが想定されている。このため、本章では、CRYPTREC が示す「PQC の導入における課題」の整理を踏まえ、移行期において顕在化する技術的課題を体系的に整理し、2.1 節以降で述べる用途別の課題整理および 3 章における導入アプローチの検討につなげることを目的とする。

2.1 用語の整理と定義

PQC 移行における前提整理を行う前に、本報告書で頻出する用語を以下の通り定義する。

鍵カプセル化（Key Encapsulation Mechanism: KEM）：

鍵カプセル化とは、セッション鍵を生成する機能と、生成された鍵を暗号化する機能を組みあわせた暗号化方式である。暗号化処理をカプセル化（encapsulation）、復号処理をデカプセル化（decapsulation）と呼ぶ。なお、NIST SP 800-227 [4]では、KEM 及び KEM を用いた鍵共有プロトコルの両方を、明示的に宣言することなく KEM と表現しているため、注意が必要である。本報告書では、後者の用法については「KEM を用いた鍵共有」と呼ぶ。

前方秘匿性（(perfect) forward secrecy）：

前方秘匿性とは、鍵共有プロトコルを通じて共有されたセッション鍵の独立性を意味する。多くの場合、セッション鍵の生成に毎回異なる乱数を付加することで、前方秘匿性を実現する。たとえば、DH 鍵共有（$(g^x)^y=(g^y)^x$）において、指数 x, y を毎回ランダムに生成する ephemeral DH（DHE）は前方秘匿性を満たすことが知られている。前方秘匿性を満たす鍵共有プロトコルは、一部のセッション鍵や長期保存鍵（long term key）が漏洩した場合であっても、漏洩する情報の範囲を限定することができる。

後方互換性：

TLS などの通信プロトコルは、接続する端末ごとに実装されているプロトコルのバージョン、暗号スイートが異なることがある。新しい規格において、古い規格との接続可能性を後方互換性（backward compatibility）と呼ぶ。例えば、TLS は ClientHello の中にプロトコルのバージョンと暗号スイートを含んでおり、サーバは ServerHello で提示されたプロトコルのバージョンと暗号スイートの中から自身が対応可能かつ優先度が高いものを返信する。

鍵素材（keying material）：

DH 鍵共有や KEM を用いた鍵共有などで共有された shared secret と追加の補助情報(other input)を入力として、鍵導出方法（key derivation method）を用いて出力されたビット列。

「ハイブリッド」について

PQC への暗号移行では、既存の公開鍵暗号と PQC を組み合わせる方式をまとめてハイブリッド方式と呼んでいた。しかし、IETF などでは、後方互換性（既存システムとの相互接続性）と安全性の観点から、ハイブリッド方式を合成（composite）と混成（hybrid）の 2 つに分けるようになっている。

合成（composite）：

署名検証もしくは鍵共有において、既存の公開鍵暗号方式と PQC を同時に必須利用し、その結果を統合して一つの処理として扱う方式。その特徴から、合成方式は後方互換性を持たない。必ず既存の公開鍵暗号方式と PQC の両方を使う、安全性を重視した方式である。

混成（hybrid）：

署名検証もしくは鍵共有において、既存の公開鍵暗号方式と PQC の処理をそれぞれ独立に行い、パーサ等の上位の処理で既存の公開鍵暗号方式と PQC の使い方を規定する方式のこと。既存の公開鍵暗号方式の処理を従来通りとすることで、後方互換性を実現できる。その一方で、混成方式は既存の公開鍵暗号方式単独での使用が可能となるため、将来的にダウングレード攻撃の対象となる可能性がある。混成方式を用いる場合、適切なタイミングで既存の公開鍵暗号方式単体での使用を禁止する措置を講じる必要があり、クリプトグラフィック・アジリティの実装がより重要となる。

本報告書では、既存の公開鍵暗号方式と PQC を組み合わせる方式を総括して「ハイブリッド」と呼び、上に挙げた各々の構成方法については「混成」「合成」と呼ぶことにする。

鍵確立／鍵合意／鍵共有／鍵交換について

本報告書では、主に公開鍵暗号の署名および鍵共有について取り扱う。しかし、本報告書の中で頻繁に引用される NIST 文書や IETF 文書では、鍵共有について別の用語が用いられていることが多く、分類の境界も異なる。混乱を避けるため、ここでそれぞれの用語の使い方について整理する。

CRYPTREC：

CRYPTREC 暗号リスト [5]は公開鍵暗号の用途を署名、守秘、鍵共有の 3 種類に分類している。電子政府推奨暗号リスト、もしくは推奨候補暗号リストに掲載されているアルゴリズムの中で、守秘に分類されているものは RSA-OAEP のみであり、DH, ECDH, PSEC-KEM が鍵共有に分類されている。なお、[1]では「また、RFC7525 においても、4.1 節において（守秘用途である）RSA key transport は利用すべきでないと記載されており...」と述べており、守秘と後述する鍵配送（key transport）を同一視している。

NIST：

NIST は単純な秘匿目的の守秘に相当する公開鍵暗号を標準化していない。その代わりに、NIST SP 800-56A, B において（二者間の）鍵確立（key establishment）を導入している。鍵確立は、エンティティの双方が共有される鍵素材（keying material）を提供する鍵合意（key agreement）と、一方のエンティティのみが鍵素材を提供する鍵配送（key transport）に分類される。2.2 節で触れるが、KEM は鍵配送と同一視されることがある。一方、NIST は、KEM をその内部構造に応じて鍵合意的に、または鍵配送的に見なすことができると述べている。たとえば SP 800-227 では「ML-KEM could be viewed as a key-agreement scheme」と記されており、ML-KEM は鍵合意として見なすことができる。同様に、RSA-OAEP は一般に鍵配送として見なされる。[4]。

IETF：

CRYPTREC 暗号リストで守秘に分類されている RSA-OAEP は、 [6]では RSAES-OAEP （RSA Encryption Scheme)であるが、 [7]では“RSA key transport mechanisms [RFC8017]”と引用されている。このように、IETF では用語の選び方に揺らぎがあるが、key transport を暗号化方式（守秘）とほぼ同義に取り扱っている。また、鍵確立（鍵合意）の代わりに鍵交換（key exchange）という用語を用いている。

本報告書の対象は主として（NIST の用語で）鍵合意プロトコルを扱い、鍵配送プロトコルは取り上げない。そこで、本報告書では原則として CRYPTREC の用語に従い、紹介する文献による呼称ではなく、一貫して鍵共有を用いる。

なお、CRYPTREC では RSA-OAEP を守秘、PSEC-KEM を鍵共有に分類している。しかし、本報告書は鍵カプセル化（KEM）および KEM を用いた鍵共有（鍵合意）プロトコルを主な調査対象としており、KEM を鍵共有に含めると混乱を生じる懸念がある。そこで、本報告書では、読者の混乱を避けるため、鍵カプセル化は CRYPTREC の特定の分類に入れずにそのまま「鍵カプセル化」（もしくは KEM）と呼ぶことにする。

2.2 PQC 移行における前提整理

PQC の導入は、従来の公開鍵暗号アルゴリズムを別の方式に単純に置き換える問題ではない。安全性の評価や影響範囲はアルゴリズム単体に留まらず、実装、プロトコル設計、運用形態を含むシステム全体として検討する必要がある。そのため、PQC 移行は暗号方式の更新ではなく、既存システム構成との整合を前提とした設計・運用上の課題として捉える必要がある。

また、PQC に関する標準化の進展状況や移行時に顕在化する課題は、暗号の利用用途によって性質が異なる。署名用途では、モジュール格子ベースデジタル署名アルゴリズム（Module-Lattice-based Digital Signature Algorithm: ML-DSA）等のアルゴリズム標準化は進みつつあるものの、X.509 証明書や暗号メッセージ構文（Cryptographic Message Syntax: CMS）、長期署名といった既存 PKI 基盤への組込みに関しては、証明書サイズの増大、相互運用性、後方互換性の確保など、運用およびデータ構造上の課題が残されている。このため、署名用途における主な論点は、署名アルゴリズム自体よりも、その利用基盤との整合にある。

一方、鍵共有用途では、従来の Diffie-Hellman（DH）系方式から鍵カプセル化（KEM）系方式への移行が通信プロトコルの設計に直接的な影響を及ぼす。その結果、アルゴリズムの置換に留まらず、通信手順やメッセージ構成、状態管理、前方秘匿性の扱いを含むプロトコル全体の再検討が必要となり、実装依存性および用途依存性が特に高いという特徴を有する。

さらに、CRYPTREC 等でも指摘されているように、PQC 移行は短期間で完了するものではなく、段階的かつ長期にわたる可能性が高い。この前提の下では、最終的な完全移行のみならず、移行途中におけるリスクの抑制を考慮した構成や運用方針を検討することが重要である。以上の点は、本章における課題整理全体の共通前提として位置付けられる。

2.3 PQC の導入における課題

本節では、2.1 節で整理した前提条件を踏まえ、PQC を既存システムへ導入する際に顕在化する課題を整理する。

PQC 移行は既存方式との共存を含む移行期を前提とするものであり、課題の性質は暗号の利用用途によって大きく異なる。

このため本報告書では、署名、守秘、鍵共有の各用途に分けて整理するとともに、これらに共通する実装・運用および相互運用性に関する課題についても整理する。

2.3.1 署名用途の PQC 移行における課題

一般に、PQC 署名方式は従来方式と比較して鍵長および署名サイズが大きくなる傾向があり、これに伴い証明書サイズや署名付きデータ量の増加が生じる。このような変化は、X.509 証明書や CMS 等に代表される既存 PKI のデータ構造や運用設計に直接的な影響を及ぼす。

また、長期にわたって検証可能性を維持することが求められるアーカイブ用の署名においては、署名生成時点では安全と考えられていた署名方式が、将来的に破られた場合の影響を考慮する必要がある。すなわち、耐量子計算機性を持たない署名方式が、将来、実用的な量子コンピュータが開発され、署名を付したドキュメントの改ざんが行われる等、署名が破られることは十分考えられる。この場合、過去に正当と検証された署名の真正性が事後的に否定される可能性が存在する。このような長期的な真正性に関するリスクは、たとえば証明書の更新が想定されていない IoT 機器や有効期限が長いパスポートなどの ID カードにおいて顕在化する可能性があり、近年の一部の実務文献において、守秘用途における Harvest Now, Decrypt Later（HNDL）に対比する形で言及されている[2]。

さらに、署名は検証者が多数存在するケースが一般的であり、後方互換性の確保が特に重要となる。単独の PQC 署名方式への移行は、既存の検証環境や運用との断絶を招く可能性がある。このため、移行期においては、既存方式との連続性を保ちながら段階的な導入を可能とする手法として、ハイブリッド構成の署名が検討対象となっている。

2.3.2 守秘用途における課題

守秘用途における課題は、公開鍵暗号方式の変更が通信システム全体の設計および運用に広範な影響を及ぼす点にある。RSA や楕円曲線暗号から PQC 方式への移行は、暗号アルゴリズムの更新に留まらず、既存の通信プロトコルやその運用形態との整合性を慎重に検討する必要がある。

CRYPTREC では、守秘目的の公開鍵暗号の用途として Key Encapsulation Mechanism - Data Encapsulation Mechanism（KEM-DEM）構成によるデータ暗号化や、鍵配送プロトコルを挙げている[1] [5]。

このうち、KEM-DEM 構成によるデータ暗号化の用途について、[1]ではドキュメントデータや（他の用途での）鍵情報を通信当事者間で共有する、暗号鍵所有者が鍵情報をバックアップするといったユースケースを挙げている。特にデータや鍵情報のバックアップでは、暗号化データの長期秘匿性が必要になると考えられる。このようなケースは「暗号化データを今から収集し、量子計算機が利用できるようになったら解読する（Harvest Now, Decrypt Later: HNDL）」という脅威に直面するため、PQC の導入が不可欠である。導入に際しては、バックアップされている大量のデータを再暗号化するコストや、再暗号化処理の途中でデータが破損するリスクが課題となると想定される。

なお、鍵配送における PQC 導入の課題は鍵共有と同様であるため、説明は 2.3.4 節に譲る。

---

[2] 将来の量子計算機により現在用いられている公開鍵署名方式が破られた場合、過去に生成・検証された署名や証明書が事後的に偽造可能となるリスクについて、近年の一部の実務・業界文献では“Trust Now, Forge Later（TNFL）”という呼称が用いられている例がある（例：Trust Now, Forge Later (TNFL) - The Overlooked Quantum Threat）。ただし、この用語は現時点では NIST、IETF、ETSI 等の標準文書において確立した専門用語として定義されているものではなく、本報告書では概念的表現として補足的に紹介するに留める。

2.3.3 鍵共有用途における課題

TLS 1.3 などの主要な鍵共有プロトコルでは、DH や楕円曲線 Diffie-Hellman（ECDH）が主流である。現在、標準化された PQC アルゴリズムは KEM のみであり DH や ECDH を直接代替する方式が無い。したがって、鍵共有用途については、従来の DH や ECDH に基づくプロトコルから、KEM に基づくプロトコルへの移行という設計上の大きな転換が求められる。また、実際の鍵共有プロトコルは中間者（Man-in-the-Middle）攻撃を避けるために公開鍵証明書ベースの認証を行うことが多いため、前述の署名に関する課題も併せて検討する必要がある。特に、インターネット標準は多種多様な条件で運用されているサーバ、端末との接続を維持するため、移行を完了していないシステムを考慮した後方互換性が重要であり、既存プロトコルとの整合を前提とした設計が不可欠となる。

2.3.4 実装・運用・相互運用性面での共通課題

用途に共通する課題として、実装および運用面、ならびに相互運用性に関する問題が挙げられる。PQC 実装の成熟度にはばらつきがあり、オープンソースソフトウェア（OSS）、ハードウェアセキュリティモジュール（Hardware Security Module: HSM）、ハードウェアアクセラレータ等の対応状況も一様ではない。また、暗号方式の変更は相互接続試験の実施を不可避とし、鍵更新、失効、ロールオーバーといった運用設計にも影響を及ぼす。

これらの課題は、CRYPTREC による PQC 移行に関する整理においても指摘されており、本節ではそれらを用途横断的な視点から再解釈した。

3. PQC 導入へのアプローチ

PQC 導入に関する基本的な考え方については、CRYPTREC の暗号技術ガイドライン［1］や研究動向調査報告書［2］、本報告書に先行するハイブリッドモード[1]に関する調査報告書［3］をはじめ、産業・実務文献や行政文書において整理が行われている。これらの文献では、PQC 移行を暗号アルゴリズム単体の置換としてではなく、業務や業務で使用する機器が利用する暗号方式の把握、既存方式との共存、段階的な更新を含む中長期的な取り組みとして進める必要性が示されている。

本章では、これらの整理を踏まえ、PQC 導入に関する全体プロセス、移行計画策定の検討事項、用途別の導入アプローチ、ならびに段階的な移行モデルであるハイブリッド構成の位置付けについて整理する。

3.1 PQC 導入に関する全体プロセス

PQC 導入への取り組みは、暗号アルゴリズム単体の置換としてではなく、複数の段階を経て進められる移行プロセスとして整理されている。PQCC（Post-Quantum Cryptography Coalition）が公表している PQC Migration Roadmap［8］においても、PQC 導入は単一の完了時点を目標とするものではなく、準備段階、移行期、移行後の運用といった複数のフェーズに分けて整理されている。

図 3-1 に示す PQC Roadmap Categories では、まず初期段階として、利用中の暗号技術やその用途を把握する作業が位置付けられている。これには、暗号アルゴリズム、プロトコル、鍵管理方式、証明書構成などを整理することが含まれており、後続の検討を行うための基礎情報として重要な工程とされている。

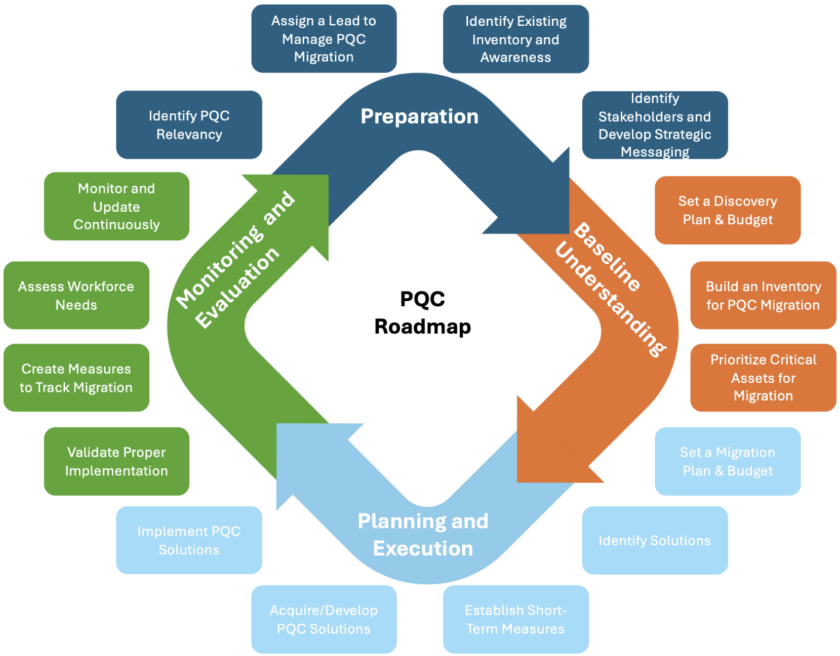


図 3-1 PQC ロードマップ［8］

次の段階では、把握された暗号方式を整理した資産を基に、影響評価や優先度付けを行い、どの領域から移行を進めるべきかを検討する工程が示されている。この段階では、暗号方式の安全性だけでなく、データの保護期間、システムの重要度、更新サイクル、外部組織との相互運用性といった要素を考慮した判断が求められる。

移行期においては、既存方式と PQC を併用する構成や、段階的な切替を前提とした運用が想定されている。PQCC のロードマップでは、この期間を通じてシステムの継続運用を維持しながら、順次

構成を更新していくことが前提とされており、一斉置換を想定しない点が特徴となっている。

さらに、移行後の段階では、導入したPQC構成を運用・維持しつつ、将来的な暗号方式の更新や標準化動向への対応を継続的に行うことが位置付けられている。このことから、PQC導入は一度限りの対応ではなく、長期的な運用プロセスの一部として捉えられている。

このような段階的な整理は、CRYPTREC の暗号技術ガイドラインや金融庁の報告書において示されている考え方とも整合的であり、PQC導入を計画的かつ中長期的に進める上での全体像を示すものと位置付けられる。本章では、この全体プロセスを踏まえた上で、3.2節以降において、移行計画策定における検討事項、用途別の導入アプローチ、段階的導入モデルおよびハイブリッド構成の位置付けについて整理する。

3.2 移行計画策定における検討事項

PQC導入計画を策定するにあたっては、導入対象や影響範囲を事前に整理することが不可欠である。特に、既存システムにおいてどの暗号技術がどの用途で利用されているかを把握する作業は、導入計画全体の基盤となる。さらに、移行計画の実現可能性を確保するためには、必要な予算の見積もりと確保がプロセスとして不可欠である。以下では、このような移行計画策定において主要となる検討事項として、暗号資産の把握と移行対象・優先度の整理について述べる。

3.2.1 暗号使用状況の調査・把握（クリプト・インベントリ）

暗号使用状況の調査・把握（クリプト・インベントリ）では、利用中の暗号アルゴリズム、プロトコル、証明書、鍵管理方式などを一覧化し、それらがどのシステムやデータ保護に用いられているかを整理する。この作業は、PQC移行において何が影響を受けるかを明確化するための出発点として位置付けられる。

具体的には、通信路における暗号化や認証、電子署名や証明書管理、鍵生成・保管・更新といった各機能について、使用されている暗号方式と運用形態を把握することが重要となる。また、暗号技術が利用されている箇所はアプリケーション層に限らず、プロトコル層やミドルウェア、ハードウェアに組み込まれている場合もあるため、システム全体を俯瞰した整理が求められる。

このように暗号資産を整理することで、PQC移行が必要となる対象や、移行時に影響を受ける範囲を把握することが可能となる。さらに、後続の優先度付けや移行計画策定において、前提情報として活用することができる。

3.2.2 移行対象や優先度の検討

クリプト・インベントリを作成した後には、収集した情報を基に、移行対象や優先度を検討する必要がある。すべての暗号技術を同時に更新することは現実的ではないため、どの領域から移行を進めるかを整理することが重要となる。

優先度の検討にあたっては、暗号によって保護されるデータの保護期間や重要度、関連するシステムの役割、更新頻度や保守性といった要素が考慮される。また、外部システムや取引先との接続関係がある場合には、相互運用性や移行時期の調整が必要となる。また、確保した予算の範囲内で、移行対象の優先順位を調整することにより、限られたリソースを効果的に活用し、計画的な移行を進めることが求められる。

これらの観点を踏まえて移行対象と優先度を整理することで、限られたリソースの中でも現実的な移行計画を策定することが可能となる。このような段階的な検討は、PQC移行を中長期的な取り組みとして進める上での基本的なプロセスとして、多くの文献において共通して示されている。

3.3 用途別の導入アプローチ

PQC 導入における具体的な対応は、暗号の利用用途によって性質が異なる。本節では、2.3 節で整理した用途別の課題に対応する形で、導入アプローチを示す。

3.3.1 署名用途における導入アプローチ

署名用途では、2.3.1 節で整理したとおり、鍵長や署名サイズの増大による証明書構造への影響、長期真正性の確保、後方互換性といった課題が存在する。このため、導入アプローチとしては、既存 PKI と独立しない形で段階的に PQC を導入することが基本となる。具体的には、既存の証明書・署名形式を維持しつつ PQC を追加する合成型や混成型の署名方式の活用、検証者側の移行状況を踏まえた複数署名の併存期間の設定、長期検証性を確保するためのタイムスタンプや署名更新の運用ルール整備が挙げられる。これにより、既存の検証環境を保持しつつ PQC を段階的に適用でき、将来的な単独 PQC 署名への移行に向けた基盤整備を進めることが可能となる。

3.3.2 守秘・鍵共有用途における導入アプローチ

守秘用途および鍵共有用途では、2.3.2 節や 2.3.3 節に示したように、量子コンピュータの実用化を前提とした長期秘匿性の確保や、DH／ECDH から KEM を用いた鍵共有への構造転換、大規模な後方互換性確保が課題となる。これらに対する導入アプローチとしては、まず長期秘匿性が求められるデータ暗号化から優先的に PQC KEM を適用し、保管データ・バックアップデータの再暗号化を段階的に実施することが重要である。また、通信プロトコルにおいては、既存方式との連続性を確保するため、PQC KEM と従来の ECDH を併用するハイブリッド鍵共有を導入し、接続性を維持しながら KEM を用いた鍵共有ベースの設計へと移行を進めることが有効である。これにより、運用環境の多様性を損なうことなく、将来的な PQC への完全移行に向けた整合的なステップを構築できる。

3.4 クリプトグラフィック・アジリティの確保

PQC 導入にあたっては、今回の移行に対応するだけでなく、将来的な暗号方式の更新にも対応可能な設計とすることが重要である。CRYPTREC や金融庁の文献においても、暗号方式の更新を前提とした柔軟な設計の必要性が指摘されている。これらは一般に、クリプトグラフィック・アジリティとして整理される。

クリプトグラフィック・アジリティの確保により、暗号アルゴリズムやパラメータの変更をシステム全体の大規模な改修を伴わずに実施することが可能となる。PQC 移行は一度限りの対応ではなく、長期的な継続対応が求められる取り組みであることから、この観点は導入アプローチ全体に共通する重要な要素となる。

3.5 ハイブリッド構成の位置付け

段階的な PQC 移行を進める上で、既存暗号方式と PQC を併用するハイブリッド構成は、移行期における代表的な選択肢として各種文献で取り上げられている。ハイブリッド構成は、既存システムとの連続性を確保しつつ、新たな暗号方式を段階的に導入するための構成として位置付けられる。

一方で、ハイブリッド構成は構成や運用が複雑化する可能性があるため、その適用範囲や設計上の留意点を整理した上で利用する必要がある。本章では、ハイブリッド構成を段階的導入モデルの一例として位置付けるに留め、具体的な方式構成や技術的詳細については、4 章以降で整理する。

4. ハイブリッド構成の構成方法に関する解説

ハイブリッド構成は、PQC 移行期における後方互換性の確保と安全性維持を目的とする構成である。CRYPTREC のガイドライン [1]で示された分類を踏まえ、既存基盤との連続性を保ちつつ、耐量子計算機性を付与するための設計指針を整理する。本章では、ハイブリッド構成の目的と適用主体を体系化し、アルゴリズム・プロトコル・システムの各レイヤーにおける技術仕様例を示す。

4.1 ハイブリッド構成の整理

ハイブリッド構成は、CRYPTREC のガイドライン [1]において、「移行期における後方互換性の確保」と「耐量子計算機性の付与による安全性維持」の両立を目的とする構成として整理されている。本報告書では、この定義を踏まえ、ハイブリッド構成を目的（後方互換性確保／安全性維持）および適用主体（アルゴリズム・プロトコル・システム）の観点から体系化する。本節では、これらの対応関係を表 4-1 に示し、4.2 節以降で各レイヤーの技術仕様を解説する。

- アルゴリズム層（鍵共有アルゴリズム／署名アルゴリズム）
  アルゴリズム層は、鍵カプセル化アルゴリズムと署名アルゴリズムを対象とする。
  アルゴリズム層では、合成型の方式のみ規定されており、後方互換性を考慮した混成型のアルゴリズムは定義されていない。混成型の仕様は個々のプロトコルに任されている。NIST SP 800-227 [4]では合成方式の KEM を用いた鍵共有プロトコルを定めている。また、DH を KEM とみなして利用する方法も定めており、ECDH+ML-KEM(Module-Lattice-based Key Encapsulation Mechanism, モジュール格子ベース鍵カプセル化方式)を合成型の KEM を用いた鍵共有として記述するための基盤を与えている。
  署名についても合成型の複数署名を 1 つの署名構造として統合するコンポジット署名が標準化されつつある。本方式では、既存の公開鍵暗号による署名と PQC 方式による署名の両方の検証が成功することが（合成型の）署名の検証アルゴリズムとなる。また、署名生成においても、それぞれの署名を分離して悪用するリスクを避けるため、補助入力の追加する方式などが提案されている。
- プロトコル層（通信プロトコル／証明書構造）
  プロトコル層は、アルゴリズムを実際の通信・認証処理でどのように活用するかを規定する層である。TLS、MLS、ETSI TS 103 744、IEEE 802.11 Proposal では、複数の KEM を組み合わせる(合成型の)ハイブリッド鍵共有方式が定義されている。証明書構造もプロトコル層で利用され、Discovery-enabled（混成）、Dual-signature（混成）、Composite-signature（合成）など、後方互換性重視の方式と安全性重視の方式の双方が標準化されている。
- システム層
  システム層は、アルゴリズムおよびプロトコルを PKI、メールシステム、鍵管理方針、更新運用などの実環境に統合する領域である。単一仕様としてハイブリッド方式が定義されているわけではなく、複数の方式を組み合わせて移行期の後方互換性と安全性維持を両立するシステム設計が重要となる。実装・運用例として、Hybrid PKI の階層設計や、メールシステムにおける段階的移行方式などがあり、これらは 6 章で実装の事例を紹介する。

ハイブリッド構成は、単一の技術要素ではなく、複数のレイヤーにまたがる設計課題を含む。目的別（互換性確保／安全性維持）と主体別（アルゴリズム、プロトコル、システム）に分類することで、どの層でどの標準仕様や設計指針が必要かを明確化し、後続の詳細解説に向けた体系的な理解を提供することを意図している。

表 4-1 ハイブリッド構成の目的と適用主体による整理

目的 / 主体	後方互換性確保	安全性維持
アルゴリズム	なし	鍵共有アルゴリズム • NIST SP 800-227(4.6. Multi-Algorithm KEMs and PQ/T Hybrids) [4] • NIST SP 800-56C Rev.2[3] [9] • draft-irtf-cfrg-hybrid-kems-07 [10] • draft-ietf-lamps-pq-composite-kem-12 [11][4]
		署名アルゴリズム • draft-ietf-lamps-pq-composite-sigs-14 [12][4]
プロトコル	通信プロトコル • RFC 8446 The Transport Layer Security (TLS) Protocol Version 1.3 [13] ※4.1.1 Cryptographic Negotiation により後方互換性を提供するが、混成方式ではない	通信プロトコル • draft-ietf-tls-hybrid-design-16 [14] • draft-ietf-mls-combiner-02 [15] • ETSI TS 103 744 [16] • IEEE 802.11 Hybrid PQC Proposal [17] • draft-ietf-lamps-pq-composite-kem-12 [11][4]
	証明書構造 • draft-ietf-lamps-certdiscovery-02 [18] • ITU-T X.509(9.8 Alternative cryptographic algorithms and digital signature extensions) [19]	証明書構造 • draft-ietf-lamps-pq-composite-sigs-14 [12][4]
システム[5]	（後方互換性確保／安全性維持　共通） • 耐量子移行を考慮した PKI ベースのハイブリッド設計・実装 Architecting PKI Hierarchies for Graceful PQ Migration(PQC Conference) [20] Hybrid PQC E-Mail Communication: Easing Migration Pain(PQC Conference) [21]	

4.2 ハイブリッド構成のアルゴリズム

[3] 本文献は、複数鍵素材を同時依存で統合する汎用 KDF を規定する仕様であり、ハイブリッド構成を直接目的とした設計ではないが、draft-irtf-cfrg-hybrid-kems-07 や draft-ietf-lamps-pq-composite-kem-12 における鍵結合の基盤技術として位置付けられる。

[4] 本文献は、複合方式による暗号アルゴリズム仕様と、証明書・プロトコルでの利用方法を併せて規定しているため、アルゴリズム層およびプロトコル層の双方に記載している。

[5]システム層については、アルゴリズム層やプロトコル層のように単一の技術仕様として規定されたハイブリッド方式は存在しない。このため本表では、複数の技術仕様を組み合わせて構成される実運用上の設計例や実装案を通じて、システムとしてのハイブリッド構成を整理している。また、システム層では、利用目的や運用段階に応じて、後方互換性確保を目的とする構成と安全性維持を目的とする構成を選択または組み合わせて用いることが想定される。

4.2.1 鍵共有アルゴリズム

ハイブリッド構成における鍵共有アルゴリズムは、複数の暗号方式を組み合わせることで安全性を強化し、耐量子計算機性を確保することを目的として設計されている。

本節では、まず IETF が前提とする鍵結合モデルの基礎を与える文書、NIST SP 800-227 を取り上げる。NIST SP 800-227 では、Multi-Algorithm KEM の設計指針が示されており、耐量子計算機性を有する（post quantum）方式と既存（traditional）の方式を組み合わせる post quantum/traditional ハイブリッド構成（以下、PQ/T ハイブリッド構成と呼ぶ）の基本的な考え方を理解する上で参照すべき文書である。なお、NIST SP 800-227 では Section 4.6「Multi-Algorithm KEMs」の中で、複数 KEM を結合した方式を Composite KEM と呼んでいる。本節では、同セクションタイトルに合わせて「Multi-Algorithm KEM」を包括的な用語として使用する。

ハイブリッド構成の Multi-Algorithm KEM は、以下の二段階で構成される。

1. 共有秘密を生成する。
   複数の KEM を用いた鍵共有を個別に実行し、方式ごとに共有秘密を生成する。これにより、アリスとボブは複数の共有秘密を保持する。
2. 共有秘密を結合する。
   生成された複数の共有秘密を、承認済みの鍵結合器（Key Combiner）を用いて単一の共有秘密鍵を導出する。

図 4-1 に示すように、鍵生成、カプセル化（Encapsulation）、デカプセル化（Decapsulation）の各フェーズで複数の KEM を用いた鍵共有が並列に動作し、最終的に Key Combine 関数によって単一の共有鍵が導出される。

なお、図 4-1 は NIST SP 800-227 の Fig. 1. Outline of key establishment using a KEM に Multi-Algorithm KEM の構築手順を反映した図である。

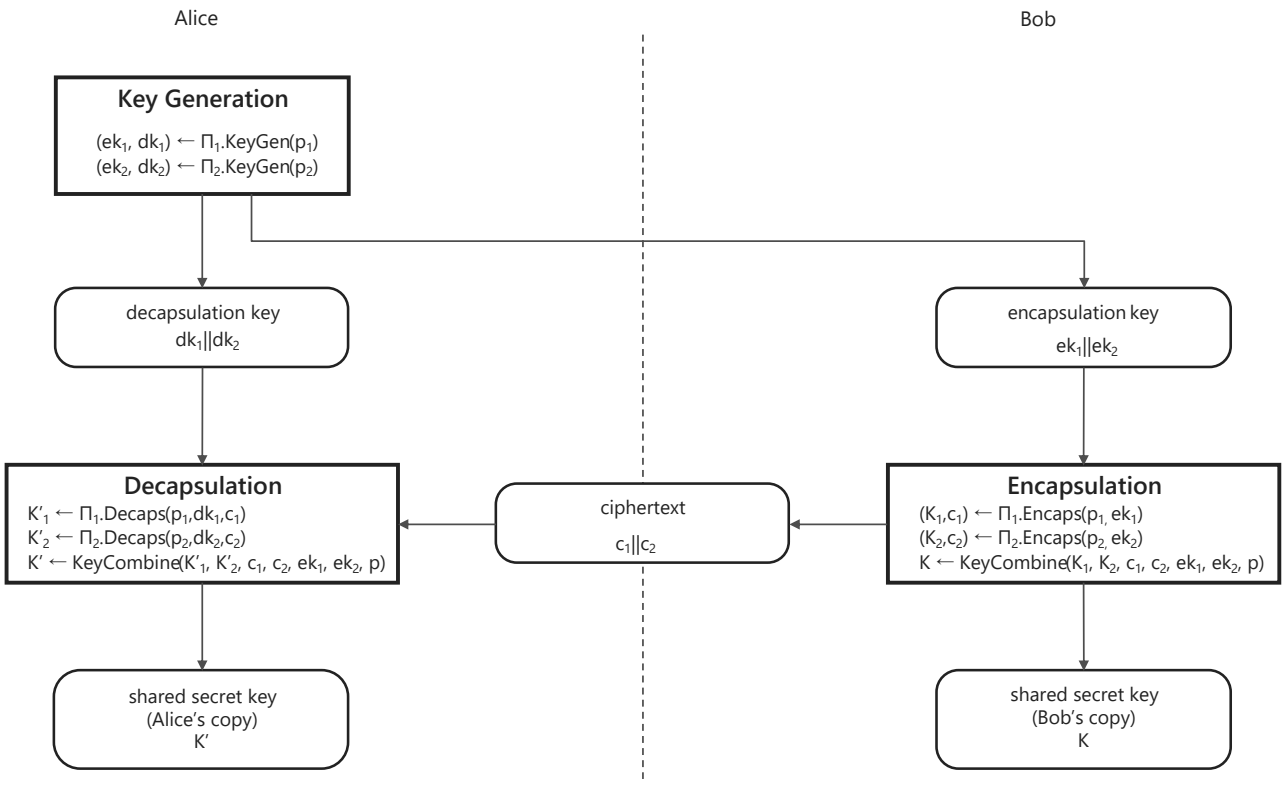


図 4-1 Multi-Algorithm KEM の概要

表 4-1 のアルゴリズム層に記載するその他の標準化文書との関係性は次の通りである。

draft-irtf-cfrg-hybrid-kems-07 は、ハイブリッド鍵カプセル化方式（ハイブリッド KEM）の設計フレームワークを定義しており、複数の KEM を組み合わせる際のセキュリティモデルや鍵結合器の要件を明確化している。具体的には、ML-KEM などの耐量子計算機性を有する KEM と従来の DH ベースの KEM を組み合わせる構成を想定している。本仕様では、DH を KEM と同じ入出力形式に抽象化して扱う枠組みを導入し、DH と PQCKEM を共通の KEM モデルに揃えた上でハイブリッド構成を定義している。

draft-ietf-lamps-pq-composite-kem-12 は、X.509 証明書や CMS などの PKI 環境において、ハイブリッド KEM を定義しており、本仕様ではコンポジット KEM という名称で規定されている。すなわち、複数の鍵共有方式（例：ML-KEM + ECDH）を一つの構造体にまとめることで、証明書や鍵管理の互換性を維持しつつハイブリッド構成を実現する。特に、証明書内で複数の公開鍵を保持し、暗号操作時に両方の鍵共有方式を実行する仕組みを提供する点で、NIST SP 800-227 の Multi-Algorithm KEM の概念を実装レベルに落とし込んでいる。なお、本仕様は証明書に格納される static 公開鍵を対象としている。

また、本仕様では表 4-2 のアルゴリズムリストが規定されている。

表 4-2 コンポジット KEM のアルゴリズムリスト

OID Name	OID	ML-KEM Variant	Traditional Algorithm	Key Size / Curve
id-MLKEM768-RSA2048-SHA3-256	1.3.6.1.5.5.7.6.55	ML-KEM-768	RSA	2048
id-MLKEM768-RSA3072-SHA3-256	1.3.6.1.5.5.7.6.56	ML-KEM-768	RSA	3072
id-MLKEM768-RSA4096-SHA3-256	1.3.6.1.5.5.7.6.57	ML-KEM-768	RSA	4096
id-MLKEM768-X25519-SHA3-256	1.3.6.1.5.5.7.6.58	ML-KEM-768	X25519	X25519
id-MLKEM768-ECDH-P256-SHA3-256	1.3.6.1.5.5.7.6.59	ML-KEM-768	ECDH	secp256r1
id-MLKEM768-ECDH-P384-SHA3-256	1.3.6.1.5.5.7.6.60	ML-KEM-768	ECDH	secp384r1
id-MLKEM768-ECDH-brainpoolP256r1-SHA3-256	1.3.6.1.5.5.7.6.61	ML-KEM-768	ECDH	brainpoolP256r1
id-MLKEM1024-RSA3072-SHA3-256	1.3.6.1.5.5.7.6.62	ML-KEM-1024	RSA	3072
id-MLKEM1024-ECDH-P384-SHA3-256	1.3.6.1.5.5.7.6.63	ML-KEM-1024	ECDH	secp384r1
id-MLKEM1024-ECDH-brainpoolP384r1-SHA3-256	1.3.6.1.5.5.7.6.64	ML-KEM-1024	ECDH	brainpoolP384r1

id-MLKEM1024-X448-SHA3-256	1.3.6.1.5.5.7.6.65	ML-KEM-1024	X448	X448
id-MLKEM1024-ECDH-P521-SHA3-256	1.3.6.1.5.5.7.6.66	ML-KEM-1024	ECDH	secp521r1

4.2.2 署名アルゴリズム

本節では、IETF LAMPS WG において標準化が進められているコンポジット署名方式（draft-ietf-lamps-pq-composite-sigs-14）について説明する。本方式は、耐量子署名アルゴリズムと従来署名アルゴリズムを一つの署名構造に統合することを目的として設計されている。コンポジット署名は、証明書やメッセージ署名において両方の署名を同時に生成・検証する仕組みを提供し、既存の PKI 基盤との整合性を維持するための不可欠な技術である。

図 4-2 は、コンポジット署名方式における鍵生成の手順を示す。コンポジット署名では、耐量子署名アルゴリズム（例：ML-DSA）と従来署名アルゴリズム（例：RSA ベースの署名方式や ECDSA）の鍵ペアをそれぞれ生成し、両者を一つの構造に統合する。この統合により、証明書や署名操作において両方の鍵を同時に利用可能となる。

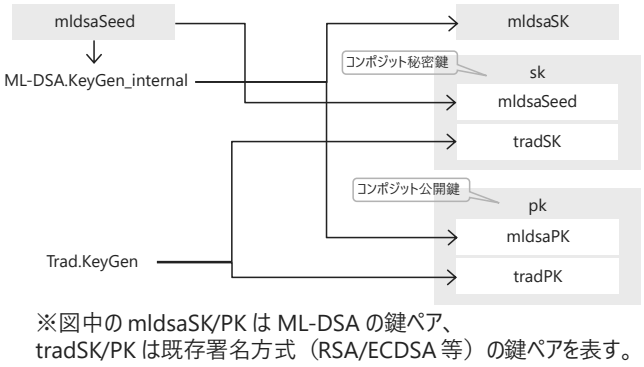


図 4-2 コンポジット鍵生成の概要

図 4-3 は、コンポジット署名の生成手順を示す。署名対象データに対してハッシュ関数による前処理を行い、その結果を両方の署名アルゴリズムに入力する。各アルゴリズムは独立に署名を生成し、最終的にコンポーネント署名構造に統合する。この構造には署名値、アルゴリズム識別子、関連パラメータが含まれ、検証者が両方の署名を確認できるよう設計されている。

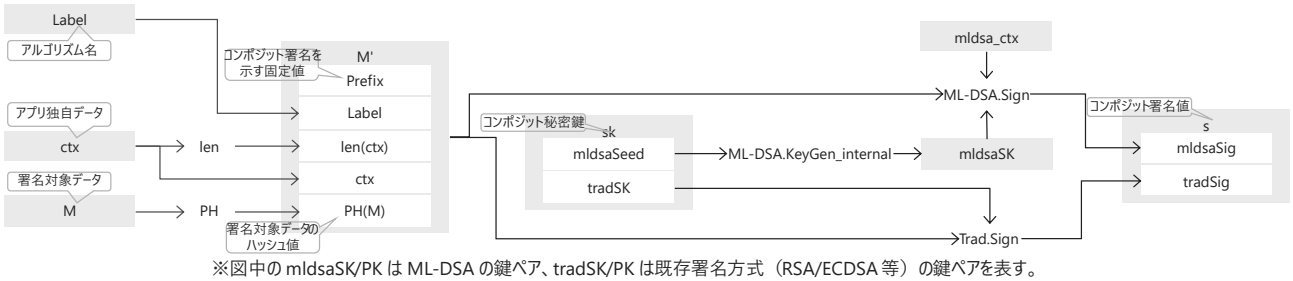


図 4-3 コンポジット署名の概要

図 4-4 は、コンポジット署名の検証手順を示すものである。検証者は署名構造から各署名値と対

応する公開鍵を取得し、両方のアルゴリズムで検証を実施する。検証処理は、コンポジット署名構造に含まれるアルゴリズム識別子とパラメータに基づいて行われ、証明書の整合性も確認対象となる。

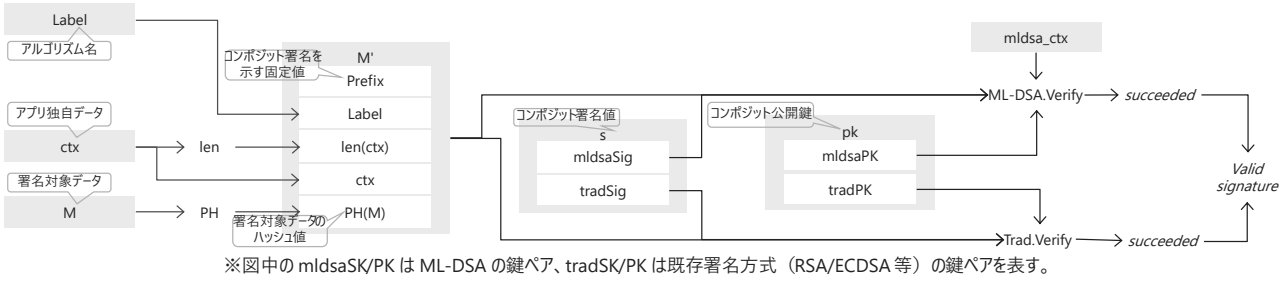


図 4-4 コンポジット署名検証の概要

また、表 4-3 のアルゴリズムリストが規定されている。

表 4-3 コンポジット署名のアルゴリズムリスト

OID Name	OID	Pre-Hash Function	ML-DSA Variant	Traditional Algorithm	Key Size / Curve
id-MLDSA44-RSA2048-PSS-SHA256	1.3.6.1.5.5.7.6.37	SHA256	ML-DSA-44	RSA	2048
id-MLDSA44-RSA2048-PKCS15-SHA256	1.3.6.1.5.5.7.6.38	SHA256	ML-DSA-44	RSA	2048
id-MLDSA44-Ed25519-SHA512	1.3.6.1.5.5.7.6.39	SHA512	ML-DSA-44	EdDSA	Ed25519
id-MLDSA44-ECDSA-P256-SHA256	1.3.6.1.5.5.7.6.40	SHA256	ML-DSA-44	ECDSA	secp256r1
id-MLDSA65-RSA3072-PSS-SHA512	1.3.6.1.5.5.7.6.41	SHA512	ML-DSA-65	RSA	3072
id-MLDSA65-RSA3072-PKCS15-SHA512	1.3.6.1.5.5.7.6.42	SHA512	ML-DSA-65	RSA	3072
id-MLDSA65-RSA4096-PSS-SHA512	1.3.6.1.5.5.7.6.43	SHA512	ML-DSA-65	RSA	4096
id-MLDSA65-RSA4096-PKCS15-SHA512	1.3.6.1.5.5.7.6.44	SHA512	ML-DSA-65	RSA	4096
id-MLDSA65-ECDSA-P256-SHA512	1.3.6.1.5.5.7.6.45	SHA512	ML-DSA-65	ECDSA	secp256r1
id-MLDSA65-ECDSA-P384-SHA512	1.3.6.1.5.5.7.6.46	SHA512	ML-DSA-65	ECDSA	secp384r1
id-MLDSA65-ECDSA-brainpoolP256r1-SHA512	1.3.6.1.5.5.7.6.47	SHA512	ML-DSA-65	ECDSA	brainpoolP256r1
id-MLDSA65-Ed25519-	1.3.6.1.5.5.7.6.48	SHA512	ML-DSA-65	EdDSA	Ed25519

SHA512					
id-MLDSA87-ECDSA-P384-SHA512	1.3.6.1.5.5.7.6.49	SHA512	ML-DSA-87	ECDSA	secp384r1
id-MLDSA87-ECDSA-brainpoolP384r1-SHA512	1.3.6.1.5.5.7.6.50	SHA512	ML-DSA-87	ECDSA	brainpoolP384r1
id-MLDSA87-Ed448-SHAKE256	1.3.6.1.5.5.7.6.51	SHAKE256	ML-DSA-87	EdDSA	Ed448
id-MLDSA87-RSA3072-PSS-SHA512	1.3.6.1.5.5.7.6.52	SHA512	ML-DSA-87	RSA	3072
id-MLDSA87-RSA4096-PSS-SHA512	1.3.6.1.5.5.7.6.53	SHA512	ML-DSA-87	RSA	4096
id-MLDSA87-ECDSA-P521-SHA512	1.3.6.1.5.5.7.6.54	SHA512	ML-DSA-87	ECDSA	secp521r1

4.3 ハイブリッド構成のプロトコル

4.3.1 通信プロトコル

本節では、通信プロトコルにおけるハイブリッド鍵共有の代表例として TLS 1.3 の構成を説明する。他にも、MLS における鍵素材の結合方式（draft-ietf-mls-combiner-02）、ETSI TS 103 744 にけるハイブリッド鍵共有方式、IEEE 802.11 におけるハイブリッド鍵共有の検討などが存在するが、本報告書では TLS 1.3 の事例に焦点を当てる。

TLS 1.3 のハイブリッド鍵共有は、ECDHE と KEM を用いた鍵共有とを独立に実行し、その結果を鍵導出で結合する合成(composite)方式に該当する。複数アルゴリズムの公開鍵や暗号文を送信する際に 連結アプローチが採用される。具体的には以下の通りである。

- KeyShareEntry.key_exchange フィールドに、構成アルゴリズムの key_exchange 値を連結して格納する。
- 追加の符号化や長さフィールドは不要で、アルゴリズムが固定されれば長さも固定される。
- NamedGroup がハイブリッド（例：MyECDHMyPQKEM）の場合、key_exchange は MyECDH.KeyGen()と MyPQKEM.KeyGen()の結果を連結したものになる。

```
例:
MyECDHMyPQKEM.KeyGen() = (MyECDH.KeyGen(), MyPQKEM.KeyGen())
KeyShareEntry {
    NamedGroup: MyECDHMyPQKEM,
    key_exchange: MyECDHMyPQKEM.KeyGen()
}
```

図 4-5 は、TLS 1.3 におけるハイブリッド鍵共有の処理手順を示すものである。ハイブリッド鍵共有では、従来の ECDHE と耐量子計算機性を有する KEM を用いた鍵共有(ephemeral)とを組み合わせ、複数の鍵素材を連結して利用する構成を採用する。クライアントおよびサーバは、それぞれ ECDHE と KEM を用いた鍵共有により 2 つの鍵を生成し、ClientHello メッセージにおいて公開鍵を送信する。サーバは受信した公開鍵に基づき、ECDHE による共有秘密と KEM によるカプセル化処理

を実行し、ServerHello メッセージでカプセル化データを返送する。

その後、クライアントはカプセル化データを復号し、ECDHE と KEM の結果を連結してハイブリッド共有秘密(ss_hybrid)を生成する。この共有秘密は HKDF-Extract に入力され、ハンドシェイク用トラフィック秘密(handshake_secret)が導出され、client_handshake_traffic_secret（クライアント→サーバ方向のハンドシェイクメッセージを暗号化する鍵）および server_handshake_traffic_secret（サーバ→クライアント方向のハンドシェイクメッセージを暗号化する鍵）が生成される。以降の EncryptedExtensions、Certificate、CertificateVerify、Finished メッセージは、この秘密値に基づいて暗号化される。

図 4-5 は、鍵生成、カプセル化、デカプセル化、HMAC ベース鍵導出関数（HMAC-based Key Derivation Function: HKDF）による鍵生成、ハンドシェイクメッセージの暗号化までの一連の流れを視覚的に示している。

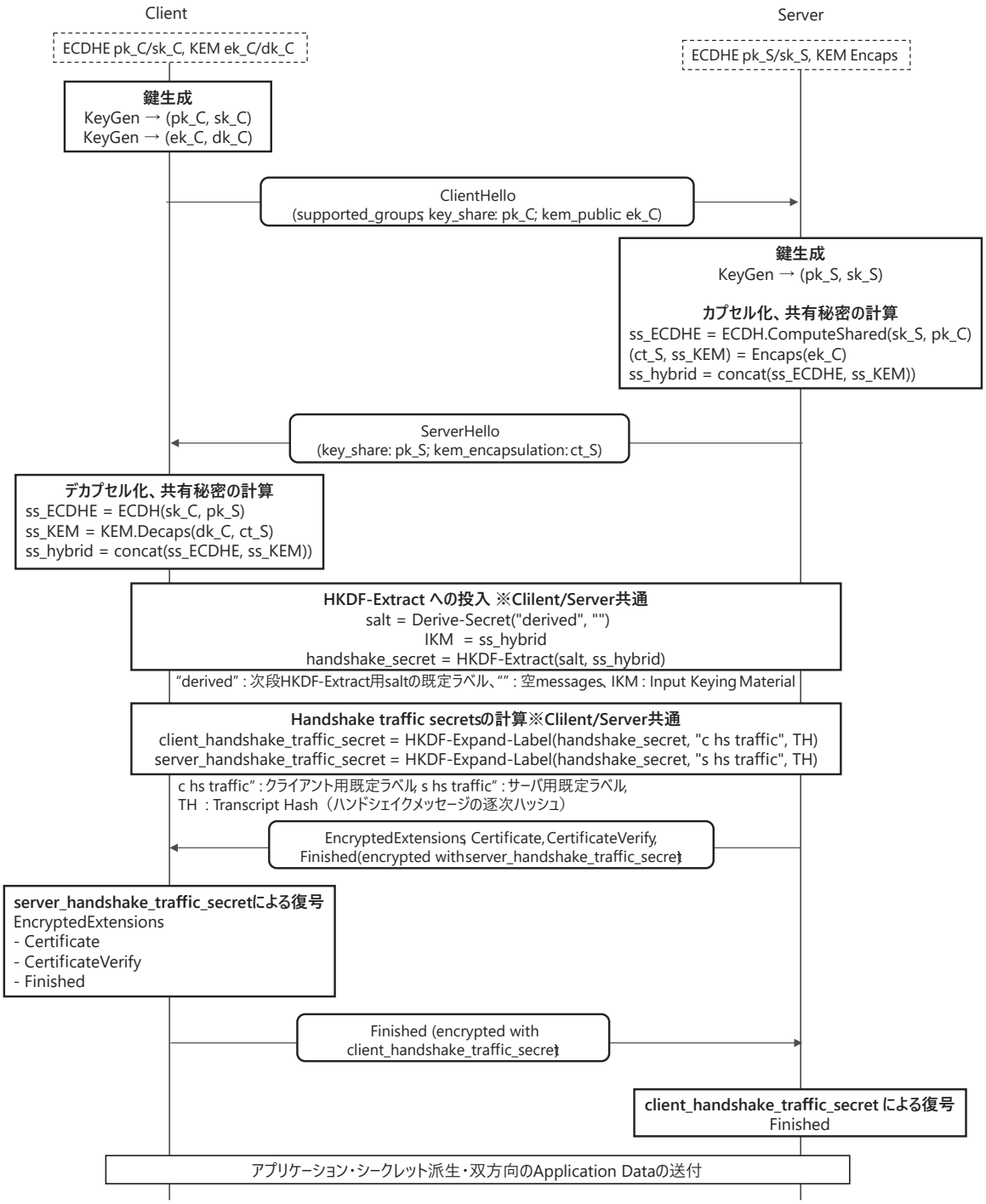


図 4-5 TLS 1.3 におけるハイブリッド鍵共有のシーケンス概要

4.3.2 証明書構造

本節では、ハイブリッド構成に対応した証明書構造の代表的な例を示し、それぞれの特徴を説明する。

図 4-6 は、ハイブリッド構成対応の証明書構造を示すものであり、後方互換性確保を目的とする方式と、安全性維持を目的とする方式の両方を含んでいる。なお、図中の名称（Discovery-enabled certificate、Dual-signature certificate、Composite-signature certificate）は、本報告書において説明の便宜上付与したものである。ハイブリッド構成の証明書構造には、標準化文書で定義された複数のアプローチが存在し、ここでは代表的な三つの例を示す。

1. Discovery-enabled certificate（draft-ietf-lamps-certdiscovery-02）
   本方式は、複数の証明書を並列に運用し、それらを証明書発見・選択可能とする構造である。Extensions フィールドに Subject Information Access を追加し、accessMethod および accessLocation により、関連する別の証明書(例：PQC または既存暗号)への参照情報を提供する。この方式では、既存の公開鍵暗号方式による証明書だけでも検証を継続できるため、2.1 節で定義する混成（hybrid）方式に該当する。
2. Dual-signature certificate（ITU-T X.509 Alternative cryptographic algorithms）
   本方式は、既存暗号の署名と PQ の署名をそれぞれ独立に保持する証明書構造である。Extensions フィールドに subjectAltSigAlg および subjectAltPubKey を追加し、追加の署名アルゴリズムと公開鍵を格納する。検証者は、従来署名のみを利用することも、PQC 署名を利用することも可能であり、移行期においてどちらの署名方式でも検証可能となる。このため、本方式は、2.1 節で定義する混成（hybrid）方式に該当する。
3. Composite-signature certificate（draft-ietf-lamps-pq-composite-sigs-14）
   本方式では、複数の署名アルゴリズム(既存暗号と PQC)を一体の署名構造として統合する。証明書には複数の公開鍵をまとめて保持し、署名アルゴリズムは

   CertSigAlg = MLDSA44-ECDSA-P256-SHA256

   のように、連結された単一の AlgorithmIdentifier として扱われる。この構成は、署名検証時にすべての構成署名が正しく検証される必要があるという前提のため、既存署名単独では利用できず、後方互換性を持たない。したがって本方式は2.1節で定義する合成(composite)方式に該当する。

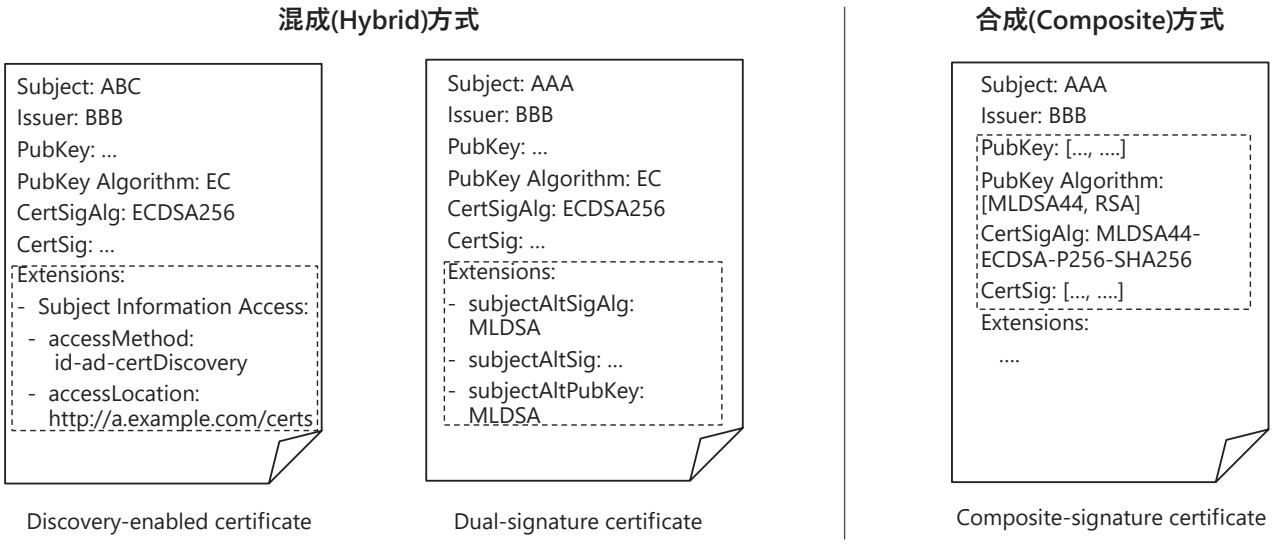


図 4-6 ハイブリッド構成における証明書構造の例

4.4 ハイブリッド構成のシステム

ハイブリッド構成は、アルゴリズムおよびプロトコルの設計のみならず、PKI 階層や証明書運用を含むシステムレベルの構成にも影響を及ぼす。近年では、PQC 移行を考慮した PKI 構成や電子メールシステムにおける段階的移行方式が検討されており、これらは実装・運用の観点から重要な論点となる。具体例については 6 章において詳述する。

5. ハイブリッド構成の安全性に関する解説

本章では、個別のアルゴリズム仕様やプロトコルの詳細な解説は行わず、標準化文書に記述されている内容を基に、ハイブリッド構成における安全性の整理を行う。特に、安全性がどのような条件で成立すると整理されているか、ならびにそれらの条件が鍵共有構成や合成構造、運用上の取扱いの中でどの構成要素に依存して具体化されているかに着目する。

5.1 脅威と移行前提

PQC への移行を扱う文書では、従来の公開鍵暗号が前提としてきた現在の計算機による攻撃に加え、将来の量子コンピュータの利用を想定した攻撃が整理されている。NIST IR 8547 [22]では、PQC への移行は段階的に進行するものであり、移行期間中は複数の暗号アルゴリズムが併存する状況が想定されている。

NIST SP 800-227 Section 4.6 [4]では、このような移行前提の下で、複数の鍵共有アルゴリズムを組み合わせるハイブリッド鍵共有アルゴリズム構成が示されている。これらの記述は、PQC への移行を一時点で完了することを前提としない構成が想定されていることを示している。

5.2 ハイブリッド構成に求められる性質

5.2.1 ハイブリッド構成の安全性の性質

RFC 9794 [23]では、既存の公開鍵暗号方式と耐量子計算機性を持つ暗号方式を組み合わせたハイブリッド構成に期待される性質として、秘匿性、認証、相互接続性、後方互換性、前方互換性を挙げている。以下、それぞれの性質について簡単に紹介する。

秘匿性（PQ/T hybrid confidentiality）：
秘匿性を実現するための構成要素である暗号方式のうち少なくとも 1 つ（たとえば ML-KEM）が安全であれば、ハイブリッド構成の秘匿性が実現されるという性質。

認証（PQ/T hybrid authentication）：
認証を実現するための構成要素である暗号方式の少なくとも 1 つ（たとえば ML-DSA）が安全であれば、ハイブリッド構成の認証が実現されるという性質。

相互接続性（PQ/T hybrid interoperability）：
当事者双方が、構成要素である暗号方式のうち少なくとも一つを共通にサポートしていれば、ハイブリッド構成のプロトコルを成功裏に完了できるという性質。

後方互換性（PQ/T backwards compatibility）：
当事者双方が既存の公開鍵暗号方式をサポートしていれば、ハイブリッド構成のプロトコルを成功裏に完了できるという性質。 [23]ではさらに、当事者双方が既存の公開鍵暗号方式と PQC 方式の両方をサポートしている場合には、その両方を用いることを要求している。

前方互換性（PQ/T hybrid forward compatibility）：
当事者双方が同じ PQC 方式の構成要素をサポートしている場合には、PQC 方式を用いて PQ/T ハイブリッド構成のプロトコルを成功裏に完了できるという性質。 [23]ではさらに、当事者双方が

既存の公開鍵暗号方式と PQC 方式の両方をサポートしている場合には、その両方を用いる選択肢を持つことを要求している。

上に挙げた性質のいくつかは互いに排他的であり、すべての性質を満たすハイブリッド構成を実現することはできない。例えば、相互接続性、後方互換性、前方互換性はいずれも相反する性質である。また、これらのいずれかが成立する場合、PQ/T ハイブリッド秘匿性（もしくは認証）を満たさない可能性がある。

なお、上記の性質の秘匿性と認証について、個別の暗号方式に対しては、それぞれ代表的な安全性を前提としている。鍵共有方式に対しては、IND-CCA（Indistinguishable under Chosen Ciphertext Attack）安全性が一般に設計上の前提条件として参照される一方、電子署名方式に対しては、EUF-CMA（Existential Unforgeability under Chosen Message Attack）や SUF-CMA（Strong Existential Unforgeability under Chosen Message Attack）といった偽造困難性に関する安全性定義が用いられている。本節では、個々の方式の安全性には立ち入らず、ハイブリッド構成における安全性成立条件との対応関係を整理する。

5.2.2 ハイブリッド署名と分離困難性

ハイブリッド署名構成の基本的なアイデアは、署名において既存の公開鍵暗号の署名方式による署名値と PQC 署名方式による署名値を連結し、検証において 2 つの署名値がいずれも正しい場合に正当なハイブリッド署名と判定する。ハイブリッド構成の署名は、従来の署名に求められる安全性を満たす必要があるが、ハイブリッド署名特有の安全性要件として分離困難性（non-separability）が挙げられる。

簡単なハイブリット署名の例として、メッセージ M に対して既存の公開鍵暗号による署名 $sig_T(M)$ と PQC 署名 $sig_{PQ}(M)$ を並べた（M, $sig_T(M)$, $sig_{PQ}(M)$）を考える。このとき、(M, $sig_T(M)$)は既存の公開鍵暗号のみをサポートするシステムにおいて正当な署名である。このような、ハイブリッド署名の一部を抜き出し、新たに正当なメッセージと署名の組を作る攻撃を stripping attack と呼ぶ。stripping attack は cross protocol attack の一種である。また、ハイブリッド署名から既存（もしくは PQC）の署名単体を取り出して使うので、ダウングレード攻撃の一種とみなすこともできる。

IETF は [24]の中で、stripping attack と、stripping attack に対する安全性として分離困難性（non-separability）について紹介している。分離困難性は、弱分離困難性と強分離困難性に分けられる。

弱分離困難性（weak non-separability）：
攻撃者が既存の署名もしくは PQC の署名のいずれかを痕跡を残さずに取り除くことはできないという性質。

強分離困難性（strong non-separability）：
攻撃者がメッセージとハイブリッド署名の組から、正しく検証に通る構成要素の署名を出力することができないという性質。

また、[24]では、強分離困難性よりも強い安全性として同時検証可能性を紹介している。

同時検証（simultaneous verification）：
ハイブリッド構成のすべての要素について検証が終わらない限り、ハイブリッド署名としての検証が完了しないという性質。

同時検証は、故障利用攻撃などの方法で検証の一部をスキップさせるような攻撃者に対する安全性を保証する。

分離困難性を実現する手段として、artifact と呼ばれる付加的な情報が用いられる。artifact は署名や公開鍵証明書など処理の各レイヤーに埋め込まれる情報である。Hybrid Signature Spectrums で定義される弱分離困難性と強分離困難性が、draft-ietf-lamps-pq-composite-sigs-14 では以下のとおり実現されている。

- 弱分離困難性
  draft-ietf-lamps-pq-composite-sigs-14 では、署名対象 M から構成される M′ (Section 2.2 の Prefix・Label・ctx を含むデータ) を ML-DSA と既存方式の双方に渡して署名を生成する。なお、ctx (context string) は、アプリケーション固有の文脈を示す補助データである。攻撃者が Composite ML-DSA 署名(M, (mldsaSig, tradSig))を分割しても、既存署名側には Prefix が静的に残るため、署名が composite 由来である痕跡が消えない。一方 ML-DSA 側では、ctx が Composite Algorithm の Label に設定されているため、ctx=""の通常の ML-DSA.Verify では検証に失敗する。
  この動作により、ML-DSA／従来方式いずれの署名も、単独署名として独立に再利用することができず、Hybrid Signature Spectrums が定義する弱分離困難性を満たしている。
- 強分離困難性
  draft-ietf-lamps-pq-composite-sigs-14 では、攻撃者が composite 署名から片側の署名を取り出し、異なるメッセージに対してその署名を“単独署名として”再利用し、対応する verifier に受理させることを難しくする仕組みが備わっている。ML-DSA 側は前述の ctx (=Composite Label) により ML-DSA 単独検証では成功しないため、限定的とはいえ強分離困難性を満たす。
  さらに X.509 では、署名対象に署名アルゴリズムの Label (＝Composite であることを示す識別子) が含まれるため、片側署名のみを残しても X.509 の検証処理で「Composite として署名されているはず」と判断され、検証が失敗する。
  また、draft-ietf-lamps-pq-composite-sigs-14 の Section 9.3 で規定される“composite と単独署名の文脈で鍵を使い回さない”という要件により、強分離困難性が実運用上さらに強化されることとなる。以上により、draft-ietf-lamps-pq-composite-sigs-14 は Hybrid Signature Spectrums の強分離困難性に対応する仕組みを備えている。

5.3 ハイブリッド鍵共有と前方／後方互換性

既存の公開鍵暗号による鍵共有プロトコルでは、DH 鍵共有などを用いて当事者間で共有秘密 (shared secret) を共有し、共有秘密と付加情報を鍵導出関数 (Key Derivation Function: KDF) に入力して通信の暗号化等に用いる鍵素材 (keying material もしくは derived keying material) を生成する。ハイブリッド鍵共有では、既存の公開鍵暗号による鍵共有を用いて共有された共有秘密 ss_T と、PQC の KEM を用いて共有された共有秘密 ss_{PQ} の両方を KDF に入力する。ハイブリッド構成を想定した KDF の使い方を鍵結合器 (key combiner) と呼ぶ。

5.3.1 NIST SP 800-56C Rev. [2]の KDF とハイブリッド KEM

NIST SP 800-56C Rev.2 [9]では、鍵共有処理により得られる共有秘密 Z から、鍵生成関数 (KDF) を用いて暗号処理に用いる鍵素材 (Derived Keying Material) を生成する手順が定義されている。

従来、共有秘密はNIST SP 800-56A, Bで共有された値を指していたが、NIST SP 800-56C Rev.2において、上記の共有秘密Zと「その他の方法」で共有された補助的な共有秘密Tを連結したZ’=Z||Tを共有秘密としてKDFの入力とする方法を規定した。この「その他の方法」としてPQC方式のKEMを用いることで、PQ/TハイブリッドKEM（鍵共有）を構成することができる。

5.3.2 NIST SP 800-227と鍵結合器

NIST SP 800-227 [4]は、ハイブリッドKEMを規定する文書であり、その概要は4.2.1節で説明した通りである。[4]第5章では、ECDHなど既存の鍵共有方法からKEMを構成する方法を与えている。また、 [4]4.6.3節において、複数のKEMで得た共有秘密から鍵素材を生成する鍵結合器を規定した。二つのKEM KEM1, KEM2の共有秘密をK1, K2とするとき、鍵結合器Key_Combineは

K ← Key_Combine(K1, K2)

で与えられる。また、鍵結合器はKEM1, KEM2のパラメータp1, p2、公開鍵をek1, ek2、暗号文をc1, c2などを補助入力として入力しても良い。Key_Combineを実現する方法として、NIST SP 800-56C Rev.2に記載の鍵導出方法（Key Derivation Method: KDM）およびNIST SP 800-133に記載の鍵導出関数が挙げられている。[4]はKDMやKDFに複数の共有秘密を入力する具体的なエンコード方法を規定していないが、一例として共有秘密をK=K1||K2と連結する方法を挙げている。

5.3.3 ハイブリッド鍵共有と前方／後方互換性

図 5-1は、NISTが2024年3月に公開した講演資料 [25]からの引用である。この図のように、鍵合成器を使うハイブリッド鍵共有はすべての方式で共通であるが、その考え方にはいくつかのバリエーションが存在する。

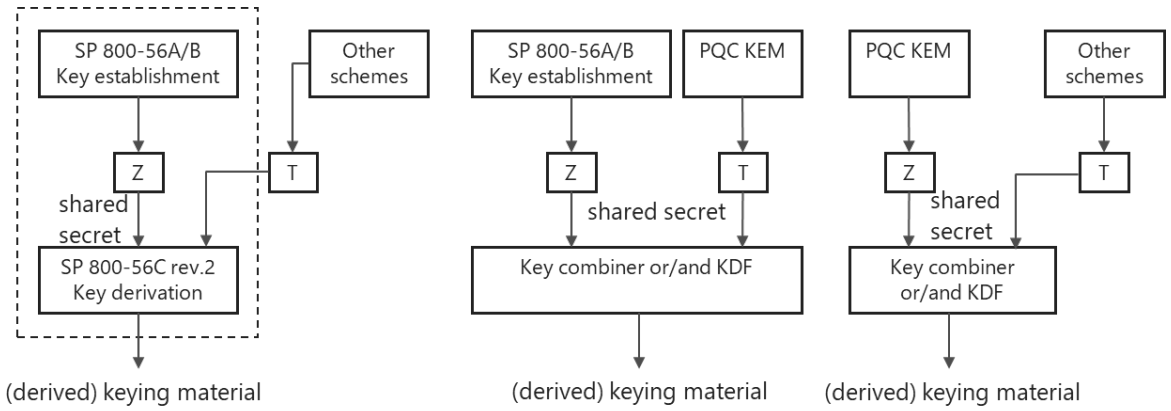


図 5-1 PQC暗号移行とKEMのハイブリッド構成［25］

まず、PQC方式のKEMや、KEMを用いた鍵共有プロトコルが標準化されていないPQC暗号移行の初期段階では、NIST SP 800-56C Rev.2の方法を用いる。すなわち、PQC方式を「その他の鍵共有方法」として、既存の標準の枠組みの中でハイブリット鍵共有を実現する（図左）。この段階では既存の公開鍵暗号が主であり、PQC方式は補助的な役割である。したがって、PQC方式の共有秘密が用いられない場合があり得る。すなわち、この方式は後方互換性を満たす設定を許すが、その場合にはPQ/TハイブリッドKEM秘匿性を満たさない。

図中央はPQC暗号移行の中間的な段階であり、NIST SP 800-227がこれに該当する。この段階では、既存の公開鍵暗号とPQC方式が同格として扱われており、鍵素材の生成に両方の共有秘密を使うことが必須となる。したがって、この方式はPQ/TハイブリッドKEM秘匿性を満たすが、前方／後方

互換性のいずれも満たさない。

最後に、図右はPQC暗号移行の最終段階であり、PQC方式が主となり、補助的に「その他の鍵共有方法」を用いる。この方式は、前方互換性を満たす使用が可能であり、その場合にはPQ/Tハイブリッド秘匿性を満たさない。

6. ハイブリッド構成の実装・運用に関する解説

本章では、PQC およびハイブリッド構成の実装に関する主要な OSS の整備状況と、実運用環境における実装・移行事例を紹介する。暗号ライブラリ、PKI、メール、ブラウザなど多様な事例で得られた知見を通じ、移行期におけるハイブリッド構成の位置付けと実務上の課題を把握する。

6.1 実装基盤としての OSS 動向

PQC および既存暗号方式とのハイブリッド構成を実運用へ導入するにあたり、標準仕様の策定と並行して、実装基盤となる暗号ライブラリの整備状況を把握することが重要である。本節では、ハイブリッド構成の実装・検証において利用されている主要なオープンソースソフトウェア（OSS）として、Open Quantum Safe（OQS）プロジェクト [26]およびその成果物、ならびに主要な暗号ライブラリの動向を整理する。

OQS プロジェクトは、PQC アルゴリズムの実装およびそれらを既存プロトコルやライブラリへ統合するための検証基盤を提供することを目的とした OSS プロジェクトである。OQS プロジェクトの中核となる liboqs [27]は、PQC KEM や署名アルゴリズムの実装を提供する暗号ライブラリであり、これらを他の暗号ライブラリやプロトコル実装から利用可能とする役割を担っている。また、OQS-OpenSSL [28]は OpenSSL に liboqs を統合した派生実装であり、TLS をはじめとする既存プロトコル上で PQC およびハイブリッド方式の挙動を検証するための実装基盤として位置付けられる。

表 6-1 暗号ライブラリと仕様の対応関係

凡例　○：実装対象、△：限定的に実装対象、—：非実装対象

仕様 / 名称	CRFG Hybrid KEMs	Composite KEM	Composite signature	TLS Hybrid
liboqs (PQC アルゴリズムの実装 (KEM・署名))	△[6]	—	—	—
OQS-OpenSSL (OpenSSL に liboqs を統合した派生実装)	—	△[7]	△[7]	○
Bouncy Castle (Java を中心とした暗号アルゴリズムおよびプロトコル実装)	—	○	○	—
wolfSSL (組込み・軽量環境向けの TLS／暗号ライブラリ実装)	△[8]	—	—	○

一方、OQS プロジェクト以外にも、用途や実装方針の異なる暗号ライブラリが存在する。Bouncy Castle [29]は Java を中心とした暗号アルゴリズムおよびプロトコル実装を提供するライブラリであり、特に PKI／X.509 環境における複合鍵（Composite KEM）や複合署名（Composite Signature）の実装を通じて、ハイブリッド構成の実装を進めている。また、wolfSSL [30]は組込み・軽量環境

6 CFRG Hybrid KEMs 構成における PQC KEM の実装要素として利用

7 検証目的での限定対応

8 TLS ハイブリッド鍵交換のように限定

を主な対象とするTLS／暗号ライブラリであり、TLS 1.3におけるハイブリッド鍵共有の実装を中心に、実装規模や性能制約を考慮した対応が行われている。

表 6-1は、これら主要な暗号ライブラリについて、ハイブリッド構成に関連する代表的な仕様に対する実装上の関与の度合いを整理したものである。本表における「○」「△」「－」は、それぞれ当該仕様が実装対象であるか、限定的・検証目的での対応にとどまるか、あるいは実装対象外であるかを示している。なお、ここで整理する対応関係は、正式な準拠宣言や仕様バージョンとの厳密な対応関係を示すものではなく、各暗号ライブラリが実装・検証の観点からどの仕様に関与しているかを俯瞰的に示すことを目的としている。

6.2 システム実装・運用の実例

本節では、PKI Consortium が主催する PQC 最新動向の国際的なカンファレンスである PQC Conference [31] [32]で報告された実装・運用事例の中から、PQCおよびハイブリッド構成を既存システムへ導入する過程で顕在化した課題と、それに対して採られた実装上・運用上の判断を紹介する。ここで紹介する事例は、単に標準仕様に準拠した実装を行うことを目的としたものではなく、既存システムとの互換性、性能や運用負荷、ベンダ対応状況といった現実的制約を踏まえ、段階的導入や暫定構成を採用している点に特徴がある。

また、Chromeは、PQCへの早期対応を目的としてハイブリッド鍵共有の導入を進めており、その実装過程ではプロトコル更新だけでなく、ネットワーク機器との互換性や段階的展開といった実運用上の課題が顕在化した。本節では、Chromeにおける実装の変遷とそこから得られた知見を紹介する。

6.2.1 実運用を想定した PQC 移行および性能評価の事例

Michiel Marcus (TNO), “Real-World Post-Quantum Migrations: Lessons Learned and Performance Results” ― OQS を用いた OpenSSL ベースの暫定的ハイブリッド構成 ― [33]

課題

既存アプリケーションにおける暗号処理はコード全体に分散して実装されており、RSAやECCといった特定アルゴリズムへの依存が強く、PQCへの単純な置換が困難であった。加えて、利用するベンダ製品がPQCに未対応であるという制約の下で移行検討を進める必要があった。

対応・工夫

暗号アルゴリズムを抽象化する設計へと段階的に改修し、crypto-agilityを確保した上で、PQCおよび既存暗号を組み合わせたハイブリッド構成を追加した。これにより、将来的なアルゴリズム変更にも対応可能な基盤を整備した。

運用上の判断

ベンダ製品のPQC対応が未成熟である状況を前提に、OQSを用いたOpenSSLベースのリバースプロキシを暫定的に配置し、アプリケーション本体を変更することなく評価・計測を可能とする構成を採用した。

知見

ハイブリッド構成による性能低下は当初想定より限定的であり、多くのケースで最大でも数十パーセント程度に留まることが確認された。一方で、暗号依存関係の整理と crypto-agility の確保が移行コストを大きく左右する重要な要因であることが明らかとなった。

6.2.2 Web PKI における段階的 PQC 導入の事例

Shane Kelly (DigiCert), “The Internet Is Ready for Some PQC Certificates” [34]

課題

Web PKI において証明書チェーン全体を純粋な PQC へ置き換えた場合、証明書サイズやハンドシェイク時の転送量が大幅に増加し、既存ブラウザやインターネット環境への影響が懸念された。

対応・工夫

PKI 全体を一括で置き換えるのではなく、エンドエンティティ(リーフ)証明書から段階的に PQC 署名を導入する構成を採用した。

運用上の判断

証明書の有効期間を短縮することで、失効や強制的な更替を伴わずにアルゴリズム移行を進められる運用モデルが提案された。

知見

ハイブリッド構成を用いることで、完全な PQC 化に伴うサイズ・性能影響を抑えつつ、実運用環境での検証と経験蓄積を同時に進められることが示された。

6.2.3 PKI 階層設計におけるハイブリッド活用の事例

Mike Ounsworth (Entrust), “Architecting PKI Hierarchies for Graceful PQ Migration” [20]

課題

PKI 階層ごとに求められるセキュリティ要件や性能要件が異なり、単一アルゴリズムによる統一が必ずしも合理的でなかった。

対応・工夫

複数証明書方式、Composite、代替公開鍵拡張など、用途に応じたハイブリッド手法を組み合わせる「ツールボックス型」の設計が採用された。

運用上の判断

TLS などの交渉型プロトコルと、S/MIME 等の非交渉型用途で異なる方式を使い分けることで、後方互換性と移行容易性の両立が図られた。

知見

ハイブリッドは単一の方式としてではなく、用途に応じた設計上の選択肢群として扱う必要があることが示された。

6.2.4 S/MIME 電子メールにおけるハイブリッド実装の事例

Jan Klaußner (Bundesdruckerei), “Hybrid PQC E-Mail Communication: Easing Migration Pain” [21]

課題

S/MIME 電子メールでは、既存クライアントが複数署名や複数証明書を前提としておらず、単純な並列方式ではユーザ体験や互換性に問題が生じた。

対応・工夫

Composite 方式や代替鍵方式を用いることで、証明書構造を大きく変えずに PQC と既存暗号を統

合する実装が試行された。

運用上の判断

メールクライアント側の変更を最小限に抑えるため、暗号ライブラリ層での対応を重視し、アプリケーション層の変更を回避した。

知見

ハイブリッド構成では、暗号方式の選択に加え、既存クライアントの期待するデータ構造との整合を考慮する必要があることが明らかとなった。

6.2.5 ブラウザにおけるハイブリッド実装の事例(Chrome)

背景

Chrome は 2023 年に X25519Kyber768 を TLS で導入し、耐量子計算機性の確保に向けたハイブリッド鍵共有の検証を開始した。これは TCP／QUIC 双方を対象とした初期展開であり、当時の Kyber は標準化前のドラフト段階であった。［35］

実装

Kyber の標準化後、Google は自社で管理する TLS 暗号ライブラリである BoringSSL に ML-KEM を実装し、TLS のハイブリッド KEM コードポイントを Kyber（0x6399）から ML-KEM（0x11EC）へ移行した。Chrome131 以降では ML-KEM への一本化が予定されている。［36］

互換性問題

ハイブリッド KEM により ClientHello のサイズが 1KB 以上増加し、通信経路上で TLS メッセージを検査・中継するネットワーク機器（middlebox）が大きなメッセージを処理できず動作不良が発生したことが報告されている。［35］

問題対応

こうした非互換を把握するため、Chrome は段階的展開を行い、ネットワーク経路機器との互換性を継続的に検証する運用方針を採った。［35］

知見

ブラウザのようにクライアント側が先行して PQC を導入すると、ネットワーク機器が追随できず互換性問題が顕在化する一方、段階的展開によりエコシステム全体の問題を早期に露出させ、改善を促進できることが示された。［35］

6.2.1 節から 6.2.4 節の事例に共通する点として、PQC 移行は単なる暗号アルゴリズムの置換ではなく、システム構成、アーキテクチャ、運用手順を含めた包括的な設計変更として捉える必要があることが示されている。さらに、6.2.5 節で示した Chrome の事例は、クライアント側が先行して PQC を導入した場合、ネットワーク機器との非互換が顕在化し得ること、そして段階的展開や検証体制がエコシステム全体の移行に重要な役割を果たすことを示している。これらの事例は総じて、ハイブリッド構成が、移行期間を現実的に支えるための実用的な選択肢として位置付けられていることを裏付けている。

7. PQC 移行に関わる標準化動向の調査結果

本章では、PQC 移行期におけるハイブリッド方式の取り扱いに着目し、国際的な標準化団体および関連組織における検討状況を整理する。調査対象の選定にあたっては、CRYPTREC が公表した「ハイブリッドモード[1]の技術動向調査」 [3]による調査報告書を参照しつつ、その後の標準化動向や産業界での議論を踏まえて補完を行い、2026 年 1 月時点で PQC 移行やハイブリッド方式に関して実質的な情報発信を行っている組織を対象としている。

具体的には、NIST、IETF、ETSI、ISO、IEEE といった標準仕様策定機関に加え、国家レベルの暗号利用方針を示す機関、産業分野における実務的整理を行う団体、ならびに研究成果や知見を共有する国際的なプロジェクトや連合体を含めて調査対象としている。本章で扱う組織には、狭義の国際標準策定機関に限らず、標準化活動を補完する立場の組織が含まれる。

調査では、「hybrid」「composite」「multi-algorithm」等の用語を手掛かりとして、各組織が公開する標準文書、技術仕様、ガイドライン等を対象に情報収集を行い、鍵生成、デジタル署名、証明書、通信プロトコルにおけるハイブリッド構成の設計方針や運用上の位置付けに着目した。7.1 節から 7.11 節にかけて各組織の動向を整理することで、PQC 移行期におけるハイブリッド方式の全体像を俯瞰的に把握することを目的とする。

表 7-1 調査対象組織の一覧

章・節番号	組織名	URL
7.1	National Institute of Standards and Technology (NIST)	https://www.nist.gov/
7.2	Internet Engineering Task Force (IETF)	https://www.ietf.org/
7.3	International Telecommunications Union (ITU)	https://www.itu.int/
7.4	European Telecommunications Standards Institute (ETSI)	https://www.etsi.org/
7.5	Institute of Electrical and Electronics Engineers (IEEE)	https://www.ieee.org/
7.6	International Organization for Standardization (ISO)	https://www.iso.org/
7.7	ANSI Accredited Standards Committee X9 (ASC X9)	https://x9.org/
7.8	National Security Agency (NSA)	https://www.nsa.gov/
7.9	Cloud Security Alliance (CSA)	https://cloudsecurityalliance.org/
7.10	PQCRYPTO	https://pqcrypto.eu.org/
7.11	Post-Quantum Cryptography Coalition (PQCC)	https://pqcc.org/

7.1 National Institute of Standards and Technology (NIST)

7.1.1 組織概要

米国国立標準技術研究所（National Institute of Standards and Technology: NIST）は、米国

商務省傘下の連邦機関として、計測標準、情報技術、サイバーセキュリティ分野における国家標準および技術ガイダンスを策定・公開している。暗号技術分野では、DES、AES、SHA シリーズをはじめとする基幹暗号標準を提供してきた実績を有し、その成果は米国政府のみならず国際的にも広く参照されている。NIST の暗号標準化の特徴は、アルゴリズム仕様にとどまらず、実装、運用、移行といった実務的観点を重視している点にある。標準は FIPS として制定される一方、Special Publication (SP) および Interagency Report (IR) を通じて、技術的背景や移行戦略、設計上の留意事項が体系的に補完されている。PQC についても、NIST は既存暗号資産を前提とした段階的移行問題として位置付け、従来暗号と PQC を併用するハイブリッド方式を移行期の重要な選択肢として明確に示している。

7.1.2 PQC およびハイブリッド構成に関する標準化動向

NIST における PQC 標準化は、2016 年に公開された Call for Proposals [37]を起点として開始された。量子コンピュータの将来的な実用化による現在の公開鍵暗号の破綻リスクを背景として、耐量子計算機性を有する新たな鍵共有方式と署名方式の公募と長期評価が実施されてきた。

PQC アルゴリズム選定と並行して、既存暗号から PQC への移行方法そのものが重要な検討対象とされており、NIST IR 8547 [22]では移行期における主要な選択肢が提示されている。同文書では、単一の PQC アルゴリズムへの移行だけでなく、PQC と量子コンピュータに対して脆弱な暗号を組み合わせるハイブリッドソリューションの役割とトレードオフが提示されている。

NIST IR 8547 では、ハイブリッドソリューションは「構成アルゴリズムの少なくとも一つが安全である場合に全体の安全性が維持される」方式として説明されており、将来のアルゴリズム安全性に関する不確実性に対応する手段と位置づけられている。また、既存暗号の継続利用要件がある場合の移行パスとしても利用される。一方で、実装やアーキテクチャの複雑化、運用コストの増大といったトレードオフが存在し、NIST はこれらを PQC への完全移行までの暫定的措置として位置付けている。

ハイブリッドソリューションに関連する具体的技術は以下のとおりである。

- Hybrid Key-Establishment Techniques
  - Hybrid Key-Establishment は複数の鍵共有方式を組み合わせた構成であり、本報告書では合成（composite）に相当する。
  - NIST は NIST SP 800-56C Rev.2 に記載された汎用合成鍵共有の利用を許容している。Z’ = Z || T は shared secret として扱われ、NIST SP 800-56C Rev.2 の任意の鍵導出方式を Z’に適用して鍵素材を導出することができる。
    Z：NIST SP 800-56A または NIST SP 800-56B に従って生成された shared secret
    T：その他のスキームにより生成・配布される shared secret
  - NIST は NIST SP 800-56C Rev.2 を改訂し、Z が現在および将来の NIST 鍵共有規格に基づき生成されることを許容する予定。Z を生成できる方式として、NIST SP 800-56A、NIST SP 800-56B、FIPS 203（ML KEM）、将来の PQC KEM 標準が含まれる。
  - NIST は、鍵結合器（Key Combiner）に関する追加ガイダンスを、今後公開予定の NIST SP 800-227 で提供する予定である。[9]
- Hybrid Digital Signature Techniques
  - Hybrid Digital Signature とは同一メッセージに対して 2 つ以上の署名を付与する方式 dual signatures である。本報告書の用語の定義では、合成(composite)にあたる。

[9] NIST IR 8547(2024 年)を発行後に NIST SP 800-227(2025 年)を発行している

- 検証時には、構成要素となるすべての署名が正しく検証される必要がある。
- dual signatures は、文書や電子メールなどのユーザデータ、あるいはデジタル証明書の署名に利用できる。
- NIST の既存標準およびガイドラインは､少なくとも 1 つの署名アルゴリズムが NIST 承認である限り、dual signatures の利用を許容している。

これら方針や仕様の実運用における妥当性は、NIST SP 1800-38C [38]において、TLS や X.509 等の既存プロトコルを対象とした相互運用性および性能評価として報告されており、NIST の PQC 移行戦略が理論にとどまらないことを裏付けている。

7.1.3 技術仕様におけるハイブリッド構成の詳細

本節では、NIST SP 800-227 Section 4.6「Multi-Algorithm KEMs and PQ/T Hybrids」 [4]に基づき､鍵共有におけるハイブリッド構成の設計思想および技術仕様を整理する｡Section 4.6 では、複数の鍵共有方式（Key Encapsulation Mechanisms: KEMs）を並行して用い、それぞれから得られる共有秘密を安全に合成する Multi-Algorithm KEM（Composite KEM）が定義されている。特に、耐量子計算機暗号（Post-Quantum: PQ）KEM と既存の（Traditional: T）KEM を組み合わせた構成は、PQ/T Hybrid として位置付けられている。Multi-Algorithm KEM の基本構造は、各 KEM による独立したカプセル化処理、複数の共有秘密の取得、そしてキーコンバイナによる最終共有鍵への統合から構成される。合成後の共有鍵が、少なくとも共有鍵の生成に使用された一つの KEM が安全である限り安全性を維持することを目標としている。ただし、この性質は自動的に保証されるものではなく、鍵結合器(キーコンバイナ)には NIST SP 800-56C Rev.2 で規定された承認済み鍵導出手法を用いることが要求されている。なお、Section 4.6 に示される Multi-Algorithm KEM の手法概要と構築方法に関しては、本報告書の 4.2 節に示しているため、本節では記載を割愛する。

また、Section 4.6 では Composite KEM のセキュリティ考慮事項として、複数方式を併用することによって実装やプロトコルが複雑化し、プロトコル内に追加の選択肢が生じることでダウングレード攻撃などのリスクが発生し得る点にも言及している。

NIST は、Multi-Algorithm KEM および PQ/T Hybrid を恒久的な解決策とは位置付けておらず、あくまで移行期における暫定的手段として利用し、長期的には単一の PQC 方式へ収束させることを想定している。

7.2 Internet Engineering Task Force (IETF)

IETF の各種 WG 及び IRTF の CFRG の状況は以下の通りである。本節では、RFC 若しくは WG Draft の文献を対象とする。

7.2.1 Transport Layer Security Working Group (TLS WG)

- WG 概要

  TLS WG は IETF においてインターネット上の安全な通信を実現する Transport Layer Security(TLS)の仕様策定を担うワーキンググループである。TLS 1.3(RFC 8446)を基盤に、暗号アルゴリズムの更改、拡張機能、運用上の相互運用性を継続的に整備している。耐量子計算機性への対応として、TLS 1.3 の鍵共有方式において複数の KEM を組み合わせて利用するハイブリッド KEM の設計を WG ドラフト（draft-ietf-tls-hybrid-design-16）で提示している。

- PQC およびハイブリッド構成に関する標準化動向
  draft-ietf-tls-hybrid-design-16 では、TLS 1.3 の鍵共有方式として複数の KEM を組み合わせるハイブリッド KEM を定義している。主要な仕様は次の通りである。

  - NamedGroup の設計：
    NamedGroup に、複数の KEM から構成される順序付き組を定義し、ハイブリッド構成として登録する。例えば、従来の ECDH ベースの鍵共有と耐量子計算機性を備えた KEM を用いた鍵共有を組み合わせた構成を MyECDHMyPQKEM のような 1 つの NamedGroup として扱い、これを TLS 1.3 の既存 NamedGroup と同様に、ClientHello／ServerHello で提示・選択されるネゴシエーション対象として登録する形式が示されている。
  - key_share の構造：
    key_share の KeyShareEntry.key_exchange には、選択対象となる NamedGroup に含まれる各 KEM の公開鍵／暗号文を固定長で連結して格納する。これらは TLS 1.3 の ClientHello／ServerHello における supported_groups および key_share 拡張の交換を通じてネゴシエーションされ、サーバが特定の NamedGroup を選択することで、最終的に鍵共有方式が合意される。
  - shared_secret の連結と HKDF の扱い：
    各々の KEM を用いた鍵共有から得られた shared_secret は、前段で選択された NamedGroup に対応する key_share の値からそれぞれ導出され、これら複数の shared_secret を連結したものを TLS 1.3 の既存 HKDF 鍵スケジュールに入力する。

  また、運用上の論点として、ClientHello に関する通信量の増加、KEM 固有の失敗確率に伴うハンドシェイク再試行、IANA Supported Groups へのハイブリッド組登録、TLS を UDP 化したプロトコル Datagram TLS への適用可能性が挙げられている。
  なお、ハイブリッド化の目的は、暗号移行に伴う不確実性に備える設計目標として示されている一方で、本ドラフトでは formal security proof（数学的モデルに基づく安全性証明）を提供していない。本ドラフトは構成要素となる KEM の種類を限定せず、特定方式に依存しないハイブリッド構成を想定する。

- 技術仕様におけるハイブリッド構成の詳細
  TLS 1.3 におけるハイブリッド鍵共有の具体的な手順については、4.3.1 節において図示しているため、本節では詳細説明を割愛する。

7.2.2 Limited Additional Mechanisms for PKIX and SMIME Working Group (LAMPS WG)

- WG 概要
  LAMPS WG は、IETF において X.509 証明書、PKIX、S/MIME、CMS 等に関する拡張仕様を策定するワーキンググループであり、既存 PKI 基盤との後方互換性を維持しつつ、暗号技術の拡張・進化を可能とする仕組みの標準化を担っている。PQC 移行期においては、証明書形式やアルゴリズム識別子、鍵および署名の表現方法が既存 PKI 基盤に与える影響が大きいため、LAMPS WG は TLS 等の通信プロトコル層とは異なり、証明書・署名・鍵管理といった PKI レイヤーに焦点を当てた標準化を進めている。

- PQC およびハイブリッド構成に関する標準化動向
  LAMPS WG における PQC への対応は、証明書および PKIX データ構造において PQC 単独方式とハイブリッド方式の共存を可能にすることを基本方針として進められている。中心的な取り組みとして、PQC と従来暗号を組み合わせた複合方式（Composite）の標準化が挙げられ、署名および鍵カプセレーションの双方について Internet-Draft が策定されている。
  合成署名については、「Composite ML-DSA for use in X.509 Public Key Infrastructure」

（draft-ietf-lamps-pq-composite-sigs-14） [12]において、ML-DSA（FIPS 204）と既存署名アルゴリズム（RSA、ECDSA、EdDSA 等）を組み合わせた Composite 署名方式が定義されている。これは単一の AlgorithmIdentifier として扱える設計を採用しており、既存の PKI 実装や証明書検証ロジックを大きく変更することなく、PQ/T ハイブリッド署名を導入可能とする点に特徴がある。

鍵共有に関しても、「Composite ML-KEM for use in X.509 Public Key Infrastructure」（draft-ietf-lamps-pq-composite-kem-12） [11]において、ML-KEM（FIPS 203）と RSA-OAEP や ECDH を組み合わせた Composite KEM が定義されている。これらの仕様は RFC 9794 で整理された PQ/T ハイブリッドの設計思想を PKIX 環境に適用するものであり、移行期におけるリスク低減を主目的としている。

さらに、「A Mechanism for X.509 Certificate Discovery」（draft-ietf-lamps-certdiscovery-02） [18]により、複数証明書を関連付けて発見可能とする仕組みが提案されており、PQC 証明書と従来証明書の併用や段階的切り替えを支援する基盤技術として位置付けられる。

なお、具体的な仕様に関しては、本報告書の 4.2 節に示しているため、本章では記載を割愛する。

7.2.3 Messaging Layer Security Working Group（MLS WG）

- WG 概要

  MLS WG は、IETF においてエンドツーエンド暗号化（E2EE）を前提としたセキュアなグループメッセージング基盤の標準化を担う作業部会である。主成果物である RFC 9420（Messaging Layer Security: MLS）は、大規模かつ動的な参加者集合を想定した効率的な鍵更新および前方秘匿性を実現するプロトコルとして位置付けられている。MLS WG では、実運用を意識した拡張性や長期安全性の確保が重視されており、PQC を含む次世代暗号技術の段階的導入についても継続的に検討が進められている。

- PQC およびハイブリッド構成に関する標準化動向

  MLS WG における PQC 対応の検討は、量子計算機の進展を見据えた長期機密性確保への対応を背景として進められている。RFC 9420 自体は従来暗号を前提としているが、その拡張として、PQC アルゴリズムを直接適用する方式や、既存の暗号方式と PQC を組み合わせるハイブリッド方式が議論されている。特に、計算量や通信量の増大という PQC 固有の課題に対し、効率性と安全性のバランスを取る実装手法が重要視されている。参考文献である draft-ietf-mls-combiner-02 [15]は、従来 MLS セッションと PQC MLS セッションを組み合わせる償却型ハイブリッド手法を提案しており、MLS WG における PQC 適用方針を具体化する中核的文書と位置付けられる。

- 技術仕様におけるハイブリッド構成の詳細

  draft-ietf-mls-combiner-02 [15]では、Amortized Post-Quantum MLS（APQ-MLS）Combiner と呼ばれる方式が定義されている。draft-ietf-mls-combiner-02 で定義される仕様は以下のとおりである。また、図 7-1 は本仕様の構造を視覚的に示したものであり、APQ-MLS Combiner の実装および性能評価を提示した学術論文 [39]の図を引用している。

  - 本方式では、1 つの PQ MLS セッションと 1 つの従来 MLS セッションを並行して運用し、PQ セッションで生成される exporter secret を従来セッションに取り込むことで、従来セッションに PQ 保証を付与する仕組みである。
  - 図中で示される Partial Update は、従来セッションのみで行われる通常の鍵更新であり、PQ 演算を伴わない。一方、Full Update は PQ セッションの exporter secret を従来セッションに注入して鍵スケジュールを更新する処理であり、従来セッションが PQ 安全性を獲得する更新である。
  - APQ-MLS では、Partial Update と Full Update を柔軟に組み合わせることで、PQ 演算の

計算量やメッセージサイズの増大を抑えつつ、MLS に求められる頻繁な鍵更新要求に対応できるよう設計されている。

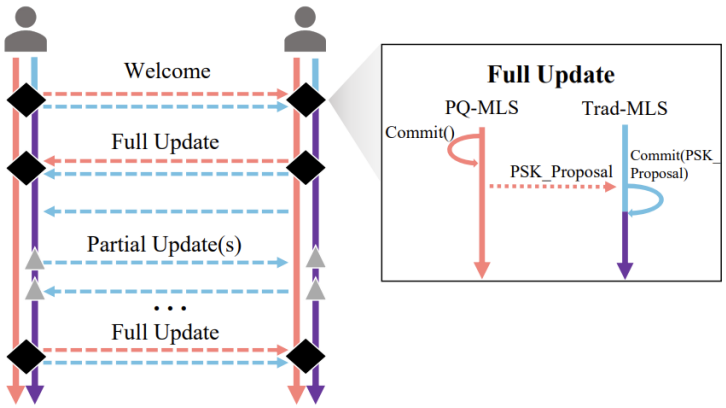


図 7-1 APQ combiner の概要 [39]。PQ MLS セッションの exporter secret を用いて、従来 MLS セッションに PQ 保証を注入する構造を示す。Partial Update は従来セッションのみで鍵更新を行い、Full Update は PQ セッションの秘密値を取り込み PQ 保証を付与する。

7.2.4 Post-Quantum Use in Protocols Working Group (PQUIP WG)

- WG 概要

  PQUIP WG は、IETF における PQC 移行期のプロトコル設計・運用指針を横断的に整理するために設置されたワーキンググループである。TLS、PKIX／LAMPS、IPsecME など個別 WG の仕様検討を補助する立場から、用語の標準化、ハイブリッド（PQ/T）方式に関する設計目標、セキュリティ特性、証明書モデルの整理などを担う。特に署名に関しては「ハイブリッド署名のスペクトラム」を提示し、非分離性（WNS／SNS）や同時検証（SV）などの概念を共有化することで、プロトコルや PKI における一貫した導入・検証を支援する。

- PQC およびハイブリッド構成に関する標準化動向

  PQUIP WG の中心成果として、PQ/T ハイブリッド方式の共通用語を定義する RFC 9794 [23]が 2025 年 6 月に発行された。同 RFC は、PQ/T ハイブリッド方式に関する用語体系を以下の 4 区分に基づき体系化している。各区分における具体的な定義内容は表 7-2 から表 7-5 に示すとおりである。

  - Primitives（暗号要素）：暗号方式の分類およびマルチアルゴリズム構成の基本概念を定義する。
  - Protocols（プロトコル）：ハイブリッド方式を取り扱うための鍵共有モデルやプロトコル構造を定義する。
  - Properties（性質）：PQ/T モードにおける秘匿性・認証性・相互運用性・後方／前方互換性などの成立条件を定義する。
  - Certificates（証明書モデル）：移行期の証明書構造や PQ/T 共存のための証明書モデルを定義する。

  併せて、インフォメーション目的の Internet-Draft「Hybrid signature spectrums」(draft-ietf-pquip-hybrid-signature-spectrums-07) [24]では、RFC 9794 で整理された PQ/T ハイブリッド方式の分類を署名分野に特化して拡張している。本ドラフトは以下の 6 区分から構成され、ハイブリッド署名の設計・検証に必要となる概念を体系的に示している。各区分における具体的な定義内容は表 7-6 から表 7-11 に示すとおりである。

- Terminology（基本概念）：ハイブリッド署名方式、構成要素、攻撃モデル、アーティファクトなどの基礎語彙を整理する。
- Goals（設計目標）：ハイブリッド署名が満たすべき安全性・互換性・非分離性等の指標を示す。
- Non-Separability Spectrum（非分離性スペクトラム）：署名の分離困難性を段階的に分類し、強度の違いを示す。
- Artifacts(痕跡の種別と配置):message･certificate･signature･protocol／policy のどこにハイブリッド方式であることの痕跡を配置するかを分類する。
- Need for Approval Spectrum（承認要件の分類）：FIPS 140 を含む既存承認モジュールとの関係性を踏まえ、ハイブリッド方式が必要とする承認レベルを分類する。
- EUF-CMA Challenges（偽造困難性に関する課題）：コンポーネント偽造、鍵再利用、分離困難性との関係など、安全性分析における注意点を整理するハイブリッド署名を構成する要素と攻撃モデルを整理する。

- RFC 9794 における用語の定義

表 7-2 Primitives(暗号要素) [23]

用語	定義
Traditional asymmetric cryptographic algorithm	整数因数分解･離散対数問題等に基づく非耐量子計算機公開鍵暗号
Post-quantum asymmetric cryptographic algorithm	耐量子計算機性を持つ公開鍵暗号
Component asymmetric algorithm	マルチアルゴリズム方式を構成する個々のアルゴリズム
Single-algorithm scheme	1 種類の非対称暗号アルゴリズムのみから構成される方式。
Multi-algorithm scheme	同一目的の処理（署名や鍵共有など）を複数の公開鍵暗号アルゴリズムを併用して構成する方式。
PQ/T hybrid scheme	PQC と既存の暗号アルゴリズムを組み合わせて構成されるマルチアルゴリズム方式

表 7-3 Protocols(プロトコル) [23]

用語	定義
PQ/T hybrid protocol	PQC と既存の暗号アルゴリズムを同時に利用するプロトコル
Composite key establishment	暗号要素のみ変更しプロトコル構造を維持する鍵生成
Non-composite key establishment	複数鍵共有をプロトコルレベルで併用する方式

表 7-4 Properties(性質) [23]

用語	定義
PQ/T hybrid confidentiality	PQ/T ハイブリッド構成に含まれる複数アルゴリズムの

	うち、少なくとも 1 つが機密性を保持している限り、全体として機密性が維持される性質
PQ/T hybrid authentication	PQ/T ハイブリッド署名に含まれる複数の署名方式のうち､少なくとも 1 つが署名の安全性を保持している限り、全体として認証が成立する性質
PQ/T hybrid interoperability	通信当事者がサポートするアルゴリズム集合に、少なくとも 1 つの共通構成要素が存在する場合に、プロトコルとして相互運用を維持できる性質
Backwards compatibility	従来方式しか扱えない検証者であっても、ハイブリッド署名に含まれる従来方式の署名部分のみを検証して受理できる互換性の性質
Forwards compatibility	将来の PQC 完全対応環境において、ハイブリッド署名の PQC 署名部分のみ、あるいは両方を選択的に利用できる柔軟性を指す性質

表 7-5 Certificates(証明書モデル) [23]

用語	定義
PQ/T hybrid certificate	PQC 用の公開鍵と既存の暗号方式用の公開鍵を含む単一証明書
Post-quantum certificate	PQC 署名アルゴリズムのみを含む証明書
Traditional certificate	既存の署名アルゴリズムのみを含む証明書
PQ/T hybrid certificate chain	各証明書が PQ/T 署名を用いる証明書チェーン(エンドエンティティ証明書からルート認証局証明書までの階層構造)
PQ/T parallel PKI	PQC 証明書チェーン(エンドエンティティ証明書からルート認証局証明書までの階層構造)と既存の証明書チェーン(エンドエンティティ証明書からルート認証局証明書までの階層構造)を並列利用

- Hybrid signature spectrums におけるハイブリッド署名の体系

表 7-6 Terminology(基本概念) [24]

項目	定義・内容
Hybrid signature scheme	2 つ以上の署名アルゴリズムから構成されるマルチアルゴリズム署名方式
Hybrid signature / Dual signature	ハイブリッド署名方式により生成された署名
Component signature scheme	ハイブリッド署名を構成する個々の署名アルゴリズム
Artifact	ハイブリッド署名であることを示す痕跡(署名を分離しても残り、ハイブリッド利用の意図や証拠

	となる情報)
Stripping attack	ハイブリッド署名から一部署名を除去し、単独署名として悪用する攻撃
Component message forgery attacks	ハイブリッド署名の構成要素となる署名だけを単独で偽造する攻撃

表 7-7 Goals(設計目標) [24]

項目	定義・内容
Hybrid authentication	いずれか 1 つの署名方式が安全であれば認証が成立する性質
Hybrid unforgeability	EUF-CMA などの署名安全性が、構成要素の 1 つでも保持されていれば維持される性質
Proof composability	ハイブリッド署名の安全性が構成署名方式の安全性に基づいて証明できる性質
Weak non-separability (WNS)	署名を分離してもハイブリッドの痕跡 (artifact) は残るが、残った署名が検証に成功する場合がある性質
Strong non-separability (SNS)	署名を分離した場合、残った署名は必ず検証に失敗する性質
Simultaneous verification	全ての署名構成要素が同時に検証されなければ成功しない性質
Backwards compatibility	従来方式しか扱えない検証者でも、従来部分のみ検証して受理できる性質
Hybrid generality	複数カテゴリの署名構造に適用可能な汎用性を持つ性質

表 7-8 Non-Separability Spectrum(非分離性スペクトラム) [24]

項目	定義・内容
No non-separability	署名を分離しても痕跡が残らず、ハイブリッドであることを検知できない性質
Weak non-separability	痕跡は残るものの、残存署名が単独署名として検証成功してしまう場合がある性質
Strong non-separability	痕跡が署名内部に存在し、分離すると必ず検証が失敗する性質
SNS + Simultaneous verification	署名が完全不可分で、全構成要素を同時に検証しない限り成功しない最強の非分離性

表 7-9 Artifacts(痕跡の種別と配置) [24]

項目	定義・内容
Artifact location in message	メッセージ内にハイブリッド署名であることを示

	す情報（ラベル等）を配置する方式
Artifact location in certificate	証明書内のフィールドに、ハイブリッド署名利用を示すメタ情報を埋め込む方式
Artifact location in signature	複数の署名要素が結合され、単独署名として切り離せない構造を署名内部に持たせる方式
Protocol / policy artifacts	プロトコル仕様やポリシー上で「ハイブリッド署名を要求する」という設定・規定を痕跡として扱う方式

表 7-10 Need for Approval Spectrum(承認要件の分類) [24]

項目	定義・内容
New algorithm	新しい署名アルゴリズムとして扱われ、個別承認（例：FIPS）が必要となる
No approved module	既存承認済みモジュールに依拠するが、実装変更が必要で追加承認の要否が不明確な状態
1-out-of-n approved	構成要素のうち少なくとも 1 つを承認済みモジュールとしてブラックボックス利用できる状態
All approved	全ての構成署名方式が承認済みモジュールとしてブラックボックス利用可能で、内部動作を変更せずに組み合わせられる状態

表 7-11 EUF-CMA Challenges(偽造困難性に関する課題) [24]

項目	定義・内容
Component forgery risk	ハイブリッド署名の構成要素署名が単独署名として偽造されるリスク
Key reuse restriction	鍵を使い回すことにより生じる偽造リスクを防ぐための鍵利用制限
SNS-based mitigation	Strong Non-Separability（SNS）により分離攻撃を不可能にし、コンポーネント偽造を根本的に防止する対策

7.2.5 Crypto Forum Research Group (CFRG)

- RG 概要

  CFRG は、IRTF（Internet Research Task Force）配下に設置された暗号技術分野の研究グループであり、IETF におけるインターネットプロトコル標準化を暗号技術の観点から技術的に支援する役割を担っている。CFRG は自ら標準仕様（Standards Track RFC）を策定する主体ではなく、暗号アルゴリズム、鍵生成方式、電子署名方式等に関する研究成果や設計上の検討結果を、Informational または Experimental RFC として公開する点に特徴がある。これらの成果は、TLS、COSE、JOSE 等の IETF ワーキンググループにおける仕様検討や、NIST や ETSI による PQC 移行方針の技術的検討において、重要な参照情報として位置付けられている。

- PQC およびハイブリッド構成に関する標準化動向
  PQC への移行期において、既存の公開鍵暗号と新たな PQC アルゴリズムを併用するハイブリッド方式は、安全性と実運用性の両立を図るための重要なアプローチとして位置付けられている。CFRG では、量子コンピュータによる攻撃および量子コンピュータではない従来型のコンピュータによる攻撃の双方に対する耐性を同時に確保する、AND セキュリティモデルを前提としたハイブリッド鍵共有方式（ハイブリッド KEM）について、設計原則や安全性要件の整理が行われており、その代表的な成果として draft-irtf-cfrg-hybrid-kems-07（Hybrid Key Encapsulation Mechanisms（KEMs））［10]が公開されている。これらの検討では、アルゴリズム単体の安全性のみならず、コンバイナ（combiner）の構成、ダウングレード攻撃への耐性、ならびに実装および相互運用性への影響といった実運用上の論点が重視されている。

7.3 International Telecommunications Union (ITU)

7.3.1 組織概要

国際電気通信連合（International Telecommunication Union: ITU）は、電気通信および情報通信技術（ICT）分野に関する国連専門機関であり、標準化部門である ITU-T が電気通信の技術・運用・料金に関する勧告（Recommendation）を策定している。ITU-T では、ディレクトリサービスや PKI の枠組み（X.500 系列）やサイバーセキュリティ、量子通信（X.1700 系列）など広範な勧告群を所掌し、国際的な相互運用性の確保に資する技術的枠組みを提供してきた。とりわけ X.509 勧告は、公開鍵基盤（PKI）および権限管理基盤（PMI）の枠組みを定義し、証明書、失効リスト、拡張、検証手順、ディレクトリスキーマに至るまでを包括的に規定する、ITU-T と ISO/IEC による共同規格（ISO/IEC 9594-8）として位置づけられている［19]。

7.3.2 PQC およびハイブリッド構成に関する標準化動向

ITU-T における量子安全（Quantum-safe）関連の標準化は、PKI／PMI を担う X.500 系列（X.509）に加え、量子鍵配送ネットワーク（QKDN）や量子安全通信を対象とする X.1700 系列を中心に展開されている。PQC 移行期におけるハイブリッドの取り扱いについて、ITU-T は暗号プロトコルそのものを定義するのではなく、勧告における拡張や技術報告を通じて、複数暗号アルゴリズムの併用を前提とした運用上の考慮事項を整理している。例えば X.509 第 9.8 節では、代替暗号アルゴリズムおよび代替デジタル署名に関する拡張が規定されており、第 6.2.3 節では暗号アルゴリズム移行に伴う一般的な考え方が示されている。

一方、通信プロトコルレベルにおけるハイブリッド鍵共有や QKD 併用の論点については、ITU-T 技術報告「Overview of hybrid approaches for key exchange with quantum key distribution（XSTR-HYB-QKD, 2022-05）」［40]において整理されている。同報告は、ETSI TS 103 744、NIST SP 800-56C Rev.2、IETF RFC 8784 など既存の仕様・ガイダンスを俯瞰し、QKD によって生成された鍵を既存プロトコルに統合する際の概念的整理や課題を示している。これらの検討は、ITU-T 自身が詳細仕様を定めるというよりも、関連標準間の整合的な理解を促す位置づけにある。

7.3.3 技術仕様におけるハイブリッド構成の詳細

ITU-T 勧告において、暗号プロトコルとしてのハイブリッド鍵共有を直接定義した仕様は存在しない。一方で、X.509 勧告は移行期における複数アルゴリズム併用を想定した拡張を備えており、PKI／PMI の運用面から複数アルゴリズム併存を可能にする拡張を提供している。以下では、X.509

(2019 版／ISO/IEC 9594-8:2020) [19]を中心に、その位置づけを整理する。

X.509 第 9.8 節では、一つのアルゴリズムの証明書や失効リストにおいて代替の公開鍵アルゴリズムや代替のデジタル署名を拡張として付与できる枠組みが定義されている。これにより、既存の暗号方式のみを理解する環境と新方式(PQC 署名等)を処理できる環境が併存する環境においても、同一オブジェクトに複数アルゴリズムに関する情報を保持する運用が可能となる。具体的な構造に関しては、本報告書の 4.2 節に示しているため、本節では記載を割愛する。

7.4 European Telecommunications Standards Institute (ETSI)

7.4.1 組織概要

European Telecommunications Standards Institute (ETSI) は、欧州を拠点とする国際標準化機関であり、情報通信技術 (ICT) 分野における国際的な技術仕様およびガイドラインの策定を担っている。通信事業者、装置ベンダ、研究機関、行政機関等から構成され、3GPP 等の標準化活動を通じて通信基盤のセキュリティ標準化に重要な役割を果たしてきた。

ETSI における PQC への取り組みは、暗号アルゴリズムの選定そのものよりも、既存プロトコルやシステムにおける移行設計・運用設計に焦点を当てている点に特徴がある。特に、Technical Committee CYBER (TC CYBER) において、Quantum-Safe Cryptography (QSC) を主題とした文書群が体系的に整備されている。

7.4.2 PQC およびハイブリッド構成に関する標準化動向

ETSI における PQC の標準化動向は、暗号アルゴリズム単体の規定ではなく、PQC を既存通信システム・暗号基盤へどのように段階的に導入するかという移行設計に主眼が置かれている。量子計算機の実用化時期や PQC アルゴリズムの成熟度に不確実性が残る中で、ETSI は移行期における安全性・相互運用性・運用継続性の確保を標準化の中心課題として位置付けている。

そして、PQC を単独で導入する「完全移行モデル」だけでなく、既存の暗号方式と PQC を併用するハイブリッド方式を重要な選択肢として明確に位置付けている。ETSI TR 103 966 V1.1.1 (2024-10) [41]は、PQC 移行におけるハイブリッド方式の役割を概念的に整理した技術報告書であり、PQC アルゴリズムの成熟度への備え、後方互換性の確保、既存プロトコル制約への対応といった観点から、ハイブリッド方式が検討される理由を体系的に示している。

同 TR では、PQC そのものに関しても、既存の暗号方式と比較した公開鍵や暗号文などの鍵材料のサイズや計算コストの増大、プロトコルメッセージ拡張、実装の複雑化といった技術的特性が整理されており、これらが移行時の障壁となり得る点が明示されている。ETSI は、こうした PQC 特有の課題を踏まえた上で、移行初期段階においてはハイブリッド方式が現実的なリスク低減策として機能することを示唆している。

実際の技術仕様としては、ETSI TS 103 744 V1.2.1 (2025-03) [16]が、耐量子計算機性を備えたハイブリッド鍵共有方式を規定している。本仕様では、従来型の ECDH による鍵共有と、FIPS 203 で標準化された ML-KEM による鍵共有を独立に実行し、それぞれから得られる複数の共有秘密を KDF により安全に結合するハイブリッド構成が定義されている。これは、PQC 単独方式へ完全移行する前段階として、既存の PKI との互換性を維持しながら耐量子計算機性を付加する設計方針を反映したものである。

さらに、ETSI TS 104 015 V1.1.1 (2025-02) [42]は、耐量子計算機性を備えたハイブリッド KEM およびアクセス制御を組み合わせた応用的仕様である。本仕様では、Computational Diffie-Hellman (CDH) に基づく従来型 KEM と Learning With Errors(LWE) に基づく PQ KEM(FIPS 203 で標

準化された ML-KEM を含む)を組み合わせることで、少なくとも一方の暗号方式が安全であれば秘匿性が維持されるハイブリッド構造を前提としつつ、PQC への移行期における実運用要件（匿名性、属性ベース制御、トレーサビリティ）を暗号仕様レベルで実現している。

以上より ETSI の PQC 標準化動向は、PQC アルゴリズム単体の安全性評価にとどまらず、ハイブリッド方式を含む複数の移行シナリオを想定した実用指向の技術体系として整理されている点に特徴がある。これは、NIST や IETF で策定されるアルゴリズム・プロトコル標準を補完し、実システムへの PQC 導入を段階的に進めるための指針を提供する役割を ETSI が担っていることを示している。

7.4.3 技術仕様におけるハイブリッド構成の詳細

ETSI TS 103 744 V1.2.1 [16]は、耐量子計算機性ハイブリッド鍵共有に関する代表的な技術仕様であり、複数の鍵共有方式から得られる共有秘密を安全に合成するための構造を体系的に定義している。本仕様では、現在の暗号方式として ECDH を、PQC 方式として ML-KEM を用いる構成を想定し、両者を並列に実行することで複数の共有秘密を生成する点が特徴である。なお、図 7-2 から図 7-6 は、これらのハイブリッド鍵共有方式（連結／カスケード、および各 keying variant）の構造を補足的に示す概念図であり、構造を視覚的に補足するために掲載している。

鍵共有メカニズムは 3 つの機能で構成される：

- KeyGen( )：秘密鍵 sk と公開鍵 P を生成する鍵生成関数。
- ResponseFunc(P)：共有秘密 k と応答値 R を生成する応答関数。ただし、処理に失敗した場合にはエラー指標⊥を返す。
- ReceiveFunc(sk, R)：秘密鍵 sk と応答値 R を受け取り、共有秘密 k を計算する関数。ただし、処理に失敗した場合にはエラー指標⊥を返す。

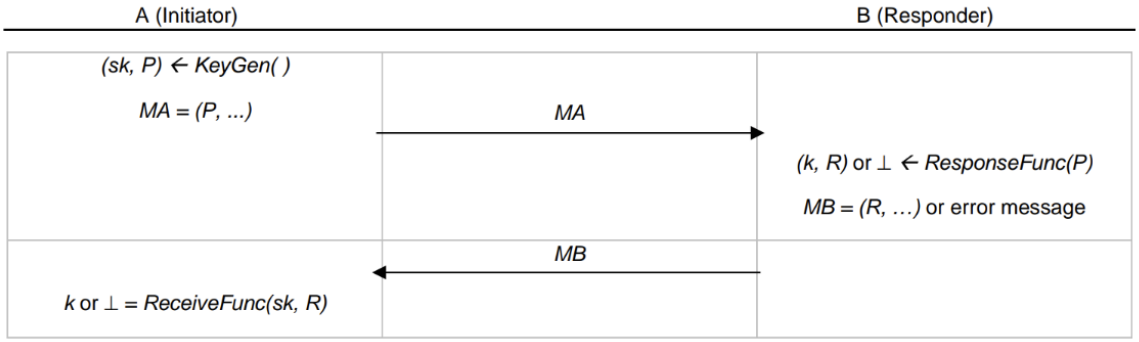


Figure 3: Key establishment abstraction

ResponseFunc がエラー指標を返した場合、レスポンダー(B)はエラーメッセージを応答し、プロセスを終了しなければならない。

イニシエータ(A)が B からエラーメッセージを受信した場合、または ReceiveFunc がエラー指標を返した場合、A はプロセスを終了しなければならない。

MA は、A から B へ送信されるオクテット列であり、1 つ以上の公開鍵の符号化を含む。必要に応じて、セッションネゴシエーション情報を含めることができる。

MB は、1 つ以上の応答値の符号化を含むオクテット文字列であり、同時にセッションネゴシエーション情報を含めることができる。

メッセージは不正な改変から保護されなければならない。メッセージは、送信者の身元に関連付けられた署名鍵によるデジタル署名を用いて保護され得る。この署名鍵は、信頼された第三者認証機関によって署名されたものである。

図 7-2 鍵共有方式の概要［16］

これらの共有秘密は単純に連結されるのではなく、CatKDF（Concatenate-based Key Derivation Function）およびCasKDF（Cascade-based Key Derivation Function）と呼ばれる二種類の鍵導出関数を用いて統合される。CatKDFは複数の共有秘密を入力としてKDFに与える方式であり、比較的実装が容易である一方、CasKDFは段階的に鍵導出を行うことで、より厳密なセキュリティ性質の保持を意図した方式である。本仕様では、これら二つの鍵導出方式について、ephemeral keying variant（セッション毎に新規生成する鍵を前提とする構成）とstatic keying variant（長期間保持される鍵を前提とする構成）の双方に対応するよう仕様化されており、利用環境やプロトコル要件に応じて適切な組合せを選択できるよう設計されている。

本条項は、ephemeral keying variantを用いた連結ハイブリッド鍵共有方式を規定する。Figure 3の鍵共有方式の記述を拡張し、単一メッセージ内で公開鍵ペアと複数の応答値を交換する。ephemeral keying variantを用いた連結ハイブリッド鍵共有方式はFigure 4に示す通り構築される。

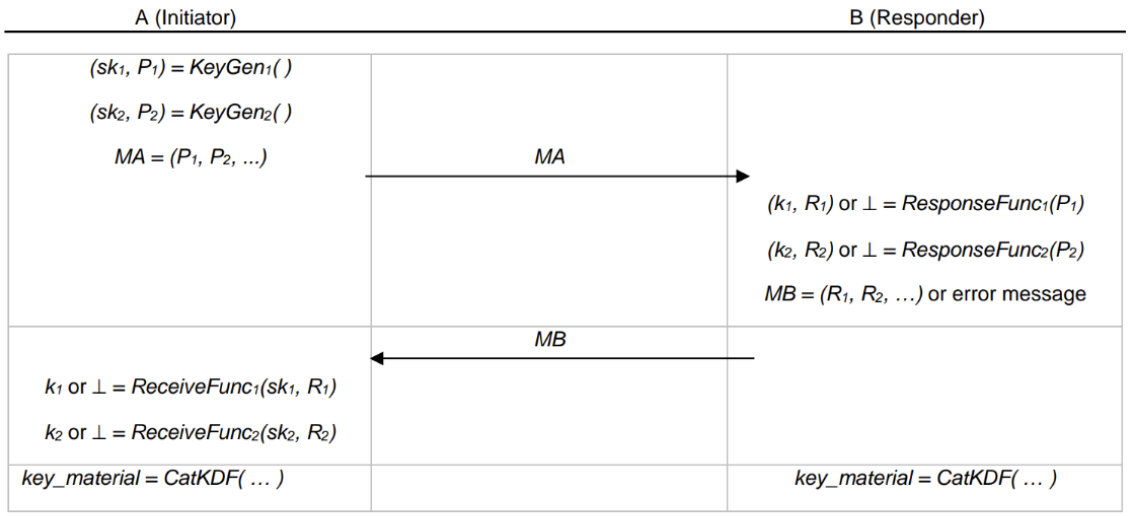


Figure 4: Concatenate hybrid key establishment - ephemeral

いずれかのResponseFunc$_i$(i={1,2})がエラー指標を返した場合、Bはエラーメッセージで応答し、プロセスを終了する。

Aがエラーメッセージを受信した場合、Aはプロセスを終了しなければならない。いずれかのReceiveFunc$_i$(i={1,2})がエラー指標を返した場合、Aはプロセスを終了しなければならない。

MAは、AからBへ送付された公開鍵Piの符号化を含むオクテット文字列とする。MAにはラベル用付加データなどのセッションネゴシエーション情報を含めることができる。

MBは、MBがエラーメッセージでない場合、応答値Riの符号化を含むオクテット文字列でなければならない。MBにはラベル用付加データなどのセッションネゴシエーション情報を含めることができる。

メッセージは不正な改変から保護されなければならない。メッセージは、送信者の身元に関連付けられた署名鍵を用いて、信頼できる第三者認証機関によって署名されたデジタル署名により保護することができる。

図 7-3 連結ハイブリッド鍵共有方式 - ephemeral keying variant [16]

本条項は、static keying variant を用いた連結ハイブリッド鍵共有方式を規定する。Figure 3 の鍵共有方式の記述を拡張し、単一メッセージにおいて、送信者(B)が受信者(A)が保持する静的な公開鍵(P_1, P_2)を信頼できる方法で取得するとともに、単一メッセージにおいて、複数の応答値（R_1, R_2）を取得する。static keying variant を用いた連結ハイブリッド鍵共有方式は Figure 5 に示すように構築される。

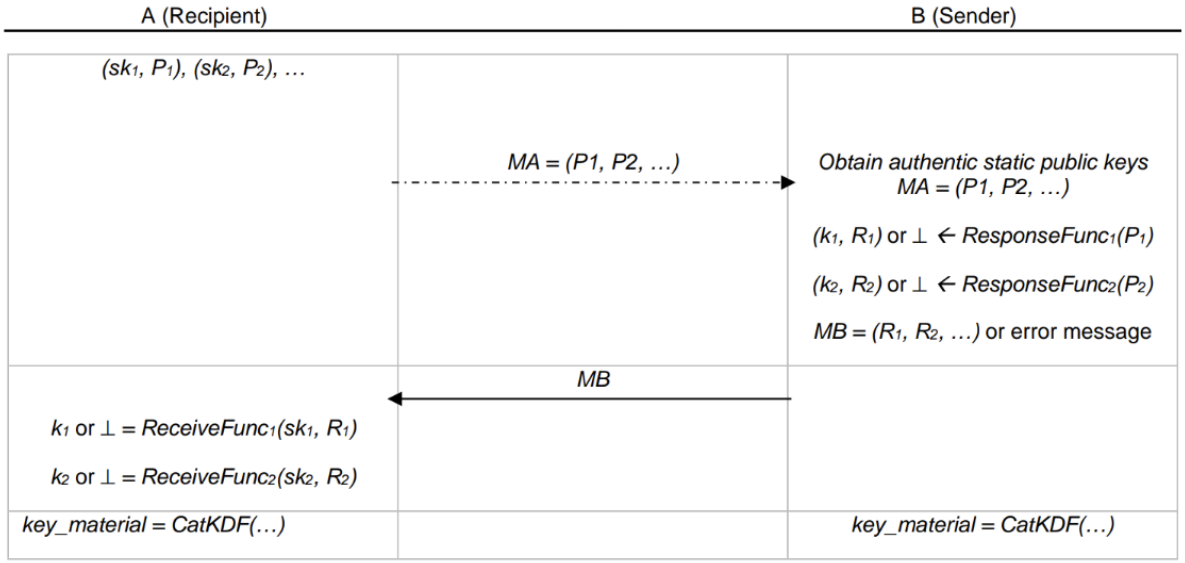


Figure 5: Concatenate hybrid key establishment - static

いずれかの $\mathrm{ResponseFunc}_i$(i={1,2})がエラー指標を返した場合、B はプロセスを終了する。いずれかの $\mathrm{ReceiveFunc}_i$(i={1,2})がエラー指標を返した場合、A はプロセスを終了する。

MA は、鍵共有の前または最中で、B が A の静的な公開鍵およびラベル用付加データ(label contribution values) などの追加値を信頼できる方法で取得していることを前提とする。

MB は、MB がエラーメッセージでない場合、応答値 R_i の符号化を含むオクテット文字列でなければならない。MB にはセッションネゴシエーション情報を含めることができる。2 つ以上の鍵共有方式が使用されている場合、MB には対応するすべての公開鍵と暗号文を含めなければならない。

メッセージは不正な改変から保護されなければならない。メッセージは、送信者の身元に関連付けられた署名鍵によるデジタル署名を用いて保護され得る。この署名鍵は信頼できる第三者認証機関によって署名される。

図 7-4 連結ハイブリッド鍵共有方式 - static keying variant [16]

本条項では、ephemeral keying variant を用いたカスケードハイブリッド鍵共有方式を規定する。Figure 3 の鍵共有方式の記述を拡張し、異なるメッセージ間で複数の公開鍵(MA_1, MA_2)を交換し、異なるメッセージ間で複数の応答値(MB_1, MB_2)を交換する。カスケードハイブリッド鍵共有方式は Figure 6 に示すように構築される。

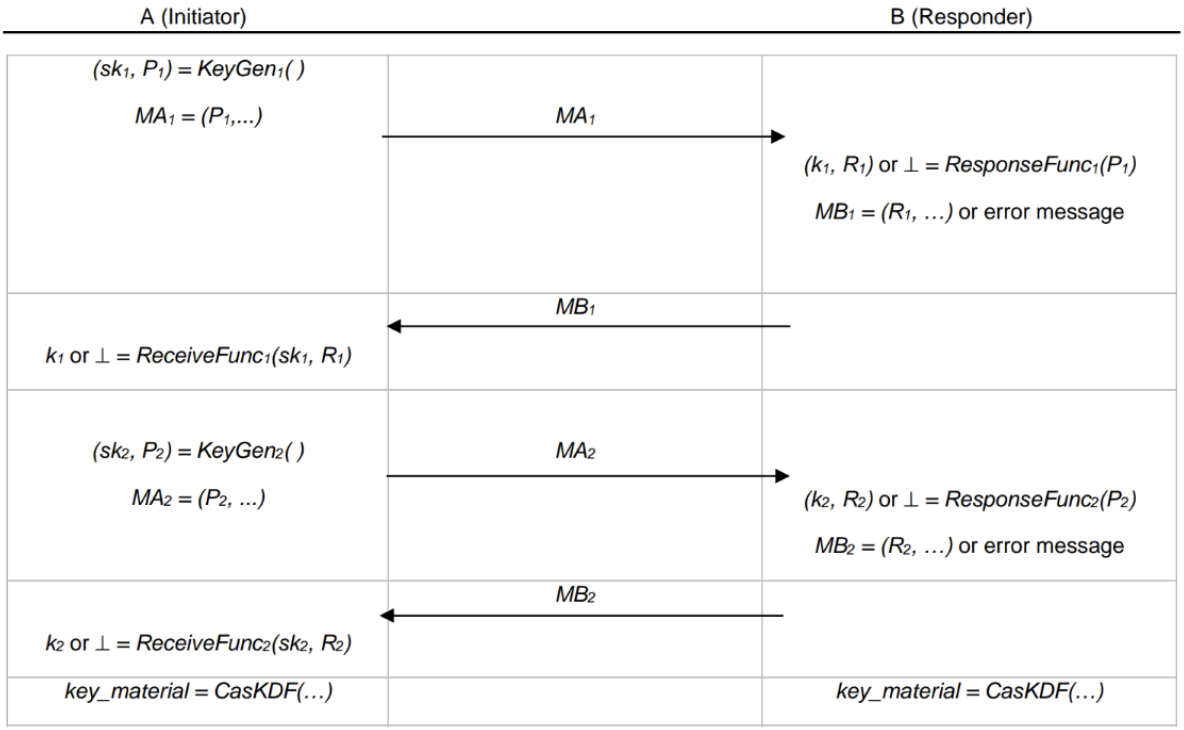


Figure 6: Cascade hybrid key establishment - ephemeral

いずれかの $ResponseFunc_i$(i={1,2})がエラーインジケータを返した場合、B はエラーメッセージを返し、プロセスを終了する。

A が B からエラーメッセージを受信した場合、A はプロセスを終了させるものとする。いずれかの $ReceiveFunc_i$(i={1,2})がエラーインジケータを返した場合、A はプロセスを終了する。

MA_i (i={1,2})は、A から B へ送付された公開鍵 P_i の符号化を含むオクテット文字列とする。MA_i (i={1,2})にはラベル用付加データなどのセッションネゴシエーション情報を含めることができる。

MB_i(i={1,2})は、応答値 R_i(i={1,2})の符号化を含むオクテット文字列とする。MB_i(i={1,2})にはラベル用付加データなどのセッションネゴシエーション情報を含めることができる。

メッセージは不正な改変から保護されなければならない。メッセージは、送信者の身元に関連付けられた署名鍵によるデジタル署名を用いて保護することができ、この署名鍵は信頼できる第三者認証機関によって署名される。

図 7-5 カスケードハイブリッド鍵共有方式- ephemeral keying variant [16]

本条項は、static keying variant を用いたカスケードハイブリッド鍵共有方式を規定する。Figure 7 に示すように構築される。

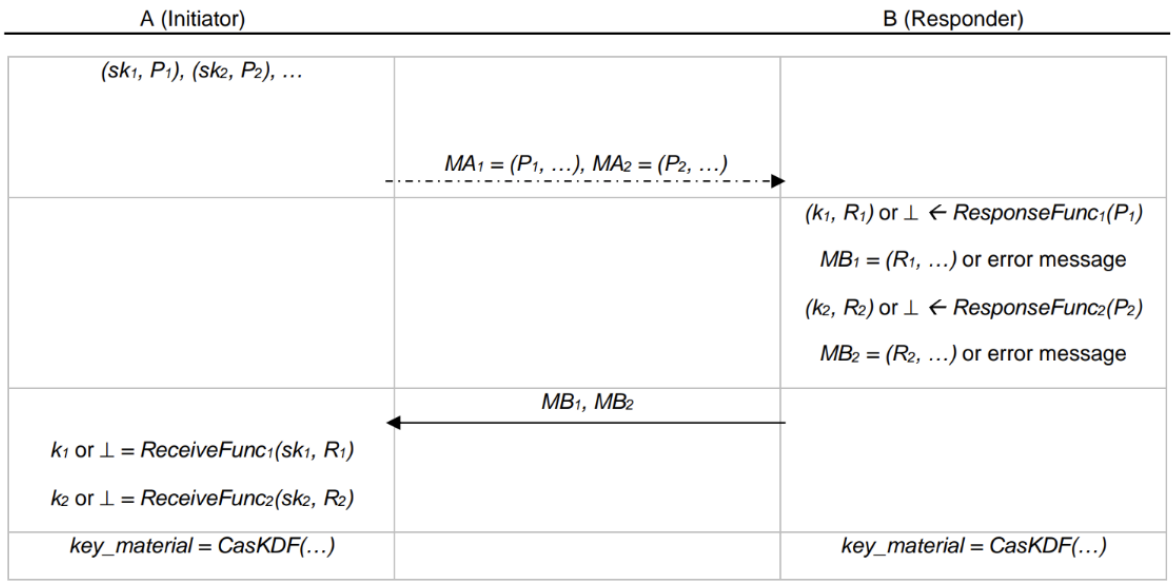


Figure 7: Cascade hybrid key establishment - static

いずれかの $ResponseFunc_i$ がエラー指標を返した場合、B はプロセスを終了する。いずれかの $ReceiveFunc_i$ がエラー指標を返した場合、A はプロセスを終了する。

MA_i(i={1,2})は、鍵確立前または鍵確立中で、B が A の静的な公開鍵およびラベル用付加データ(label contribution values) などの追加値を信頼できる方法で取得していることを前提とする。

MB_i(i={1,2})は、MB_i がエラーメッセージでない場合、応答値 Ri の符号化を含むオクテット文字列でなければならない。MB_i にはセッションネゴシエーション情報が含まれる場合がある。

メッセージは不正な改変から保護されなければならない。メッセージは、送信者の身元に関連付けられた署名鍵によるデジタル署名を用いて保護することができ、この署名鍵は信頼できる第三者認証機関によって署名される。

図 7-6 カスケードハイブリッド鍵共有方式- static keying variant [16]

重要な点として、ETSI TS 103 744 [16]は、ハイブリッド鍵共有方式の安全性が自動的に保証されるわけではないことを明示している。特定の構成では能動的攻撃に対して脆弱性が生じ得るため、鍵導出関数の選定、入力パラメータの整合性、プロトコルメッセージの完全性保護が不可欠であることが、セキュリティ考察として整理されている。これらは、PQC 単独方式に移行するまでの暫定的手法であるという位置付けとも整合している。

一方、ETSI TS 104 015 V1.1.1 [42]は、ハイブリッド方式を単なる通信路の鍵共有方式に留めず、より高度なデータ保護モデルへ拡張した技術仕様である。本仕様で定義される Hybrid Traceable KEM with Access Control (HTKEMAC) は、従来型 Non-Interactive Key Exchange(NIKE) と PQC KEM を組み合わせ、属性ベースのアクセス制御とトレーサビリティを同時に実現する構造を持つ。これにより、「誰が復号可能か」を暗号レベルで制御すると同時に、不正利用時には利用者を追跡可能とする設計が可能となる。

HTKEMAC では、暗号方式の選択をブラックボックス的に扱う構造が採用されており、将来的に PQC アルゴリズムが更新された場合でも、全体構造を維持したまま差し替え可能である点が意図されている。この点は、ETSI が PQC およびハイブリッド方式を恒久的解としてではなく、技術進展に応じて更新される移行期技術として捉えていることを反映している。

以上のように、ETSI の技術仕様におけるハイブリッド方式は、単なる防御的多重化ではなく、PQC

移行期における安全性・運用性・将来拡張性を同時に満たす設計指針として体系化されている。この点が、アルゴリズム中心の PQC 標準とは異なる ETSI 仕様の特徴である。

7.5 Institute of Electrical and Electronics Engineers (IEEE)

7.5.1 組織概要

Institute of Electrical and Electronics Engineers (IEEE) は、電気・電子・情報通信分野を中心とする国際的な専門家組織であり、標準化組織である IEEE Standards Association (IEEE SA) を通じて多数の技術標準を策定している。通信ネットワーク、無線通信、ネットワークセキュリティ分野においては、IEEE 802 シリーズをはじめとする規格群が広く利用されている。PQC への移行については、通信プロトコルおよびネットワーク基盤への適用を主な対象として検討が進められており、既存の暗号方式との相互運用性を維持しながら段階的に PQC を導入する観点から、ハイブリッド方式を含めた検討が行われている。

7.5.2 PQC およびハイブリッド構成に関する標準化動向

IEEE における PQC およびハイブリッド方式に関する標準化の検討は、主として通信プロトコルおよびネットワークセキュリティ分野を中心に進められている。無線 LAN 技術を対象とする IEEE 802.11 では、将来の量子計算機による暗号解読リスクを踏まえ、既存の公開鍵暗号方式と PQC 方式を併用する移行の中間時期の形態が検討対象となっている。IEEE 802.11 TGbt に提出された文書「Proposed Texts for Hybrid PQC」[17]では、認証フェーズで交換される管理フレームにおいて、従来の DH パラメータに加えて、ML-KEM のパラメータを格納可能とする拡張が提案されている。これは認証フレームというフレーム種別を変更するものではなく、同フレーム内に含めるパラメータ要素 (elements) を拡張する形でハイブリッド鍵共有を実現するものである。本提案では、既存の RSN (Robust Security Network) および AKM (Authentication and Key Management) の枠組みを保持しつつ、その拡張としてハイブリッド方式を導入する設計が示されている。これにより、PQC を利用しない既存端末との相互運用性を保ちながら、DH と ML-KEM の双方から得られる複数の鍵素材を KDF に投入してセッション鍵を生成する構成が整理されている。このような設計は、移行期において耐量子計算機性を確保しつつ、既存実装の構造や端末との実装互換性を維持することを目的としている。

一方、IEEE Standards Association において検討が進められている IEEE SA P1943 (Standard for Post-Quantum Network Security) [43]は、特定の通信プロトコルに限定せず、ネットワークセキュリティ全般を対象とした耐量子計算機化の枠組みを整理する標準である。P1943 では、鍵共有および認証における PQC の適用や、完全な PQC 移行に至るまでの過渡期におけるハイブリッド方式の位置づけが整理されている。これらの検討は、IEEE 802 系列規格における個別のプロトコル仕様検討を補完する位置づけとして整理されている。

7.5.3 技術仕様におけるハイブリッド構成の詳細

IEEE 802.11 Submission: Proposed Texts for Hybrid PQC [17]では、IEEE 802.11 における耐量子計算機化対応として、既存の鍵共有方式と PQC 方式を併用するハイブリッド構成が具体的な仕様変更案として示されている。本提案では、本提案では、認証フェーズにおいて DH による共有秘密 DH_ss と ML-KEM による共有秘密 SS_pq を双方生成し、最終的にセッション鍵を導出する Hybrid Key Derivation が定義されている

この構成を無線 LAN のフレーム交換に統合するため、以下の 2 つの仕様拡張が提案されている。

(1) AKM Suite Selector の拡張：ハイブリッド構成を明示的に示す新しい AKM Suite を追加し、RSNE／RSNXE 内で交渉可能とする。

(2) RSN Extension (RSNXE) の拡張：ML-KEM のパラメータセット (ML-KEM-768/1024) および DH Group Identifier を広告するための新規フィールドが追加される。

さらに、従来の Diffie-Hellman Parameter element を拡張し、Diffie-Hellman and ML-KEM Parameter element として再定義することで、

- DH の有限体群識別子 (Finite Cyclic Group field)
- ML-KEM パラメータセット

の双方を格納し、両パラメータの組み合わせに対して Hybrid Parameter Identifier を割り当てる構造が示されている。これにより CNSA 2.0 で要請される ML-KEM-1024 や、IoT 端末向けの ML-KEM-768 を、DH と組み合わせて選択可能となる

また、認証および鍵導出処理の全体フローは 802.1X 認証および Fast Transition (FT) に対応する形で体系化されており、Hybrid PQC に対応した以下の手順が規定されている：

- DH 公開鍵の妥当性検証 (Finite Cyclic Group の整合性検証を含む)
- ML-KEM パラメータ検証 (RSNXE／Parameter element に基づく整合性確認)
- Hybrid Key Material の整合性確認 (DH_ss と SS_pq の生成可否チェック)

図 7-7 では、Authentication frame#1/#2 における DH_s_pub、MLKEM_s_pub の送信、AP による公開鍵検証、ML-KEM を用いたカプセル化の実行、STA による ML-KEM を用いたデカプセル化を例示する。

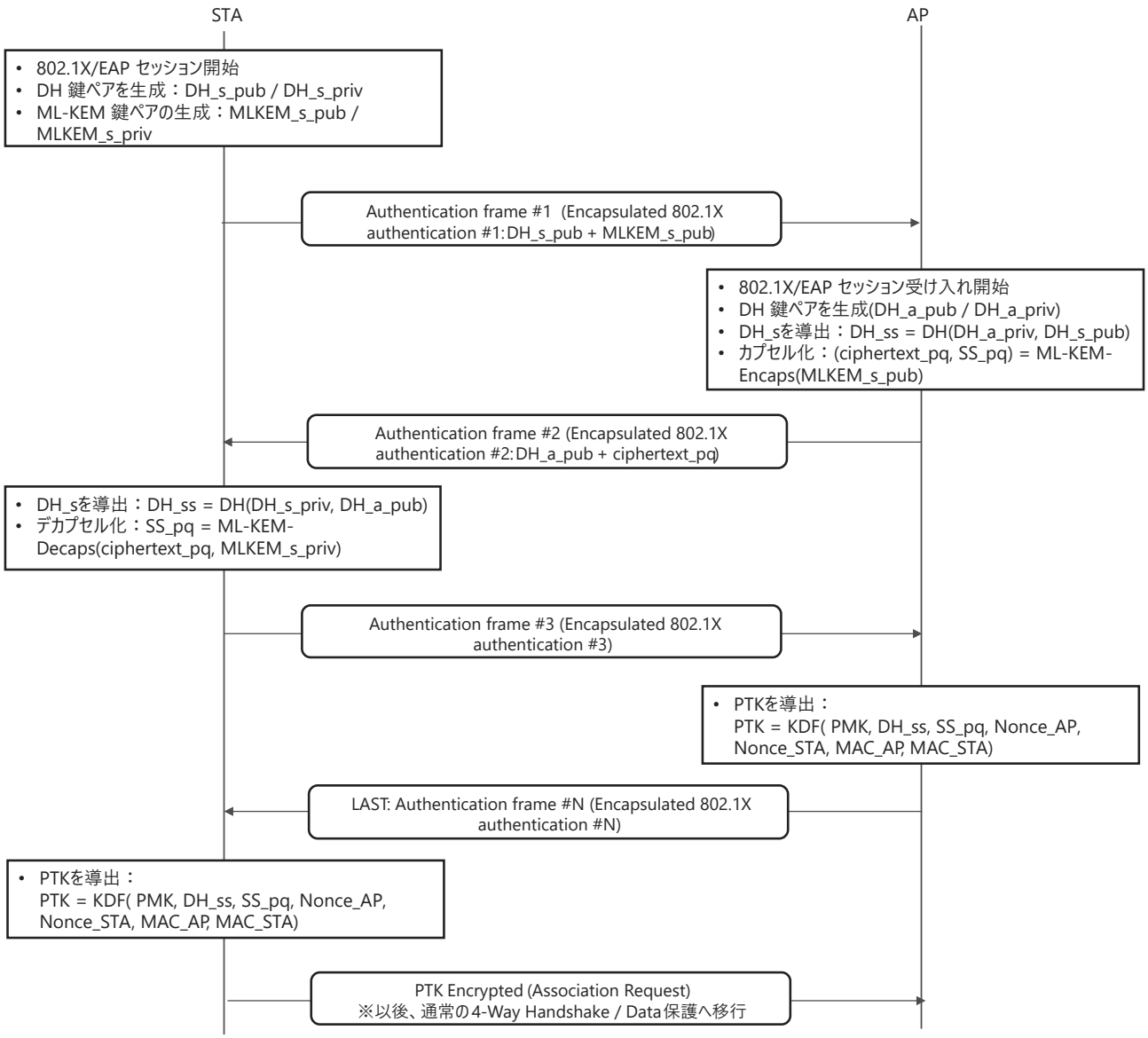


図 7-7 Hybrid PQC を適用した認証フレーム交換例。IEEE 802.11 Submission [17]の図を基にしつつ、原図では省略されている処理（EAP セットアップ、DH_ss／SS_pq の具体的導出、PTK 導出式など）を日本語で補足した再構成図である。

7.6 International Organization for Standardization (ISO)

7.6.1 組織概要

国際標準化機構（ISO）は、国際的な標準の策定を通じて、製品およびサービスの相互運用性や品質確保を目的とする国際標準化団体である。情報技術分野については、国際電気標準会議（IEC）との合同技術委員会である ISO/IEC JTC 1 が所管しており、暗号およびセキュリティ、プライバシ保護に関する標準化は、その下位分科会である SC 27（Information security, cybersecurity and privacy protection）が担当している。SC 27 の中でも、暗号アルゴリズム、鍵管理、デジタル署名などの暗号技術を扱うのが WG 2（Cryptography and Security Mechanisms）であり、ISO/IEC 18033（暗号アルゴリズム）、ISO/IEC 14888（デジタル署名）、ISO/IEC 11770（鍵管理）などの国際規格シリーズの策定および維持を行ってきた。PQC については、量子計算機の将来的な実用化を見据え、2015 年頃から検討が進められており、SC 27 Journal 等を通じて、標準化に向けた準備状況や基本的な考え方が共有されている。

7.6.2 PQC およびハイブリッド構成に関する標準化動向

ISO/IEC JTC 1/SC 27 における PQC の標準化は、新たな暗号方式を単独で規定するというよりも、既存の暗号標準体系の中に耐量子計算機性のある暗号方式を段階的に取り込む形で進められている。この取り組みの基盤整備として、SC 27/WG 2 により策定された Standing Document (SD8) である。SD8 では、ハッシュベース署名、格子ベース暗号、符号ベース暗号、多変数暗号、イソジェニー暗号といった PQC メカニズムの主要な５つのカテゴリについて、基本概念、安全仮定、代表的方式例が整理されている。

SC 27 Journal (Vol.1, Issue 3, 2022) [44]に掲載された「Paving the Runway for Standardization of Post-Quantum Cryptography」では、量子計算機の進展が既存の公開鍵暗号方式に与える影響を整理するとともに、PQC 移行に向けた取り組みの方向性が示されている。本記事では、PQC メカニズムの主要なカテゴリや、それらに関連する安全性の前提、実装上の特性などを整理することが、今後の標準化活動に向けた基盤的なステップとして紹介されている。一方、TLS や X.509 証明書のような具体的なプロトコル仕様については、本稿では取り扱われておらず、SC 27 では暗号プリミティブの枠組み整理や基礎的事項の把握が中心的な役割として位置付けられている。

この方針に沿い、ISO においては、PQC と既存の暗号方式を併用するハイブリッド方式についても、具体的な構成や鍵・署名の組合せ方法を技術仕様として規定していない。これらは主として IETF や ETSI などの標準化機関において議論・仕様化が進められており、ISO/IEC SC 27 はリエゾンを通じてそれらの動向と整合を図りつつ、暗号プリミティブの枠組み整理や概念レベルでの整理を担う位置付けとなっている。

一方で、PQC メカニズムの一部については、既存規格の分冊として規格化が進められている。具体例として、ISO/IEC 14888-4 (Stateful hash-based mechanisms) があり、ハッシュベース署名を既存のデジタル署名の規格体系に取り込む取り組みが行われている。しかし、複数の暗号方式を同時に利用するハイブリッド構成そのものについては、ISO 規格として規範的な定義は与えられていない。

以上より、ISO における PQC およびハイブリッド方式への取り組みは、NIST、IETF、ETSI 等が策定する具体的な技術仕様や移行ガイダンスを補完する形で、国際標準としての暗号技術体系全体の整合性を確保する役割を担っていると整理できる。

7.7 ANSI Accredited Standards Committee X9 (ASC X9)

7.7.1 組織概要

ANSI Accredited Standards Committee X9 (ASC X9) は、米国金融サービス業界向けの標準を策定する ANSI 認定標準化団体であり、1974 年に設立された。銀行・証券・保険・決済事業者などの金融業界を対象に、決済処理、資金移動、証券決済、カード取引、クリアリングなどの金融業務に関わる、暗号技術、公開鍵基盤 (PKI)、データ保護、認証および情報セキュリティ管理に関する標準や指針を策定してきた。

ASC X9 は ISO/TC 68 (金融サービス) における米国の国内審議団体として国際標準化活動にも関与しており、国内標準と国際標準との整合性を考慮した活動を行っている点に特徴がある。中でも X9F 委員会は暗号および情報セキュリティ分野を担当し、金融業界が直面する実務的なセキュリティ課題に対応した文書を多数公表してきた。

近年は量子計算機の進展を受け、既存の暗号方式の長期的安全性に対するリスクへの対応を重要

な検討課題として位置づけている。特に金融分野では、長期的なデータ保護や業界横断的な相互運用性確保が不可欠であることから、PQC への移行に関する調査・検討を段階的に進めている。

7.7.2 PQC およびハイブリッド構成に関する標準化動向

ASC X9 における PQC への対応は、NIST が主導する PQC アルゴリズム標準化を技術的前提としつつ、金融業界における移行準備とリスク管理の観点から整理されている点に特徴がある。2025 年に公開された「Post-Quantum Cryptography Financial Readiness Needs Assessment」 [45]は、ASC X9F による Informative Report として、金融機関が PQC 移行に向けて検討すべき事項を体系的に整理した文書である。

同レポートでは、量子計算機の実用化時期に不確実性が残る一方で、公開鍵暗号の長期的安全性が将来的に損なわれる可能性を前提に、移行準備を早期に開始する必要性が指摘されている。特に金融分野では、取引データや顧客情報など長期にわたり保護すべき情報が多く、Harvest Now, Decrypt Later（HNDL）攻撃への対応が重要な検討課題として位置づけられている。

このような前提の下、ASC X9 は PQC 移行を単純なアルゴリズム置換としてではなく、組織全体の暗号利用状況を把握した上で段階的に進めるべき長期的プロセスとして整理している。具体的には、PKI、TLS、決済ネットワーク、外部委託先を含む暗号利用環境について、暗号資産の棚卸しおよびリスク評価を実施し、影響範囲や優先順位を明確にすることが出発点として示されている。

また、移行期間が長期化することを前提に、将来的な標準やアルゴリズムの変更に柔軟に対応可能な暗号アジリティの確保が重要な要素として言及されている。特定の暗号方式や実装に強く依存した構成は、PQC 移行やその後の標準更新において運用上の制約となる可能性があるため、暗号ライフサイクル管理やアーキテクチャ設計の観点からの見直しが必要とされている。

さらに、PQC およびハイブリッド方式の導入に伴う実務的影響として、相互運用性、性能、証明書サイズの増大といった課題が整理されている。金融インフラでは、単一組織内にとどまらず業界全体での移行方針や実装方針の整合性を確保しつつ、他機関やベンダとの相互運用性を維持しながら移行を進める必要があるとされている。ASC X9 は、ハイブリッド方式を最終形として固定化するのではなく、PQC への完全移行までの移行期における現実的なリスク低減手段として位置づけている点に特徴がある。

7.8 National Security Agency (NSA)

7.8.1 組織概要

米国国家安全保障局（National Security Agency: NSA）は、米国政府における国家安全保障分野の通信・情報システムを担当する機関であり、暗号技術に関しては国家安全保障システム（National Security Systems: NSS）を対象とした標準化方針および運用ガイダンスを策定・提示する役割を担っている。NSA は、暗号アルゴリズムそのものを国際標準として策定する立場にはないものの、NIST が標準化する公開鍵暗号技術を前提として、NSS における利用要件、移行期限、運用上の制約条件を明確化する政策的・技術的指針を提示してきた点に特徴がある。従来は CNSA Suite 1.0 を通じて RSA や楕円曲線暗号を中心とした暗号スイートを規定してきたが、量子計算機の進展を背景として、PQC への移行を国家レベルで推進する方針を明確化している。

7.8.2 PQC およびハイブリッド構成に関する標準化動向

NSA における PQC 移行方針は、2022 年に公表された Commercial National Security Algorithm

Suite 2.0（CNSA Suite 2.0） [46]により明確化された。同文書では、将来的に既存の暗号方式に対する解析能力を有する量子計算機（Cryptographically Relevant Quantum Computer: CRQC）が実用化されることを前提として、NSS において使用される公開鍵暗号を既存の RSA や ECDH／ECDSA から耐量子計算機性を有する方式へ移行する必要性が示されている。CNSA Suite 2.0 では、一般用途の公開鍵暗号として NIST が標準化を進める CRYSTALS-Kyber（鍵カプセル化方式）および CRYSTALS-Dilithium（電子署名方式）を将来の必須アルゴリズムとして位置付けている。

注目すべき点として、NSA は PQC への移行を単純な一括置換ではなく、既存の暗号方式と PQC を併用するハイブリッド構成を前提とした段階的移行として整理している。プロトコル標準や製品成熟度、相互運用性の制約によっては、一定期間ハイブリッド方式の利用が許容、あるいは必要となることを明示しており、ハイブリッド方式を恒久的解決策ではなく移行期の現実的対処として位置付けている点が特徴である。

さらに、CSfC（Commercial Solutions for Classified）プログラムにおいては、2025 年に公表された CSfC Post Quantum Cryptography Guidance Addendum [47]により、TLS、IPsec、EAP-TLS など具体的なプロトコル単位での PQC およびハイブリッド方式の適用方針が整理されている。同ガイダンスでは、既存の鍵共有方式と ML-KEM を組み合わせた構成や、Pre-Shared Key を併用した耐量子計算機性確保の考え方が示され、運用面の現実性を重視した段階的な移行方針が示されている。

7.9 Cloud Security Alliance (CSA)

7.9.1 組織概要

Cloud Security Alliance（CSA）は、クラウドコンピューティング環境におけるセキュリティの向上を目的として 2009 年に設立された国際的な非営利団体である。クラウドサービス利用者、クラウドプロバイダ、セキュリティベンダ、研究機関など幅広いステークホルダーが参加しており、クラウド特有のリスクや運用課題に対するベストプラクティスの整理と普及を主な活動目的としている。CSA はアルゴリズムやプロトコルの標準仕様を策定する標準化機関ではなく、クラウド利用の実務者を主対象としたガイドライン、リスク評価手法、統制フレームワークを提供する立場にある点が特徴である。代表的な成果物として、Cloud Controls Matrix（CCM）や各種セキュリティホワイトペーパーが挙げられる。近年は量子計算機の実用化を見据え、PQC に関する実務的ガイダンスの提供にも注力している。

7.9.2 PQC およびハイブリッド構成に関する標準化動向

CSA における PQC への取り組みは、アルゴリズムやプロトコルの仕様策定ではなく、クラウド利用者が直面するリスク評価および段階的移行判断を支援する実務ガイダンスの提示に主眼が置かれている。代表的な文書である「A Practitioner’s Guide to Post-Quantum Cryptography (2025)」[48]では、量子計算機による将来的な既存の公開鍵暗号方式の解読リスク、とりわけ Harvest Now, Decrypt Later（HNDL）攻撃を背景として、クラウド環境における現実的な移行ステップが整理されている。CSA は、全面的な PQC 移行が短期間では困難である点を踏まえ、移行初期段階における有効なリスク低減策としてハイブリッド方式の活用を明確に位置付けている。具体的には、TLS 1.3、SSH、IPsec／VPN などの通信プロトコルを対象とし、既存の暗号方式と PQC アルゴリズムを組み合わせたハイブリッド鍵共有を優先的に導入すべき対象として挙げている。特に、TLS 1.3 における X25519 と ML-KEM（Kyber）を組み合わせたハイブリッド鍵共有は、移行期間中の代表的構成例として紹介されている。この方式では、複数の鍵素材を Combiner により結合し、いずれか一方の方式が安全であればセッション鍵全体の機密性を維持できる設計思想が採用されている。また CSA は、

証明書およびデジタル署名に関しても、Composite 証明書や Dual 署名といったハイブリッド署名モデルを検討することを推奨している。これらは既存の検証基盤との互換性を維持しつつ耐量子計算機性を段階的に導入するための現実的手法と位置付けられている一方、運用ポリシー不整合やダウングレード攻撃といった新たなリスクについても注意が必要であるとされている。総じてCSAは、NISTやIETFが策定する技術標準を前提としつつ、クラウド実運用の観点からPQCおよびハイブリッド方式導入時の判断指針を補完する役割を果たしている。

7.10 PQCRYPTO

7.10.1 組織概要

PQCRYPTOは、欧州委員会Horizon 2020（プロジェクト番号ICT-645622）の支援を受け、2015年3月から2018年にかけて実施された欧州主導の研究プロジェクトである。量子計算機の実用化によって既存の公開鍵暗号方式が破綻するリスクを背景に、長期的に安全なPQCの研究推進と国際標準化への橋渡しを主要目的とした。協調機関はアイントホーフェン工科大学（TU Eindhoven）であり、複数の欧州大学・研究機関が参画した。プロジェクトでは、候補暗号技術の整理・評価に加え、標準化団体との連携を担うワークパッケージ（WP5）が設けられ、ISO/IEC、ETSI、IEEE、IETF/IRTF等の標準化活動への貢献と情報発信が体系的に進められた。プロジェクト自体は2018年に終了しているが、その知見と人的ネットワークは現在のPQC標準化に継続的な影響を与えている。

7.10.2 PQCおよびハイブリッド構成に関する標準化動向

PQCRYPTOは、個別技術仕様を策定する立場ではなく、PQCの標準化初期段階において、各標準化団体の活動を横断的に整理し、研究成果を標準化議論へ投入する役割を果たした。最終報告書（Deliverable D5.2）[49]では、ETSI、NIST、ANSI X9、IEEE、IETF/IRTF、ISO/IEC JTC 1/SC 27、ドイツ連邦情報セキュリティ庁（BSI）などにおけるPQC関連動向が体系的に整理されている。特に、ハッシュベース署名（XMSS、LMS）や格子暗号を中心とするPQC候補技術の成熟状況、ならびに既存セキュリティ基盤との統合可能性が主要論点として扱われた。

ハイブリッド方式に関しては、量子計算機が直ちに実用化されない一方で「保存して後に解読（Harvest-then-Decrypt）」攻撃のリスクが顕在化しつつあるという問題意識の下、移行期における現実的な安全確保手段として位置付けられている。D5.2では、IETFにおけるTLS 1.3向け耐量子計算機性を有する安全なハイブリッド鍵共有案や、ハイブリッド暗号方式におけるアルゴリズム選定指針などの検討状況が紹介されている。これらは、既存の暗号方式とPQCを併用し、複数の鍵素材や署名を構成する複数の要素(署名値・認証パス・付随パラメータなど)を組み合わせることで、少なくとも一方が安全であれば全体の安全性を維持する設計思想に基づくものであり、段階的移行と後方互換性の確保を両立する点に特徴がある。

またPQCRYPTOは、ETSIのQuantum-Safe Cryptography(QSC)ワークショップへの継続的参加や、ISO/IEC JTC 1/SC 27（WG2）とのリエゾン確立を通じて、欧州を中心とした標準化議論に直接関与した。ISO/IECでは、WG2におけるPQCの検討に対し「Initial recommendations of long-term secure post-quantum systems」を提供し、後続のスタンディングドキュメント（WG2/SD）策定に寄与している。さらに、NISTのPQC標準化プロセスに対しても、プロジェクト参加者が提案アルゴリズムの開発・評価・攻撃解析のいずれの側面でも関与し、グローバルな合意形成に影響を与えた。

以上のようにPQCRYPTOは、PQCおよびハイブリッド方式を「標準として定義する主体」ではなく、「研究成果を国際標準化活動へ橋渡しする触媒」として機能した点に特徴がある。2018年のプロジェクト終了後も、D5.2で整理された標準化課題や設計観点は、ETSI技術仕様、IETF RFC／イ

ンターネットドラフト、NIST SP／IR 等に引き継がれており、現在の PQC 移行戦略を理解する上で重要な歴史的参照点となっている。

7.11 Post-Quantum Cryptography Coalition (PQCC)

7.11.1 組織概要

Post-Quantum Cryptography Coalition (PQCC) は、量子計算機の実用化に伴う暗号リスクに対応することを目的として設立された、産学官横断の国際的アライアンスである。PQCC は特定の標準化機関とは異なり、アルゴリズムやプロトコルの規格制定を直接行う立場ではない。一方で、ソフトウェアベンダ、クラウド事業者、暗号ライブラリ開発者、研究者などが参加し、PQC 移行に関する実践的課題の整理、ユースケース分析、実装上の留意点の共有を行っている点に特徴がある。特に、OS イメージやライブラリといった既存のデジタル資産（ソフトウェア資産）が広範に展開されているソフトウェア流通・サプライチェーンを対象とし、PKI やコード署名(ソフトウェアの真正性証明)における量子コンピュータによる暗号解読リスク低減を重要テーマとして位置付けている。PQCC の議論は、将来の正式標準を先取りする形での設計判断や運用上の知見を提供する役割を担っており、NIST や IETF 等の標準化動向を実装・運用の観点から補完する存在といえる。

7.11.2 PQC およびハイブリッド構成に関する標準化動向

PQCC は、PQC の長期安全性に対する不確実性と、既存の PKI やソフトウェア署名基盤が直ちに全面刷新できない現実を踏まえ、移行期におけるハイブリッド方式の必要性を強調している。PQCC が公開している文書「Artifact Signing: Dual Post Quantum / Traditional Hybrid Signatures and Downgrades」[50]では、ソフトウェア配布物やアップデートファイルの署名（Artifact Signing）を対象に、PQC と現在の暗号方式を併用するデュアル署名型ハイブリッド方式の設計意義とリスクが整理されている。同文書では、単一の PQC 署名への一足飛びの移行は現実的ではなく、既存検証基盤との互換性を維持しつつ耐量子計算機性を付加できる段階的移行が実務上不可欠であるとの立場が示されている。

具体的には、既存の署名方式（例：RSA／ECDSA）と PQC 署名（例：ML-DSA）を同一成果物に対して並列に付与するデュアル署名構成を採用し、検証側は両方、もしくは少なくとも一方を検証できる柔軟性を持つ設計が議論されている。これにより、PQC 対応が未整備なクライアント環境でも既存の署名方式による署名検証が可能となる一方、量子計算機出現後を想定した耐量子計算機性の検証経路も確保される。

ただし PQCC は、この柔軟性が逆にダウングレード攻撃や検証ポリシーの不整合を招く可能性があることを明示的に指摘している。具体的な失敗シナリオとしては、攻撃者が意図的に PQC 署名を除去し、脆弱となった既存署名のみを提示した場合に、検証者が（互換性維持のための設定により）それを正当なものとして受理してしまうケースが挙げられる。このように、検証ポリシーの設定が不適切であれば、PQC 署名が実質的に無視され、耐量子計算機性が失われるリスクがある。

そのため PQCC 文書では、ハイブリッド署名を導入する際には、技術仕様そのものよりも、検証ポリシー、署名優先順位、失効・更新時の運用ルールが安全性を左右すると整理している。この観点は、NIST や IETF が主にアルゴリズムや構成方式を定義しているのに対し、PQCC が運用面での失敗シナリオを具体的に提示している点に特徴がある。また、PQC アルゴリズム自体も標準確定直後は実装成熟度が十分でない可能性があるため、単独 PQC 署名への即時依存を避ける意味でも、移行期のハイブリッド方式はリスク分散手段として有効であると結論付けている。

8. 調査結果に関する考察

2020 年時点では、ハイブリッド構成は概念レベルに留まり、標準化活動は議論開始段階にあった。しかし、2025 年から 2026 年初頭にかけて、主要な標準化機関により以下の事実が確認される。

- NIST
  ML-KEM および ML-DSA を FIPS 203 および FIPS 204 として標準化。SP 800-227 で Multi-Algorithm KEM および PQ/T ハイブリッドの設計指針を提示。SP 800-56C Rev.2 で複数の共有秘密(Z, T)を連携した Z’＝Z || T を KDF の入力として扱う方法を規定。SP 1800-38C により TLS や X.509 を対象とした相互運用性評価を報告。
- IETF
  TLS WG が draft-ietf-tls-hybrid-design-16 で TLS 1.3 におけるハイブリッド鍵共有を定義。LAMPS WG が Composite 署名・証明書モデルを策定（draft-ietf-lamps-pq-composite-kem-12、draft-ietf-lamps-pq-composite-sigs-14）。PQUIP WG は RFC 9794 で PQ/T ハイブリッド用語を標準化し、Hybrid signature spectrums を提示。
- ETSI
  TS 103 744 でハイブリッド鍵共有および CatKDF／CasKDF による複数の共有秘密の統合方式を仕様化。TR 103 966 で移行設計上の考慮事項を整理。
- NSA
  CNSA Suite 2.0 で Kyber＋ECDH 構成を推奨。CSfC ガイダンスで TLS、IPsec におけるハイブリッド構成の適用方針を明示。
- その他の機関
  IEEE は 802.11 におけるハイブリッド鍵共有の提案を提示し、P1943 でネットワークセキュリティ全般の耐量子計算機化枠組みを検討。ISO/IEC は概念整理とリエゾン調整を担当し、具体的なハイブリッド仕様は定義していない。CSA はクラウド環境におけるハイブリッド導入ガイドを公表。PQCC はソフトウェア署名におけるデュアル署名モデルを提示。

2026 年 1 月時点で、ハイブリッド構成は主要標準化機関により技術仕様として確立され、TLS、X.509 などのプロトコルにおいて、具体的な実装指針が整備されている。

- TLS1.3 の例
  draft-ietf-tls-hybrid-design-16 により、複数 KEM の公開鍵・暗号文を連結して key_share に配置し、双方が得た shared_secret を HKDF で統合する手順が明確化され、ハイブリッド鍵共有が実装可能な形で定義されている。
- X.509／PKI の例
  LAMPS WG による Composite 署名および Composite KEM、代替署名拡張(Alternative Signature Algorithm) により、PQC 方式と従来方式を単一証明書内で合成また混成できる構造が仕様レベルで整備され、移行期の証明書運用における選択肢が確立した。
- ハイブリッド鍵共有方式（ETSI TS 103 744）の例
  ECDH＋ML-KEM のハイブリッド鍵共有方式、並びに CatKDF／CasKDF を用いた共有秘密の結合方法が規定され、プロトコル実装者が参照可能な手順として提示されている。

これらの具体的な整備により、ハイブリッド構成は単なる概念段階を超えて、既存プロトコル・証明書・運用基盤の中で「移行期に採用可能な実装構成」として体系的に確立されたと評価できる。

9. 参考文献

[1] CRYPTREC 暗号技術調査ワーキンググループ（耐量子計算機暗号），“CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）2024 年度版,” 3 2025. [オンライン]. Available: https://www.cryptrec.go.jp/report/cryptrec-gl-2007-2024.pdf.

[2] CRYPTREC 暗号技術調査ワーキンググループ（耐量子計算機暗号），“耐量子計算機暗号の研究動向調査報告書,” 3 2025. [オンライン]. Available: https://www.cryptrec.go.jp/report/cryptrec-tr-2001-2024.pdf.

[3] レピダム，“ハイブリッドモードの技術動向調査,” 12 2020. [オンライン]. Available: https://www.cryptrec.go.jp/exreport/cryptrec-ex-3004-2020.pdf.

[4] NIST, “Mechanisms Recommendations for Key-Encapsulation Mechanisms,” 9 2025. [オンライン]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-227.pdf.

[5] デジタル庁・総務省・経済産業省，“電子政府における調達のための参照すべき暗号のリスト（CRYPTREC 暗号リスト），” 16 5 2023. [オンライン]. Available: https://www.cryptrec.go.jp/list/cryptrec-ls-0001-2022r1.pdf.

[6] K. Moriarty, B. Kaliski, J. Jonsson , A. Rusch, “PKCS #1: RSA Cryptography Specifications Version 2.2,” 11 2016. [オンライン]. Available: https://datatracker.ietf.org/doc/html/rfc8017.

[7] R. Housley , S. Turner, “Use of the RSA-KEM Algorithm in the Cryptographic Message Syntax (CMS),” 2 2025. [オンライン]. Available: https://datatracker.ietf.org/doc/rfc9690/.

[8] PQCC, “Post-Quantum Cryptography (PQC) Migration Roadmap,” 5 2025. [オンライン]. Available: https://pqcc.org/wp-content/uploads/2025/05/PQC-Migration-Roadmap-PQCC-2.pdf.

[9] NIST, “NIST Special Publication 800-56C Recommendation for Key-Derivation Methods in Key-Establishment Schemes, ” 8 2020. [オンライン]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr2.pdf.

[10] D. Connolly, R. Barnes , P. Grubbs, “Hybrid PQ/T Key Encapsulation Mechanisms,” 20 10 2025. [オンライン]. Available: https://datatracker.ietf.org/doc/draft-irtf-cfrg-hybrid-kems/07/.

[11] M. Ounsworth, J. Gray, M. Pala, J. Klaußner , S. Fluhrer, “Composite ML-KEM for use in X.509 Public Key Infrastructure, ” 7 1 2026. [オンライン]. Available: https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-kem/12/.

[12] M. Ounsworth, J. Gray, M. Pala, J. Klaußner , S. Fluhrer, “Composite ML-DSA for use in X.509 Public Key Infrastructure, ” 8 1 2026. [オンライン]. Available: https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-sigs/14/.

[13] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3,” 7 3 2020. [オンライン]. Available: https://datatracker.ietf.org/doc/rfc8446/.

[14] D. Stebila, S. Fluhrer , S. Gueron, “Hybrid key exchange in TLS 1.3,” 18 11 2025. [オンライン]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/16/.

[15] X. Tian, B. Hale, M. Mularczyk , Joël, “Amortized PQ MLS Combiner,” 4 11 2025. [オンライ

ン]. Available: https://datatracker.ietf.org/doc/draft-ietf-mls-combiner/02/.

[16] ETSI, “ETSI TS 103 744 V1.2.1 CYBER; Quantum-Safe Cryptography (QSC); Quantum-safe Hybrid Key Establishment, ” 3 2025. [オンライン]. Available: https://www.etsi.org/deliver/etsi_ts/103700_103799/103744/01.02.01_60/ts_103744v010201p.pdf.

[17] IEEE, “ Proposed Texts for Hybrid PQC, ” 25 11 2025. [オンライン]. Available: https://mentor.ieee.org/802.11/dcn/25/11-25-2051-01-00bt-proposed-texts-for-hybrid-pqc.docx.

[18] T. Okubo, C. Bonnell, J. Gray, M. Ounsworth , J. Mandel, “A Mechanism for X.509 Certificate Discovery,” 19 11 2025. [オンライン]. Available: https://datatracker.ietf.org/doc/draft-ietf-lamps-certdiscovery/02/.

[19] ITU, “ITU-T Recommendations,” 10 2019. [オンライン]. Available: https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=X.509.

[20] M. Ounsworth, “ “Architecting PKI Hierarchies for Graceful Post-Quantum Migration,” presented at the PKI Consortium Post-Quantum Cryptography Conference 2025,” 1 2025. [オンライン]. Available: https://pkic.org/events/2025/pqc-conference-austin-us/WED_BREAKOUT_1200_Mike-Ounsworth_Architecting-PKI-Hierarchies-for-Graceful-PQ-Migration.pd.

[21] J. Klaußner, “ “Hybrid PQC E-Mail Communication: Easing Migration Pain,” presented at the Post-Quantum Cryptography Conference 2025, ” 1 2025. [オンライン]. Available: https://pkic.org/events/2025/pqc-conference-austin-us/WED_BREAKOUT_1430_Jan-Klaussner_Hybrid-PQC-E-Mail-Communication-Easing-Migration-Pain.pdf.

[22] NIST, “NIST IR 8547 ipd Transition to Post-Quantum,” 11 2024. [オンライン]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf.

[23] F. D, M. P , B. Hale, “RFC 9794 Terminology for Post-Quantum Traditional Hybrid Schemes,” 13 6 2025. [オンライン]. Available: https://datatracker.ietf.org/doc/rfc9794/.

[24] N. Bindel, B. Hale, D. Connolly , F. D, “Hybrid signature spectrums,” 17 9 2025. [オンライン]. Available: https://datatracker.ietf.org/doc/draft-ietf-pquip-hybrid-signature-spectrums/07/.

[25] L. Chen, “Cryptographic Agility and Transition R&D and Plans,” 21 3 2024. [オンライン]. Available: https://csrc.nist.gov/Presentations/2024/cryptographic-agility-and-transition-rd-and-plans.

[26] Open Quantum Safe Project, “ Open Quantum Safe, ” [オンライン]. Available: https://openquantumsafe.org/.

[27] Open Quantum Safe Project, “liboqs: C library for post-quantum cryptography,” [オンライン]. Available: https://github.com/open-quantum-safe/liboqs.

[28] Open Quantum Safe Project, “ OQS-OpenSSL, ” [オンライン]. Available: https://github.com/open-quantum-safe/openssl.

[29] The Legion of the Bouncy Castle, “Bouncy Castle Crypto APIs,” [オンライン]. Available: https://www.bouncycastle.org/.

[30] wolfSSL Inc., “Hybrid Post-Quantum Key Exchange in wolfSSL,” [オンライン]. Available:

https://www.wolfssl.com/hybrid-post-quantum-key-exchange-in-wolfssl-5-8-0/.

[31] PKI Consortium, “Post-Quantum Cryptography Conference,” 15-16 1 2025. [オンライン]. Available: https://pkic.org/events/2025/pqc-conference-austin-tx/.

[32] PKI Consortium, “Post-Quantum Cryptography Conference,” 28-30 10 2025. [オンライン]. Available: https://pkic.org/events/2025/pqc-conference-kuala-lumpur-my/.

[33] M. Marcus, “"Real-World Post-Quantum Migrations: Lessons Learned and Performance Results,” presented at the Post-Quantum Cryptography Conference 2025,” 10 2025. [オンライン]. Available: https://pkic.org/events/2025/pqc-conference-kuala-lumpur-my/THU_P_1400_michiel-marcus_real-world-post-quantum-migrations-lessons-learned-and-performance-results_merged.pdf.

[34] S. Kelly, “The Internet Is Ready for Some PQC Certificates,” presented at the Post-Quantum Cryptography Conference 2025, ” 10 2025. [オンライン]. Available: https://pkic.org/events/2025/pqc-conference-kuala-lumpur-my/WED_B_1130_shane-kelly_the-internet-is-ready-for-some-pqc-certificates_merged.pdf.

[35] S. Barker, “X25519Kyber768: Paving the Way for Post-Quantum Security,” 21 9 2024. [オンライン]. Available: https://expertbeacon.com/x25519kyber768-paving-the-way-for-post-quantum-security/.

[36] Google, “Protecting Chrome Traffic with Hybrid Kyber KEM,” 10 8 2023. [オンライン]. Available: https://blog.chromium.org/2023/08/protecting-chrome-traffic-with-hybrid.html.

[37] NIST, “ Post-Quantum Cryptography, ” 3 1 2017. [オンライン]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/call-for-proposals.

[38] NIST, “ NIST SP 1800-38 Migration to Post-Quantum Cryptography: Preparation for Considering the Implementation and Adoption of Quantum Safe Cryptography,” 19 12 2023. [オンライン]. Available: https://csrc.nist.gov/pubs/sp/1800/38/iprd-(1).

[39] B. Halee, X. Tiane , L. Wang, “Benchmarking of the Amortized Post Quantum Combiner for MLS,” 8 1 2026. [オンライン]. Available: https://eprint.iacr.org/2026/034.pdf.

[40] ITU, “Overview of hybrid approaches for key exchange with quantum key distribution,” 20 5 2022. [オンライン]. Available: https://www.itu.int/dms_pub/itu-t/opb/tut/T-TUT-ICTS-2022-1-PDF-E.pdf.

[41] ETSI, “ETSI TR 103 966 V1.1.1 CYBER Security (CYBER);Quantum-Safe Cryptography (QSC); Deployment Considerations for Hybrid Schemes,” 10 2024. [オンライン]. Available: https://www.etsi.org/deliver/etsi_tr/103900_103999/103966/01.01.01_60/tr_103966v010101p.pdf.

[42] ETSI, “ETSI TS 104 015 V1.1.1 Cyber Security (CYBER); Quantum-Safe Cryptography (QSC); Efficient Quantum-Safe Hybrid Key Exchanges withHidden Access Policies,” 2 2025. [オンライン]. Available: https://www.etsi.org/deliver/etsi_ts/104000_104099/104015/01.01.01_60/ts_104015v010101p.pdf.

[43] IEEE, “Standard for Post-Quantum Network Security,” [オンライン]. Available: https://standards.ieee.org/ieee/1943/10957/.

[44] L. CHEN, “PAVING THE RUNWAY FOR STANDARDIZATION OF POST-QUANTUM CRYPTOGRAPHY,” *SC27 Journal,* 第 巻 Vol 1, 第 Issue 3, pp. pp. 11-19, 2 2022.

[45] ASC X9, “New X9 Report Supplies Guidance on Migrating to Post-quantum Cryptography Safely and Cost-effectively,” 8 2025. [オンライン]. Available: https://x9.org/new-x9-report-migrating-to-post-quantum-cryptography/.

[46] NSA, “Announcing the Commercial National Security,” 9 2022. [オンライン]. Available: https://media.defense.gov/2025/May/30/2003728741/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS.PDF.

[47] NSA, “CSfC Post Quantum Cryptography Guidance Addendum 1.0,” 4 4 2025. [オンライン]. Available: https://www.nsa.gov/Portals/75/documents/resources/everyone/csfc/capability-packages/CSfC%20Post%20Quantum%20Cryptography%20Guidance%20Addendum%201_0%20Draft%20_5.pdf.

[48] CSA, “A Practitioner’s Guide to Post-Quantum Cryptography,” 10 11 2025. [オンライン]. Available: https://cloudsecurityalliance.org/artifacts/a-practitioners-guide-to-post-quantum-cryptography.

[49] PQCRYPTO , “Post-Quantum Cryptography for Long-Term Security,” 9 4 2018. [オンライン]. Available: https://pqcrypto.eu.org/deliverables/d5.2-final.pdf.

[50] P. Kampanakis , D. V. Geest, “Artifact Signing: Dual, Post-Quantum/Traditional Hybrid Signatures and Downgrades,” 4 2025. [オンライン]. Available: https://pqcc.org/artifact-signing-dual-post-quantum-traditional-hybrid-signatures-and-downgrades/.

資料４

2025 年度　暗号技術活用委員会活動報告

1.　2025 年度の活動概要

1.1.　活動目的

暗号技術活用委員会では、情報システム全般のセキュリティ確保に寄与することを目的として、暗号の取り扱いに関する観点から、運用ガイドライン／ガイダンスの作成を行う。

2025 年度は、昨今の耐量子計算機暗号（PQC）をめぐる社会的動向を踏まえ、耐量子計算機暗号の取扱い基準や運用ガイドライン／ガイダンスにおける PQC の位置づけ・記載内容等についての検討を開始した。また、暗号鍵管理ガイダンスの拡充活動の一環として、クラウドサービスを利用したシステム構築を対象とする「クラウド鍵管理ガイダンス」作成のため、クラウド鍵管理ガイダンス WG を設置した。同 WG では、作成するガイダンスの位置づけや内容を検討し、2026 年度末でのガイダンス完成を目標に作業を進めた。

1.2.　活動概要

今年度の活動概要は以下のとおりである。

(1) PQC の扱いに関わる検討

PQC をめぐる社会的動向を踏まえ、PQC の取扱い基準や位置づけ・記載内容等について検討を開始した。具体的には各国政府・公的機関等が公表している「PQC への移行方針」や関連ガイドラインについて政策的側面から整理し、また「CRYPTREC 暗号リスト」での PQC に関する位置づけや移行ルールを変更すべきかどうかを検討した。これらの検討結果を踏まえて、暗号技術活用委員会としての見解を取りまとめた。

(2) クラウドにおける鍵管理ガイダンスの作成

クラウド鍵管理ガイダンス WG を設置し、クラウドサービスを利用した情報システムにおける暗号鍵管理のガイダンス作成を開始した。2026 年度末の完成に向けて、今年度はガイダンスの骨子を整理した。

(3) 「暗号鍵管理システム設計指針（基本編）」の改訂

暗号鍵管理システムの設計に関わる解説書として作成した「暗号鍵管理システム設計指針（基本編）（以下「設計指針」と表記）」及び副読本である「暗号鍵管理ガイダンス Part 1」「暗号鍵管理ガイダンス Part 2」について、設計指針の作成から約５年が経過したことに伴い、記載の古さや誤記の指摘などの問題があったため、改訂を行うこととした。改訂版は「設計指針 v1.1」とした。

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、2025 年度に開催された暗号技術活用委員会の開催状況は表 1-2 のとおりである。

表 1-1　暗号技術活用委員会　委員構成

委員長	松本　勉	国立研究開発法人産業技術総合研究所　フェロー 横浜国立大学　先端科学高等研究院　上席特別教授
委員	上原　哲太郎	立命館大学　情報理工学部　情報理工学科　教授
委員	垣内　由梨香	日本マイクロソフト　リージョナルセキュリティチーム リスクマネージャー
委員	菊池　浩明	明治大学　総合数理学部　先端メディアサイエンス学科　教授
委員	佐藤　直之	SCSK セキュリティ株式会社　コンサルティング本部 シニアプロフェッショナルコンサルタント
委員	佐藤　雅史	セコム株式会社　IS 研究所 デジタルプラットフォームディビジョン　主幹研究員
委員	須賀　祐治	株式会社インターネットイニシアティブ インターネットサービス事業本部　セキュリティ本部 セキュリティ情報統括室　シニアエンジニア
委員	田村　裕子	日本銀行　金融研究所　情報技術研究センター　企画役
委員	寺村　亮一	GMOサイバーセキュリティbyイエラエ株式会社 上席執行役員 CMO（最高管理責任者） サイバーセキュリティ事業本部　本部長
委員	三澤　学	三菱電機デジタルイノベーション株式会社 情報管理・セキュリティソリューション統括室　セキュリティ技術部 製品セキュリティソリューション　グループマネージャー
委員	満塩　尚史	順天堂大学　健康データサイエンス学部　健康データサイエンス学科 准教授
委員	山口　利恵	東京大学　大学院情報理工学系研究科 ソーシャル ICT 研究センター　准教授
委員	渡邊　創	国立研究開発法人産業技術総合研究所 サイバーフィジカルセキュリティ研究部門　研究部門長

（2026 年 2 月 24 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
第一回	2025 年 7 月 18 日	● 2025 年度暗号技術活用委員会活動計画の確認 ● 2025 年度クラウド鍵管理ガイダンス WG 活動計画の審議 ● PQC の扱いに関する検討 ● 「暗号鍵管理システム設計指針（基本編）」の修正について
第二回	2026 年 2 月 24 日	● PQC の扱いに関する検討 ● 2025 年度クラウド鍵管理ガイダンス WG 活動状況及び活動報告の審議 ● 「暗号鍵管理システム設計指針（基本編）」の修正案の審議 ● 2025 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

以下に成果概要を記載する。詳細については、CRYPTREC Report 2025 暗号技術活用委員会報告[1]を参照されたい。

2.1.　PQC の扱いに係る検討

①　「PQC への移行方針」の政策やガイドラインについて

各国政府・公的機関等が発行している PQC 移行に関する政策やガイドラインの情報を収集・最新化し、時系列的観点での分析を行った。

2025 年 4 月の G7 会合で「Accelerating the Transition to Quantum-Safe Communication: A Call for Global Collaboration and Action[2]」が取りまとめられたことを受け、G7 各国は協調して PQC への移行を進めるため、明確な期限を定めた移行スケジュールを策定し、進捗状況を管理することが謳われた。欧米諸国では、PQC 移行スケジュールの策定・改定が進められており、代表例として以下を整理した。

- US Executive Order 14306 (2025.6)
- NIST SP800-131A Revision 3 draft
- EU, “A Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography”
- 英国, “Timelines for migration to post-quantum cryptography”
- カナダ, “Roadmap for the migration to post-quantum cryptography for the Government of Canada (ITSM.40.001)”

[1] CRYPTREC Report 2025 暗号技術活用委員会報告, https://www.cryptrec.go.jp/promo_cmte.html
[2] https://www.think7.org/publications/accelerating-the-transition-to-quantum-safe-communication-a-call-for-global-collaboration-and-action/

このほかドイツ、フランス、オーストラリアなどでも同様の文書が公表されている。これらの情報をまとめたのが以下の図である。

各国とも、移行の優先度が高い、国家安全保障システムや高セキュリティシステムなどについては 2030 年頃、その他のシステムについては 2035 年を移行期限の目途としていることが分かる。一方で、既存暗号、特に現行の公開鍵暗号との関連では、利用を止めるのかどうか、ハイブリッド実装を活用するのかどうかについては国ごとに違いがある。

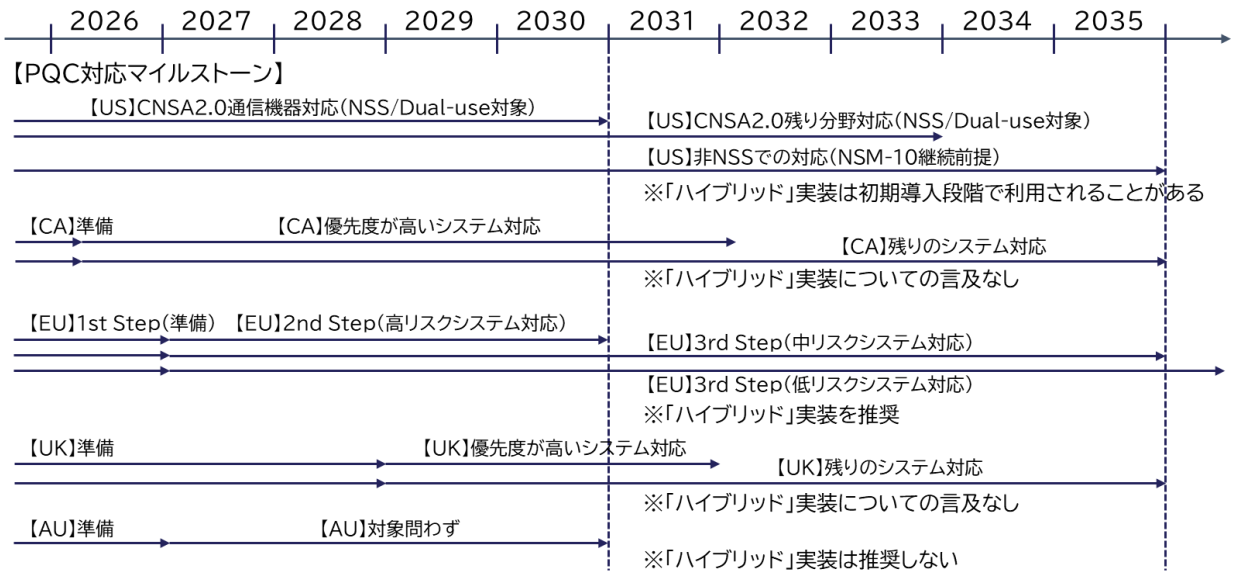


図 2-1　各国の PQC 移行スケジュール

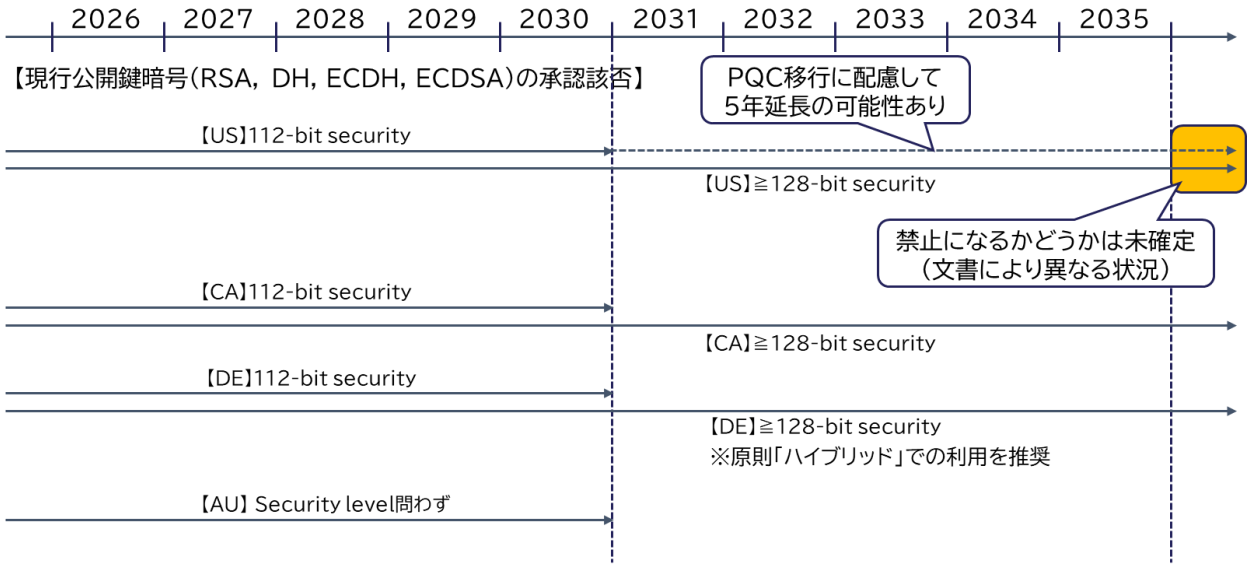


図 2-2　既存暗号（特に現行の公開鍵暗号）の利用可否

② 「CRYPTREC 暗号リスト」上の PQC の位置づけや移行ルールについて

PQC に対応した CRYPTREC 暗号リストの在り方について議論し、以下の通り、暗号技術活用委員会としての見解をまとめた。また、これらの見解を踏まえつつ作成された「耐量子計算機暗号（PQC）に対応した CRYPTREC 暗号リストの在り方（案）」の内容を確認し、暗号技術活用委員会として同意した。

- PQC を取り込んだ CRYPTREC 暗号リストの形式

  現行の CRYPTREC 暗号リスト（「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」）の技術分類をそのまま使うか、「PQC」カテゴリを追加するか、あるいは、それらのリストとは別の独立した「量子計算機耐性を備えた暗号方式に関わる独立したリスト（PQC リスト）」を作るべきかを議論した。

  その結果、現時点においては既存の CRYPTREC 暗号リストに PQC を加えるのではなく、新たに量子コンピュータに耐性のある暗号方式のみを記載する独立したリストを作成したほうがよいとの見解で一致した。

  また、新たなリストに掲載するアルゴリズムについて、量子計算機耐性を備えた公開鍵暗号だけではなく、共通鍵暗号やハッシュ関数なども含めて、量子計算機耐性を備えた暗号方式全般を含むリストとするのがよいとの見解で一致した。さらに、アルゴリズムの強度を規定するパラメータもリスト中に記載するのがよいとの意見もあった。これは、PQC リストを参照することで量子計算機耐性を備えた暗号方式を包括的に選択可能にし、利用者にとって利便性が高いと考えられるためである。

- 移行ルール・選定ルールについての検討

  「PQC リストを新たに作成すること自体は妥当であり、そのための移行・選定ルールが設けられることも理解できるが、電子政府推奨暗号リストとの関係は整理すべき」との指摘があった。これは、「電子政府推奨暗号リスト」を参照して暗号方式を選択するようにしている事例が政府機関のみならず民間事業者にも多いため、「電子政府推奨暗号リスト」の中に「現行暗号リスト」と新たな「PQC リスト」の２つのリストが併存する形になった場合、両リストの関係について分かりやすい説明をすべきとの観点での指摘である。例えば、両リストの使い分けをどうするのか、両リストは将来的に一本化されていくのか、両リストに掲載されるアルゴリズム、片方にしか掲載されていないアルゴリズムについてどのように考えればいいのか、PQC リストでも将来は監視暗号リストが必要である、などである。

  これらの意見は、「耐量子計算機暗号（PQC）に対応した CRYPTREC 暗号リストの在り方（案）」を作成する際の参考としてもらうこととした。

- 「耐量子計算機暗号（PQC）に対応した CRYPTREC 暗号リストの在り方について（案）」に対する意見

「耐量子計算機暗号（PQC）リスト」の説明文に関連して、例えば、現行の電子政府推奨暗号リストにのみ掲載されている方式について、いずれ PQC リストにも掲載される可能性があるのか、あるいは PQC リストには掲載されない方式であるのかが明確でない等の課題があるため、誤解が生じないように修正したほうがよいとの指摘があった。

また、ハイブリッドモードの取扱いについて、PQC リストではプリミティブとなるアルゴリズムだけにとどめるほうがよいのではないかとの意見があった。これは、ハイブリッドモードを含めると暗号リストとして複雑になることが予想されるため、例えば TLS 暗号設定ガイドラインのようなプロトコルやアプリケーションを対象に作成するガイドラインの中で対応するのがよいのではないかとの指摘である。

この点については、「耐量子計算機暗号（PQC）リスト検討タスクフォース（仮称）」にて検討を継続することとなった。

2.2. クラウドにおける鍵管理ガイダンスの作成

今年度はクラウド鍵管理ガイダンスの骨子となる資料を作成した。以下に、骨子の概要をまとめる。

クラウド鍵管理ガイダンスの目次案

本ガイダンスの目次案は以下のとおりである。

1. はじめに
2. 基礎知識
3. クラウド鍵管理サービスについて
4. クラウド鍵管理サービスに関わる責任分界
5. 暗号鍵管理システムのフレームワーク要求からの整理
6. その他

Appendix. 参考資料

1 章ではイントロダクションとして、本ガイダンスの位置づけや想定読者をまとめる。2 章では、クラウド鍵管理に関わる技術や政府システムにおける要件などの基礎知識をまとめる。3 章では、クラウド鍵管理サービスを体系化して比較する。4 章では、クラウド鍵管理サービスにおけるクラウドサービスプロバイダ（CSP）と利用者の責任分界の原則を説明する。5 章では、NIST SP 800-130 を基に鍵管理における管理策を抽出して、クラウド鍵管理サービス利用時の責任分界の適用例や利用者側の実施事項について説明する。6 章では、本ガイダンスでとりあげていない周辺事項に触れる。

以降では、ガイダンスの中核となる 1 章、3 章、5 章の概要を説明する。

「１．はじめに」の概要

本ガイダンスの目的、位置づけ、想定読者、スコープは以下のとおりである。

目的・位置づけ

- クラウドサービスの活用では、情報をクラウドサービスに預けることから情報の保存に関し、オンプレミスとは異なる考慮事項も生じる。クラウドサービスにおける暗号鍵管理サービスを適切に選択・構築・運用することによって、そのような事項に対処できる部分がある。クラウドサービスにおける暗号鍵管理の仕組みや注意事項をまとめた解説書を作成し、クラウド環境で安全に暗号を運用するための一つのガイダンスとする。
- CRYPTREC では、NIST SP 800-130（A Framework for Designing Cryptographic Key Management Systems）に基づいて暗号鍵管理システムを設計する際の解説書として「暗号鍵管理システム設計指針（基本編）」及び「暗号鍵管理ガイダンス」を作成しており、今回作成する解説書は、これらの CRYPTREC 文書を補強する位置づけにもなる。

想定読者

- クラウドサービスを利用した情報システムの構築者（SI 事業者等）、運用者。本ガイダンスで「利用者」と表記した場合は、これらの情報システム構築者や運用者を指す。

スコープ

- IaaS や PaaS のクラウドサービスを利用して、情報システムを構築するケースを対象に、クラウドサービスに保存するデータ及び鍵情報の機密性及び完全性を確保する観点から、どのようにクラウド鍵管理サービスを利用するかをターゲットとする。なお、SaaS や利用者が開発するアプリケーションにおいて、暗号鍵管理サービスの SDK を組み込んで実装する形態は、本ガイダンスの直接の対象外とする。

「３．クラウド鍵管理サービスについて」の概要

クラウドに利用者が保存するデータの保護を対象として、CSP が提供する鍵管理サービスを体系化する。それらの違いや選択時の考慮事項を整理する。

クラウド鍵管理サービスの分類

クラウド鍵管理サービスを分類する要素を示し、一般に使われるクラウドネイティブ鍵管理、BYOK（Bring Your Own Key）、HYOK（Hold Your Own Key）、BYOE（Bring Your Own Encryption）等の用語との関係を説明する。

図 2-3 に各方式の構成概要を示す。鍵の統制権、鍵の保管場所、データの暗号・復号処理の実行場所、の 3 点に着目することでそれぞれの特徴を整理できる。

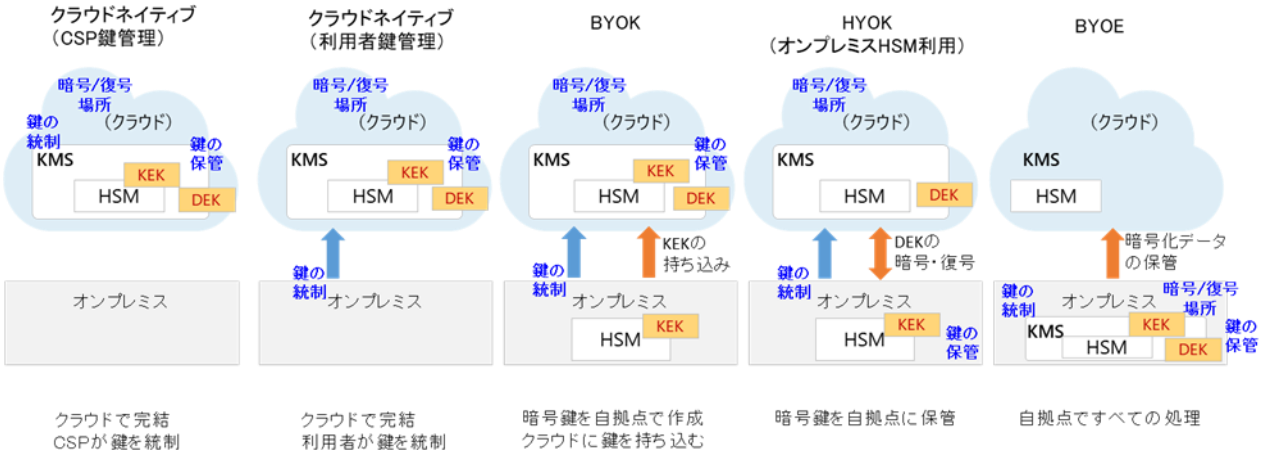


図 2-3　クラウド鍵管理サービスの構成概要

クラウド鍵管理サービスの比較

クラウド鍵管理サービスを比較し、選択時の考慮事項をまとめる。

一般に図 2-3 の右側の方式になるほど暗号鍵に関わる利用者の管理レベルが上がる一方で、利用者側の鍵管理基盤の構築や運用に関わるコストが増加する。各方式の中で「クラウドネイティブ（利用者鍵管理）」が選択上の基準となり、他の方式はそれぞれの方式の特徴に関わる要件が自組織のポリシーとして存在する場合に選択候補として考えるのが適当である。また、HYOK や BYOE は対象とする IaaS や PaaS におけるクラウドサービスとの連携がされていない場合が多く、選択できないケースがあることにも注意すべきである。

「５．暗号鍵管理システムのフレームワーク要求からの整理」の概要

クラウドサービスを導入した場合でも暗号鍵管理システムの設計・運用において検討すべきことは変わらないが、オンプレミスでの構築と比べて利用者が重点的に検討すべきことは変わる。本章では、利用者から見てどのような項目に重点が置かれるかを NIST SP 800-130 の検討項目の視点から整理する。

CKMS フレームワーク要求の適用について

NIST SP 800-130 のフレームワーク要求（FR）は CKMS 設計者を主対象に書かれており、FR として扱うと CSP が検討すべき内容が多数となる。そこで、同文書を元に暗号鍵管理システムの管理策として項目を整理する。このようにして、クラウド鍵管理サービスを利用する場合の責任分界の検討や利用者側の実施項目の抽出において参照可能となる管理項目一覧を作成する。

CKMS フレームワーク要求のクラウド鍵管理サービスへの適用例

上記のように作成した管理項目一覧を基に、クラウドネイティブ（利用者鍵管理）、BYOK、HYOK（オンプレミス HSM 利用）の 3 方式において利用者の実施項目を抽出した例を説明

する。ここでは、クラウドネイティブ（利用者鍵管理）における責任分界の例を表 2-1 に示す。

表 2-1　クラウドネイティブ（利用者鍵管理）における責任分界の例

管理項目	責任分界（実施責任）	クラウド鍵管理での実施内容
暗号アルゴリズム及びメタデータの選択		
暗号アルゴリズム・鍵長選択	[利用者]	DEKの暗号化及びデータ保護に利用する暗号アルゴリズム・暗号利用モード・鍵長は、CSPの提供方式の中から利用者が選択する。
メタデータの管理	[利用者]	メタデータ（特に鍵のタイプや保護対象とするデータ）はCSPの選択メニューから利用者が設定する。
ライフサイクルの定義と管理		
暗号機能の実行場所	CSP	KEK及びDEKを利用した暗号処理の実行場所はCSPの提供サービスに依存する。KEKを利用した暗号処理はクラウドHSM内で実行され、DEKを利用した暗号処理はクラウドサービス環境において実行される。
鍵生成	[利用者]	KEKの生成は利用者がクラウドKMSを操作して実施する。DEKはクラウド側で自動的に生成される。
鍵情報の破壊	[利用者]	KEKの破壊（消去）は利用者がクラウドKMSを操作して実施する。KEKの破壊（消去）によって、DEK及び保護対象データの暗号化消去が行われる。
鍵更新（＝鍵ローテーション）	[利用者]	KEKのローテーションに関わる更新周期やローテーションの自動/手動を利用者が設定する。自動ローテーションはクラウド側で実行され、手動ローテーションは利用者がクラウドKMSを操作して実施する。DEKのローテーションは原則として行われない。
鍵活性化、鍵非活性化、鍵失効、鍵の一時停止	[利用者]	KEKの非活性化や一時停止等の状態管理は、クラウドKMSの提供機能に基づいて利用者が操作して実施する。
鍵情報へのアクセスコントロール		
アクセスコントロールシステム	[利用者]	鍵情報へのアクセス権限は利用者が設定する。クラウドにおけるIAM機能と連携してアクセス管理が実施される。
鍵の保管・鍵の確立		
保管中の鍵情報のセキュリティ	CSP	KEK及びDEKのストレージ保管とそこで実施されるセキュリティは、CSPの提供サービスによる。KEKはクラウドHSMでの保管、DEKはKEKで暗号化してクラウドサービス環境における保管などの形態がとられる。
鍵情報のバックアップ	CSP	鍵情報のバックアップはCSPの提供サービスによる。
鍵情報のアーカイブ	CSP	古い世代の鍵情報の保存（アーカイブ）はCSPの提供サービスによる。
鍵情報の復元	CSP	バックアップやアーカイブされた鍵情報を復元して利用可能とする機能は、CSPの提供サービスによる。
鍵確立	CSP	クラウドシステム内やインターネットにおける各種通信に関わる保護機能（鍵確立を含む）は、CSPの提供サービスによる。
鍵の喪失・危殆化の対策		
鍵情報の喪失・破損時の対策	CSP	鍵情報の喪失や破損に関わる対策は、CSPの提供サービスによる。クラウドHSMを含めて冗長構成などがとられる。
鍵情報の危殆化時の対策	CSP	鍵情報の危殆化対策は、CSPの提供サービスによる。
その他（システムレベルの管理策）		
CKMSセキュリティポリシー	利用者	CKMSのセキュリティポリシーは利用者側の要件であり、それに適合したクラウド鍵管理サービスを選択する。クラウド鍵管理に関わるエンティティ及び役割は、CKMSセキュリティポリシーに基づいて利用者側が定める。
暗号モジュール	CSP	KEK管理に用いるHSMの暗号鍵保護メカニズムやHSMの第三者認証は、CSPの提供サービスによる。
CKMSのセキュリティコントロール	CSP	CKMSのシステムとしてのセキュリティコントロール（物理的セキュリティコントロール、コンピュータシステム・セキュリティコントロール、ネットワーク・セキュリティコントロール）はCSPの提供サービスによる。利用者鍵（KEK）の操作に関わるログ機能はCSPの提供サービスによる。
CKMSの障害・災害対策	CSP	CKMSのシステムとしての障害・災害対策はCSPの提供サービスによる。
将来的な移行対策	[利用者]	暗号アルゴリズム及び鍵長の移行（PQCを含む）、鍵確立プロトコルの移行、鍵管理デバイス（HSMなど）の移行といった対策は、CSPの提供サービスによる。暗号アルゴリズムや鍵長の移行のコントロールは、利用者が行う。

※[利用者]はCSPの基盤上で利用者側での操作・設定などの運用が要求される項目。[ ]がないものは運用以外に設計や構築も要求される項目。

<u>クラウド鍵管理サービスにおける利用者の管理事項</u>

利用者側の管理事項をより詳しく解説する。以下の事項等について説明を行う予定である。

- クラウド鍵管理方式の選択
- 暗号アルゴリズムと鍵の生成パラメータの設定、選択
- 鍵のライフサイクル

- 鍵へのアクセス管理
- ログ管理、監視

2.3. 「暗号鍵管理システム設計指針（基本編）」の修正

2020 年に発行した「暗号鍵管理システム設計指針（基本編）」の修正について検討した。

検討の背景は、2023－2024 年度に実施した「暗号鍵管理ガイダンス Part 2」の作成過程で、同ガイダンスの親文書に相当する設計指針に対する修正意見が幾つかあったことである。さらに、設計指針の執筆から約 5 年が経過し、記載内容の最新化や明確化を行った方がよい箇所や誤記も見つかっているため、修正すべき箇所と修正案を議論し、「設計指針 v1.1」として改訂を行う方針とした。

以下に、記載内容の最新化に関わる主な修正箇所をまとめる。この他の修正箇所については記載を省略する。

該当箇所	「設計指針」記載内容の問題	「設計指針」の修正方針及び修正案
「2.1 暗号鍵管理の必要性」本文（p.10）	「政府機関の情報セキュリティ対策のための統一基準（平成 30 年度版）」（平成 30 年 7 月 25 日、サイバーセキュリティ戦略本部）が参照されている。 また、**「「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。」**と書かれているが、最新の「統一基準」に暗号鍵の管理手順を定める記載はない。	最新の統一基準（令和 7 年度版）を参照する。また、暗号鍵の管理手順を定めることが基本対策事項として書かれている「政府機関等の対策基準策定のためのガイドライン（令和 7 年度版）の一部改定（令和 7 年 9 月）」を参照して、該当箇所の本文及び引用内容を更新する。 **「「対策基準策定のためのガイドライン」でも暗号鍵の管理手順を定めることが基本対策事項になっているように、…」**とする。
「2.2.4 Guidance」本文（p.15）、及び図 2-2（p.12）	Guidance 文書に**「リストガイド（鍵管理）」**と書かれているものがあるが、対象文書が明確でない。	「リストガイド（鍵管理）」を**「暗号強度要件（アルゴリズム及び鍵長選択）、暗号鍵設定ガイダンス」**の 2 文書に変更する。
「2.6.1 SP800-57」本文（p.24）	SP 800-57 Part 1 の最新の更新版 rev.5 が書かれていない。**「2005 年に初版が発行され、2006 年、2007 年、2012 年（revision 3）、2016 年（revision 4）に改訂されている。」**と書かれている。	**「…、2012 年、2016 年、2020 年（revision 5）に改訂されている。」**と修正する。

該当箇所	「設計指針」記載内容の問題	「設計指針」の修正方針及び修正案
「4.7 将来的な移行対策の必要性」 ② 本文（p.52）	暗号解読に関わる量子コンピュータやPQCについて2020年当時の状況が書かれている。現状を踏まえて更新した方がよい。 **「・新しい計算機技術の発展** **現状の脅威で最も高い関心が払われているものは、暗号鍵を復元するのに十分な能力を持つ量子コンピュータの発展である。** **例えば、大きなキュービットの量子コンピュータが構築されれば、既存の公開鍵暗号アルゴリズムのセキュリティが脅かされるかもしれず、これらのアルゴリズムに暗号鍵の確立を依存する CKMS に対して重大な影響を与える可能性がある。** **一方、量子コンピュータに耐性がある公開鍵暗号アルゴリズム（耐量子計算機暗号）についての研究や標準化が現在進行中であるが、現時点で広く受け入れられている解はまだ見出されていない。」**	以下の記載に変更する。 **「・新しい計算機技術の発展** **現状の脅威で最も高い関心が払われているものは、現在広く使われている暗号の秘密鍵を復元するのに十分な能力を持つ量子コンピュータの発展である。** **そのような量子コンピュータによる攻撃にも耐性がある公開鍵暗号（耐量子計算機暗号）について各国の組織で研究や標準化活動が進展しており、米国 NIST では複数の標準方式が選定された。」** さらに、脚注を設けて「CRYPTREC暗号技術ガイドライン（耐量子計算機暗号）」のリンクを記載する。 https://www.cryptrec.go.jp/tech_guidelines.html
「6.1.1 暗号アルゴリズムのセキュリティ強度」 本文及び表 6-1（p.79-81）	ビットセキュリティの表 6-1 を SP 800-57 Part 1 を参照して記載しているが、CRYPTREC では後に「暗号強度要件に関する設定基準」を発行している。同表には「強度要件」では推奨されていない方式やセキュリティ強度も記載されている。 また、表 6-1 について本文に以下のように書かれている。 **「そこで、暗号アルゴリズムの選択においては、“x ビットセキュリティ”の“x ビット”に着目して、長期的な利用期間の目安とする使い方ができる。例えば、NIST SP800-57 Part 1 revision 4 を参考にすると、電子政府推奨暗号リストに記載の暗号アルゴリズムのビットセキュリティは表 6-1 のように表現できる。」**	「暗号強度要件に関する設定基準」に沿った表記に更新する。 表 6-1 を強度要件に記載されているセキュリティ強度と暗号アルゴリズムに変更する。 さらに、表 6-1 に関わる本文を以下の記載に修正する。 **「「暗号強度要件に関する設定基準」において、CRYPTREC 暗号リストに掲載されている暗号アルゴリズムのビットセキュリティによるセキュリティ強度、並びに、電子政府システムにおけるセキュリティ強度要件の基本的な設定方針が示されている。同文書に基づいて CRYPTREC 暗号リストに掲載されている暗号アルゴリズムと基本的な利用期間の設定方針をまとめると表 6-1 のようになる。」**
「7.1 鍵情報の種類」 表 7-1 （p.83）	「表 7-1 鍵タイプ一覧」が SP 800-130(2012 年)に基づいて書かれており、表内に**「乱数生成プライベート鍵」**と**「乱数生成公開鍵」**が含まれているが、最新の SP 800-57 Part 1, rev.5(2020 年)ではこの 2 つの鍵タイプは削除されている。本文に**「SP800-130 で分類する鍵タイプは表 7-1 の通りである」**と書かれている。	表 7-1 から**「乱数生成プライベート鍵」**と**「乱数生成公開鍵」**を削除する。併せて、本文を**「SP 800-57 Part 1, rev.5 で分類する鍵タイプは表 7-1 のとおりである。」**と修正する。

3. 今後に向けて

2026 年度は以下の活動を実施する予定である。

1) 暗号強度要件ガイドラインに PQC の取り扱いを含める形で見直しを実施し、2026 年度末での完成を目指す。
2) TLS 暗号設定ガイドラインについては、PQC サポートに向けた TLS1.3 への移行や PQC 関連技術の取り込み、現行のセキュリティ例外型の削除など、現在の TLS 暗号設定ガイドラインの前提条件が大きく変わりつつことを踏まえ、記載内容の見直し検討を 2026 年度より開始する。2027 年度末でのガイドライン見直しの完成を目指す。
3) 引き続き、「クラウド鍵管理ガイダンス」の作成を進め、2026 年度末に完了する予定である。

政府機関等における耐量子計算機暗号（PQC）への移行について

資料５

- ✓ 量子計算機技術の進展に伴い、現在広く利用されている公開鍵暗号の安全性の低下・危殆化が予想
- ✓ 耐量子計算機暗号(PQC)への移行には、技術的課題のほか、安全保障、産業政策、サービス安定供給、対応支援策、国際連携など多岐にわたる課題に対応する必要

2025年６月30日　：　第１回 関係府省庁連絡会議の開催

⇒「政府機関等における耐量子計算機暗号（PQC）利用に関する関係府省庁連絡会議」で検討開始

会議構成員

議　長	内閣官房副長官補（内政担当）
副議長	内閣官房内閣審議官（国家安全保障局）、内閣官房内閣審議官（NCO）
主　査	デジタル庁統括官（デジタル社会共通機能担当）、総務省サイバーセキュリティ統括官、経済産業省商務情報政策局長
構成員	内閣官房内閣審議官（内閣官房副長官補付）、内閣府科学技術・イノベーション推進事務局統括官、警察庁長官官房技術総括審議官、デジタル庁統括官（戦略・組織担当）、外務省大臣官房サイバーセキュリティ・情報化参事官、文部科学省研究振興局長、経済産業省イノベーション・環境局長、防衛省大臣安房サイバーセキュリティ・情報化審議官

2025年11月19日　：　第２回 関係府省庁連絡会議の開催

⇒　中間とりまとめにおいて、政府機関等の移行に向けた工程表（ロードマップ）の骨子を策定

中間とりまとめの主な項目

○ **現状の整理**
- 量子計算機の開発・普及状況及びそれに伴い安全性が低下・危殆化する暗号技術の特定とその時期、諸外国の動向の把握、耐量子計算機暗号（PQC）の安全性等の評価・確認とその時期

○ **現状の整理を踏まえた移行期限、支援策等**
- 耐量子計算機暗号（PQC）への移行期限及び安全性が低下・危殆化した暗号技術の利用に係る停止の時期、政府機関等の移行への対応に必要な支援策等

○ **政府機関等の移行に向けた工程表(ロードマップ）の策定**
- 工程表（ロードマップ）の方向性、工程表（ロードマップ）に盛り込むべき事項等

○ **その他**

2025年12月23日　：　サイバーセキュリティ戦略（閣議決定）

⇒　**原則として、2035年までの移行**を目指し、**2026年度に工程表（ロードマップ）を策定**

政府機関等における耐量子計算機暗号（PQC）への移行について

✓ 関係府省庁連絡会議の中間とりまとめにおいて、政府機関等の耐量子計算機暗号（PQC）への移行に向けた**工程表（ロードマップ）の骨子を策定**

工程表(ロードマップ)の骨子（概要）

移行対象	• **「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」(※)の適用対象**となる情報システム ※サイバーセキュリティ基本法に基づく、**政府機関等（政府機関及び独立行政法人等）**の情報セキュリティ水準を維持・向上させるための統一的な枠組み
移行期限	• 原則として、**2035年を目処**に移行。ただし、**情報の重要性や暗号技術の利用状況等を把握**した上で、どのように移行を進めるかを検討し、適切に判断 • **例えば、特に機微な情報や保護期間が非常に長期となることが想定される情報等を扱う場合等**においては、**より早期に移行**を行うことも含め、情報システムごとに適切に検討を行う
移行に向けた取組	• 今後策定する工程表（ロードマップ）において、政府機関等が移行に向けた計画を策定できるよう、**移行に向けた計画に盛り込むべき基本的事項や留意すべき事項**を示す • **政府機関等は、今後策定する工程表（ロードマップ）を踏まえ、移行に向けた計画を策定し、移行期限までにPQCへ移行**を行う

関係府省庁の連携の下、**2026年度中に、工程表（ロードマップ）を策定する予定**

中間とりまとめにおけるCRYPTREC関係部分（抜粋）

2.(3) 耐量子計算機暗号（PQC）の安全性等の評価・確認とその時期について

CRYPTRECにおいて、CRYPTREC暗号リストの更新が可能となるよう、耐量子計算機暗号（PQC）の安全性評価・実装性能評価に関する活動を開始している。具体的には、2024年８月にNIST標準として公開されたFIPS 203(ML-KEM)、FIPS 204(ML-DSA)、FIPS 205(SLH-DSA)を対象として、順次、安全性評価・実装性能評価を実施中である。

3.(1)イ 耐量子計算機暗号（PQC）の安全性等の評価・確認とその時期について

政府機関等は、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群において、電子政府推奨暗号リストに基づき、情報システムで使用する暗号等を定めることとされているところ、電子政府推奨暗号リストに掲載された暗号技術については、安全性維持が困難と判断された場合、CRYPTRECにおいて、運用監視暗号リストに当該暗号技術を移行することとなっている。また、互換性維持の継続利用として使うにしても安全性維持が極めて困難で、互換性維持の継続利用が容認できないとCRYPTRECが判断した場合等には、運用監視暗号リストから当該暗号技術を削除するといった運用が行われている。

上記の運用における暗号技術の利用に係る停止の時期については、量子計算機技術の進展状況を踏まえた暗号技術の安全性評価、政府機関等における耐量子計算機暗号（PQC）への移行等の状況、諸外国の状況等を十分に踏まえながら、CRYPTREC暗号リストの取扱い等について、具体的方策の検討を進める。

5．その他

（略）移行においては、CRYPTRECが公表しているガイドライン*等も参考に、情報システムによって、必要とする保護期間や移行作業量が異なることを踏まえ、その優先度等に応じて対応を行ったり、移行対象の詳細な把握のためにクリプト・インベントリを構築したりするなど、移行の必要性や方法等について検討を進める必要がある。

(*) CRYPTREC暗号技術ガイドライン（耐量子計算機暗号）2024年度版（2025年3月CRYPTREC暗号技術調査ワーキンググループ（耐量子計算機暗号））。

工程表（ロードマップ）骨子　3.(2) 耐量子計算機暗号（PQC）の安全性確認

移行すべき耐量子計算機暗号（PQC）の安全性評価等が行われ、**安全性及び実装性能が確認された耐量子計算機暗号（PQC）について、CRYPTREC暗号リストに反映されるよう、その掲載方法も含め、CRYPTRECにおいて必要な検討を行う**こととする。

また、**政府機関等については、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群において、電子政府推奨暗号リストに基づき、情報システムで使用する暗号等を定めることが現状規定されているため、上記の状況も踏まえて必要な検討を行う**こととする。

資料６－１

耐量子計算機暗号（PQC）に対応した CRYPTREC暗号リストの在り方について

検討ポイント①　文書として別にするか

➢ PQC対応のリストについて、**現行の文書（CRYPTREC暗号リスト）の中にリストを追加する方法【案１】**と、**新たに別の文書を作成する方法【案２】**とのどちらにすべきか。
※「電子政府における調達のために参照すべき暗号のリスト（CRYPTREC暗号リスト）」（CRYPTREC LS-0001-2022R1）

【案１】 現行の文書（CRYPTREC暗号リスト）の中にPQC対応のリストを追加する。

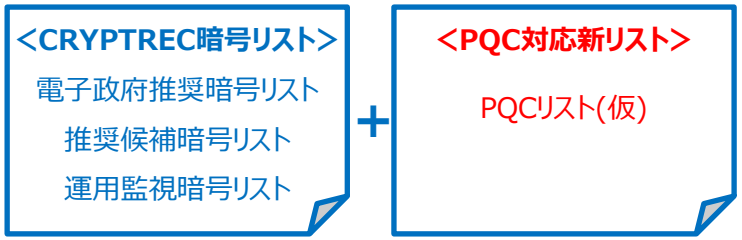
✓ （特に懸念点等はない。）

＜案１のイメージ＞

【案２】 現行の文書（CRYPTREC暗号リスト）とは別に新たに別の文書を作成する。

✓ 参照すべき文書が２つになることは、現状の運用から大きく変わることになり、利用する政府機関側にとってわかりにくく混乱を招きかねない。

＜案２のイメージ＞



→ **案１、現行の文書の中にPQC対応のリストを追加してはどうか。**
※政府統一基準を担当する国家サイバー統括室(NCO)においても案１とすることに特段の異論がないとの回答。

検討ポイント②　リストとして別にするか

➢ PQC対応のリストについて、**「電子政府推奨暗号リスト」の中にPQC対応の表を追加する方法【案１】**と、**既存の３リストとは別に、新たにリストを追加する方法【案２】**とのどちらにすべきか。

【案１】「電子政府推奨暗号リスト」の中にPQC対応の表を追加する。

✓「電子政府推奨暗号リスト」の名称は変わらず、政府統一基準群等の関連文書の変更が最低限となる。

✓ CRYPTREC暗号リスト内の「電子政府推奨暗号リスト」の定義文について変更検討が必要。
※「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」という部分のPQCへの適用について、PQCは今後の普及が見込まれると判断できるか検討し、合致しない場合は定義の文面の変更について検討が必要。

✓「電子政府推奨暗号リスト」に、表１・表２などの細区分が必要で、名称や説明文の有無の検討が必要。

✓ 長期的にPQC移行完了後も、３リスト構成自体は変更されないため長期的にわかりやすい。
※移行が進展した場合、PQC対応していない暗号技術は、電子政府推奨暗号リストから運用監視暗号リストに移行されると想定。

＜案１のイメージ＞

＜CRYPTREC暗号リスト＞
①電子政府推奨暗号リスト
　表１　現行暗号リスト(仮)
　表２　PQCリスト(仮)
②推奨候補暗号リスト
③運用監視暗号リスト

【案２】 既存の３リストとは別に、新たにリストを追加する。

✓ リスト構成が変わるため、政府統一基準群等の各種文書で利用する用語の全面見直しが必要。

✓「電子政府推奨暗号リスト」の定義文について再検討する必要がない。

✓ １リストで１つの表となるため、文書構成はわかりやすいが、PQCのリストが「電子政府推奨暗号リスト」と同程度(以上)に推奨されることを明確化する必要。

✓ 長期的にPQC移行完了後も、改めて全面的なリスト構成の見直しが必要。

＜案２のイメージ＞

＜CRYPTREC暗号リスト＞
①電子政府推奨暗号リスト
②推奨候補暗号リスト
③運用監視暗号リスト
④PQCリスト(仮)

→ 政府関連文書への影響を鑑み**案１、「電子政府推奨暗号リスト」の中にPQC対応の表を追加してはどうか。**

※NCOにおいても次の理由で案１とすることに特段の異論がないとの回答。

○政府統一基準においては、「暗号技術検討会及び関連委員会（CRYPTREC）により安全性及び実装性能が確認された「電子政府推奨暗号リスト」に基づき」としており、利用実績や普及見込には直接言及していないため、案1のように「電子政府推奨暗号リスト」の中にPQC対応の表を追加する方向性であれば、政府統一基準の改定をすることなく対応することが可能と考えられる（ガイドラインの改定については、別途、検討が必要）。

○一方で、案2のように別途リストを作成し、当該リストを「電子政府推奨暗号リスト」と同レベルの位置付けにしようとするのであれば、サイバーセキュリティ戦略本部決定を経た上で政府統一基準の改定が必要となる可能性がある。また、リスト構成の見直し等の都度、同様に政府統一基準の改定が必要となる可能性がある。

検討ポイント③　量子計算機耐性を持つ共通鍵暗号等の取扱い

➢ PQC対応のリストについて、**公開鍵暗号方式の狭義のPQCだけをリスト(表)とする方法【案１】**と、**量子計算機耐性を持つ共通鍵暗号等についてもリスト(表)に追加する方法【案２】**のどちらにすべきか。

【案１】 公開鍵暗号方式の狭義のPQCだけをリスト(表)とする。

✓ 共通鍵暗号等の量子計算機耐性については、
CRYPTREC暗号リストの脚注か暗号強度要件※に規定することで十分。
※「暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準」（CRYPTREC LS-0003-2022r1）

＜案１のイメージ＞

技術分類		暗号技術
公開鍵暗号	署名	ML-DSA
		SLH-DSA
	鍵共有	ML-KEM

【案２】 量子計算機耐性を持つ共通鍵暗号等についてもリスト(表)に追加する。

✓ リスト(表)を確認すれば量子計算機耐性を持つ暗号が確認でき、利用者の利便性が向上する。

✓ 従来は、一つの暗号技術について、３つリストに重複して掲載されることはなかったが、
AES等については、検討ポイント②で細分した表１（現行暗号の表）と、
表２（PQC対応の表）に重複して掲載することになるため、
これが利用者の混乱を招くおそれはないか。
→CRYPTREC暗号リストにおいて混乱を招かないように十分に配慮。

＜案２のイメージ＞

技術分類		暗号技術
公開鍵暗号	署名	ML-DSA
		SLH-DSA
	鍵共有	ML-KEM
共通鍵暗号		AES
ハッシュ関数		SHA-2
		SHA-3
（略）		（略）

→ PQC移行（＝量子計算機対応）は、これまでの移行に比べても、何が“耐量子”で何が“非耐量子”なのかが直感的に分かりにくいことに加え、既存システムの大規模な改修が必要となるなど、より多くの関係者に理解してもらう必要があることから、**案２、量子計算機耐性を持つ共通鍵暗号等についてもリスト(表)に追加してはどうか。**
※NCOにおいても案２とすることに特段の異論がないとの回答。

検討ポイント④　リストの名称

➢ 検討ポイント②・③で新リストの概要が定まったところ、「電子政府推奨暗号リスト」の中に新たに作成される**表１（現行暗号の表）と、表２（PQC対応の表）**についてどのような**名称**とすべきか。

【考え方】

✓ 表１については、移行に際して現在主に用いられている暗号である趣旨がわかりやすいよう「現行暗号リスト」としてはどうか。
ただし、今後、PQC移行の進展に伴い、必要に応じて名称を見直すことも視野にいれるべきではないか。

✓ 表２について、政府部内において「耐量子計算機暗号（PQC）」という用語が定着していることも踏まえ、「耐量子計算機暗号（PQC）リスト」としてはどうか。
※2025年６月から検討が開始された政府の会合も「政府機関等における耐量子計算機暗号（PQC）利用に関する関係府省庁連絡会議」であり、サイバーセキュリティ戦略（閣議決定文書）においても「耐量子計算機暗号（PQC）への移行」などとされている。

✓ 検討ポイント③を踏まえると、表２は量子計算機への耐性を有する共通鍵暗号等も含む。
このため、「耐量子計算機暗号（PQC）」という用語について、改めて検討すると次のとおりであり、今後、必要に応じてどちらの意味で用いているか誤解のないようにしていく必要がある。

- 狭義には、量子計算機でも解読困難な数学的課題に基づいた公開鍵暗号方式を指す。
- 広義には、量子計算機への耐性を有する暗号技術を指す。
  例えば「PQC移行」という場合には共通鍵暗号の移行（例：AES-128→AES-256）も含んで議論されることが多い。

→ **表１**については、「**現行暗号リスト**」とし、今後、必要があれば名称を見直すこととしてはどうか。
表２については、「**耐量子計算機暗号（PQC）リスト**」としてはどうか。

検討ポイント⑤　PQCのパラメータセットの取扱い

➢ PQCでは、通常、パラメーターセットが定義されており、**各パラメーターセットもリストに掲載すべきか否か。**

【背景】 PQCでは複数のパラメーターセットが定義。

✓ 演算内容や安全性証明の枠組みは共通しているが、格子の次元やノイズ分布などの安全性を調整するパラメータが異なる。

✓ 例えばML-KEMではFIPS 203で次の３種類が定義される。
（パラメータセットとともにセキュリティのカテゴリもFIPSに明記。）
- ML-KEM-512 (Category 1)
- ML-KEM-768 (Category 3)
- ML-KEM-1024 (Category 5)

＜参考：カテゴリのレベル＞
レベル１　128ビット鍵を持つブロック暗号に対する鍵探索（例: **AES-128**）
レベル２　256ビットのハッシュ関数に対する衝突探索（例: **SHA-256** / SHA3-256）
レベル３　192ビット鍵を持つブロック暗号に対する鍵探索（例: **AES-192**）
レベル４　384ビットのハッシュ関数に対する衝突探索（例: **SHA-384** / SHA3-384）
レベル５　256ビット鍵を持つブロック暗号に対する鍵探索（例: **AES-256**）

【考え方】

✓ システム調達時に暗号技術を選択する上でパラメータセットは重要な選択肢であり、CRYPTREC暗号リストで明示することが適当ではないか。
※現行のCRYPTREC暗号リストでも、例えばSHA-384とSHA-512では、演算内容が同じで出力長だけが異なるが、別の暗号技術として掲載される例がある。

＜パラメータセットの記載イメージ＞

暗号技術	
名称	**パラメーターセット**
ML-KEM	ML-KEM-512 (Category 1) ML-KEM-768 (Category 3) ML-KEM-1024 (Category 5)

✓ 「カテゴリ」についても、暗号技術を選択する上で重要な要素であり、カテゴリが定義されているものは併記することが適当ではないか。
また、表２におけるAESやSHAについてもカテゴリを併記する（表１については従来どおりの表記とする）ことが適当ではないか。
※現行暗号は「Xビットセキュリティ」で暗号強度要件の議論をしていたが、PQCでは、「カテゴリX」として暗号強度要件の議論を行うことが想定される。

✓ ハッシュ関数について、SHA2とSHA3を名称としてグループ化（各暗号技術はパラメーターセットの扱い）とすることでよいか。

✓ PQC対応としている製品においてもパラメーターセットまで明示しているものは少なく、明示について周知啓発も必要ではないか。

→システム調達時に暗号技術を選択する上で重要な選択肢となることから、
各パラメーターセットについて、カテゴリとともにリストに掲載してはどうか。
※「CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）」の2024年度版では「安全性レベル」と表記していたが、今後は「カテゴリ（Category）」とする。
これは、定義が置かれた米国NISTの文書では「Security Strength Categories」とされているものを指すことをわかりやすく（紛れがないように）するものである。

検討ポイント⑥　引き続き議論が必要な課題

➢ CRYPTREC暗号リストのPQC対応に当たり、次の課題が未整理であり引き続き検討する必要があるのではないか。
課題① **Category 1・2**の暗号の取扱い　　課題② **暗号利用モードや認証暗号等**の取扱い
課題③ **ハイブリッド構成**の取扱い　　課題④ **PQCの安全性評価等の進め方**

課題① Category 1・2（128ビットセキュリティ程度相当）の暗号の取扱い

✓ 暗号強度要件※では、128ビットセキュリティは2040年までは「利用可」だが、2041年以降は「移行完遂期間」とされ、2051年以降は順次「利用不可」となる。
※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（CRYPTREC LS-0003-2022R1）

✓ 政府システムのPQC移行は原則2035年が期限とされており、この時期を念頭においた際、128ビットセキュリティの暗号技術を「推奨」することとしてよいか検討が必要。

✓ 海外機関では、Category 3以上を想定する記載が多く見られるが、Category 1・2を排除するような記載はなく、相互運用性・国際調達の観点からも検討が必要。

＜参考：関係する暗号技術の例＞

暗号技術	
名称	パラメーターセット
ML-KEM	**ML-KEM-512 (Category 1)** ML-KEM-768 (Category 3) ML-KEM-1024 (Category 5)
AES	**AES-128 (Category 1)** AES-192 (Category 3) AES-256 (Category 5)
SHA2	**SHA-256 (Category 2)** **SHA-512/256 (Category 2)** SHA-384 (Category 4) SHA-512 (Category 5)

課題② 暗号利用モードや認証暗号等の取扱い

※CRQC: Cryptographically Relevant Quantum Computer

✓ 現行暗号のCRQCへの耐性について、AES、SHA2、SHA3については、CRYPTRECの外部評価報告書※等から明らか。
※量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価 2024年度版（CRYPTREC-EX-3401-2024）

✓ 一方で、暗号利用モード、メッセージ認証コード、認証暗号、エンティティ認証やCamellia、KCipher-2については検討が必要。

課題③ ハイブリッド構成の取扱い

※従来は「ハイブリッドモード」と称していたが暗号技術評価委員会での取扱いを踏まえ今後は「ハイブリッド構成」と称する。

✓ 現行暗号とPQCを組み合わせたハイブリッド構成について、CRYPTRECとしてどのように取り扱うか検討が必要。

課題④ 公開鍵暗号方式のPQCの安全性評価等の進め方

✓ 今後策定予定のFIPS標準等を始めとするPQCについて、どのような順序で安全性評価等を実施すべきか検討が必要。

→ これら課題の取扱いについては安全性評価の観点や利活用・普及促進の観点から横断した検討が必要であり、暗号技術評価委員会及び暗号技術活用委員会の協力も得ながら、暗号技術検討会の直下に、新たに**「耐量子計算機暗号（PQC）リスト検討タスクフォース（仮称）」**を立ち上げ、検討を行ってはどうか。

検討ポイント⑦　リストの説明文等

➢ これまでの検討ポイントを踏まえ、「電子政府推奨暗号リスト」の説明文を修正する必要があるか。
また、表１と表２に説明文を追加する必要はあるか。

【考え方】

- ✓ 政府機関等において原則2035年までにPQC移行が行われることとなり、PQCの普及が見込まれることから、当面の間、PQCは「今後の普及が見込まれる」と判断することでよいか。
- ✓ 「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、現在はPQCに対する規定がなく、2026年度に見直し予定。このため、規定の適用関係が明確になるように脚注を追加してはどうか。
- ✓ 表１について、従来のリストと同じであり特段の補足説明事項は不要ではないか。
- ✓ 表２について、新たに作成したものであり、表１との違いを説明する必要があるのではないか。
（広義の意味でPQCを用いていることに留意。）
- ✓ 表２について、今後、FIPS標準の評価を順次実施するほか、暗号利用モードや認証暗号等の取扱いについても今後議論を行うなど、暗号技術を今後も追加する見込みであり、予見可能性を高めるためにその旨の説明を追記。

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。

凡例
赤下線：現行からの改正部分
赤文字：本頁における検討部分

<表１　現行暗号リスト>
【表略】

<表２　耐量子計算機暗号（PQC）リスト>
現行暗号の解読に利用可能な水準の量子計算機（CRQC: Cryptographically Relevant Quantum Computer）への耐性を有することが確認された暗号技術のリスト[4]。
【表略】

1･2 （略）

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
なお、当該設定基準の見直しの検討を行う予定であり、当面の間、表２（耐量子計算機暗号（PQC）リスト）の公開鍵暗号は、当該設定基準を適用しない。

4 暗号技術の耐量子計算機暗号（PQC）リストへの追加について検討中である。
https://www.cryptrec.go.jp/[前ページの資料を整形して掲載]

→上に示すように、「電子政府推奨暗号リスト」の**説明文や注釈について、必要な修正**を行うこととしてはどうか。

検討ポイント⑦’ リストの説明文等

➢ 今回の更新と合わせ、CRYPTREC暗号リストの記載を適正化するため、次の修正を合せて行うこととする。

- ✓「安全性」について「safety」と「security」のどちらであるか明確とするため、追記を行うこととする。
- ✓ RSA暗号の強度について、注１として「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RSA 1024に係る移行指針」の記載があるが、現在は強度について「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」に規定されており、当該注釈は冗長であるため削る。
  ※運用監視暗号リストの注８も同様に削る。
- ✓ 注番号及び脚注番号について、番号がこれまで追記してきた順となっており複雑化していたことから、文書冒頭からの出現順となるよう番号を振替え。
- ✓ このほか、暗号技術評価委員会において提案のあった注釈を追加。
  ※ECDHに「使用するMACはHMAC又はCMACに限る。」との注を付すもの。

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（セキュリティ）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」[53]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。

<表１　現行暗号リスト>

技術分類		暗号技術
公開鍵暗号	署名	DSA（注18）
		ECDSA
		EdDSA
		RSA-PSS~~（注１）~~
		RSASSA-PKCS1-v1_5~~（注１）~~
	守秘	RSA-OAEP~~（注１）~~
	鍵共有	（略）
（略）	（略）	（略）

<表２　耐量子計算機暗号（PQC）リスト>

【略】

（略）

~~（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）~~

（略）

検討ポイント⑧　CRYPTREC暗号リスト・移行ルール

- 「電子政府推奨暗号リスト」を表１・表２としたこと、及びPQC（ML-KEM等）については、移行のためその普及を積極的に図っていく必要があること等から、**移行ルールについて次のように赤字部分を追加してはどうか**。
- 今後、PQC移行の進捗状況やPQC対応製品の普及状況を踏まえ必要に応じ、移行ルールは適宜見直しを行う。

＜表２への追加に関するルール＞
標準化等により将来的な利用が見込まれ、安全性や実装性能が十分にあると暗号技術検討会が決定した場合
（公募や事務局提案等）

①電子政府推奨暗号リスト

表１ 現行暗号リスト
表２ 耐量子計算機暗号(PQC)リスト

安全性及び実装性能が確認された暗号技術で、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、利用を推奨するもののリスト

＜表２新設に関する臨時ルール＞
表１の暗号技術のうち、CRQCへの耐性があると暗号技術検討会が決定したものを表２に追加する

次の条件のいずれかを満たすと暗号技術検討会が決定した場合
- ✓ ５年ごとの利用実績調査により、複数の利用実績を確認した場合
- ✓ その他、普及していることが明らか又は急速な普及が大いに見込まれる場合

安全性維持が困難（危殆化した）と暗号技術検討会が決定した場合

※ 電子政府推奨暗号リストに掲載された暗号技術は、利用者がいる前提であり、原則として、危殆化以外の理由では遷移させず、また、移行のための時間を確保する必要があるため、いきなりリストから削除することはしない。

標準化等により将来的な利用が見込まれ、安全性や実装性能が十分にあると暗号技術検討会が決定した場合
（公募や事務局提案等）

②推奨候補暗号リスト

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暗号リストへの掲載から20年を超えた後に実施する最初の利用実績調査までに、十分な利用実績を確認できなかったもの
- 公募提案暗号について、提案会社より自主取下げ要望があり、暗号技術検討会における審議の結果「今後の普及が見込まれない公募提案暗号」であると判断されたもの

③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確認されたが、互換性維持のために継続利用を容認する暗号技術のリスト

次の条件のいずれかを満たすと暗号技術検討会が決定した場合、削除猶予期間を定めて周知を行った上で、その期間の満了後に自動的に削除する。
- ✓ 運用監視暗号リストに掲載している注釈で示した互換性維持のための利用形態が必要なくなり、削除が妥当と判断した場合
- ✓ 互換性維持の継続利用として使うにしても安全性維持が極めて困難で、互換性維持の継続利用が容認できないと判断した場合
- ✓ その他、運用監視暗号リストに掲載している必要性の根拠を満たさなくなったと判断した場合

安全性維持が困難（危殆化した）と判断した場合

リストから削除

※ 利用実績調査の具体的な実施内容・評価基準は、暗号技術活用委員会において検討し、暗号技術検討会の承認を経た上で実施する。

※ ①・②への追加・遷移における安全性については、CRQCへの耐性を見据えて評価を行う。

（参考）カテゴリ関連記載：カテゴリの定義（米国NIST）

NIST (2016.12) “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

4.A.5 Security Strength Categories（抜粋）
NIST will base its classification on the range of security strengths offered by the existing NIST standards in symmetric cryptography, which NIST expects to offer significant resistance to quantum cryptanalysis. In particular, NIST will define a separate category for each of the following security requirements (listed in order of increasing strength):

1) Any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for key search on a block cipher with a 128-bit key (e.g. AES128)
2) Any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for collision search on a 256-bit hash function (e.g. SHA256/ SHA3-256)
3) Any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for key search on a block cipher with a 192-bit key (e.g. AES192)
4) Any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for collision search on a 384-bit hash function (e.g. SHA384/ SHA3-384)
5) Any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for key search on a block cipher with a 256-bit key (e.g. AES 256)

（参考）カテゴリ関連記載：FIPS 203･204･205

FIPS203 “Module-Lattice-Based Key-Encapsulation Mechanism Standard”
https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.203.pdf

8. Parameter Sets（抜粋）
Concretely, ML-KEM-512 is claimed to be in security category 1, ML-KEM-768 is claimed to be in security category 3, and ML-KEM-1024 is claimed to be in security category 5.

Table 2. Approved parameter sets for ML-KEM

	n	q	k	η_1	η_2	d_u	d_v	required RBG strength (bits)
ML-KEM-512	256	3329	2	3	2	10	4	128
ML-KEM-768	256	3329	3	2	2	10	4	192
ML-KEM-1024	256	3329	4	2	2	11	5	256

FIPS204 “Module-Lattice-Based Digital Signature Standard”
https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.204.pdf

4. Parameter Sets（抜粋）
Concretely, the parameter set ML-DSA-44 is claimed to be in security strength category 2, ML-DSA-65 is claimed to be in category 3, and ML-DSA-87 is claimed to be in category 5.

Table 1. ML-DSA parameter sets

Parameters (see Sections 6.1 and 6.2 of this document)	Values assigned by each parameter set ML-DSA-44	ML-DSA-65	ML-DSA-87
q - modulus [see §6.1]	8380417	8380417	8380417
ζ - a 512th root of unity in $\mathbb{Z}_q$ [see §7.5]	1753	1753	1753
d - # of dropped bits from $\mathbf{t}$ [see §6.1]	13	13	13
τ - # of ± 1's in polynomial c [see §6.2]	39	49	60
λ - collision strength of $\tilde{c}$ [see §6.2]	128	192	256
γ_1 - coefficient range of $\mathbf{y}$ [see §6.2]	2^{17}	2^{19}	2^{19}
γ_2 - low-order rounding range [see §6.2]	$(q-1)/88$	$(q-1)/32$	$(q-1)/32$
(k, ℓ) - dimensions of $\mathbf{A}$ [see §6.1]	(4,4)	(6,5)	(8,7)
η - private key range [see §6.1]	2	4	2
$\beta = \tau \cdot \eta$ [see §6.2]	78	196	120
ω - max # of 1's in the hint $\mathbf{h}$ [see §6.2]	80	55	75
Challenge entropy $\log_2 \binom{256}{\tau} + \tau$ [see §6.2]	192	225	257
Repetitions (see explanation below)	4.25	5.1	3.85
Claimed security strength	Category 2	Category 3	Category 5

FIPS205 “Stateless Hash-Based Digital Signature Standard”
https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.205.pdf

11. Parameter Sets（抜粋）
Concretely, the parameter sets with $n = 16$ are claimed to be in security category 1, the parameter sets with $n = 24$ are claimed to be in security category 3, and the parameter sets with $n = 32$ are claimed to be in security category 5.

Table 2. SLH-DSA parameter sets

	n	h	d	h'	a	k	lg_w	m	security category	pk bytes	sig bytes
SLH-DSA-SHA2-128s SLH-DSA-SHAKE-128s	16	63	7	9	12	14	4	30	1	32	7 856
SLH-DSA-SHA2-128f SLH-DSA-SHAKE-128f	16	66	22	3	6	33	4	34	1	32	17 088
SLH-DSA-SHA2-192s SLH-DSA-SHAKE-192s	24	63	7	9	14	17	4	39	3	48	16 224
SLH-DSA-SHA2-192f SLH-DSA-SHAKE-192f	24	66	22	3	8	33	4	42	3	48	35 664
SLH-DSA-SHA2-256s SLH-DSA-SHAKE-256s	32	64	8	8	14	22	4	47	5	64	29 792
SLH-DSA-SHA2-256f SLH-DSA-SHAKE-256f	32	68	17	4	9	35	4	49	5	64	49 856

（参考）カテゴリ関連記載：CRYPTREC PQCガイドライン

「CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）2024年度版」CRYPTREC GL-2007-2024

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

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

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

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

（参考）カテゴリ関連記載：海外政府機関による推奨状況①

暗号技術		ドイツ (BSI)	フランス (ANSSI)	オランダ (AIVD)	オーストラリア (ACSC)
名称	パラメーターセット				
ML-KEM	ML-KEM-512 (Category 1)				
	ML-KEM-768 (Category 3)	Recommended	préférence	Acceptable	approved*
	ML-KEM-1024 (Category 5)	Recommended	préférence	Recommended	approved
ML-DSA	ML-DSA-44 (Category 2)				
	ML-DSA-65 (Category 3)	Recommended	préférence	Acceptable	approved*
	ML-DSA-87 (Category 5)	Recommended	préférence	Recommended	approved
SLH-DSA	SLH-DSA-[SHA2/SHAKE]-128[s/f] (Category 1)		SLH-DSAの記載なし	Acceptable	強度の記載なし
	SLH-DSA-[SHA2/SHAKE]-192[s/f] (Category 3)	Recommended	SLH-DSAの記載なし	Recommended	強度の記載なし
	SLH-DSA-[SHA2/SHAKE]-256[s/f] (Category 5)	Recommended	SLH-DSAの記載なし	Recommended	強度の記載なし
AES	AES-128 (Category 1)	Recommended		Acceptable	approved*
	AES-192 (Category 3)	Recommended			approved*
	AES-256 (Category 5)	Recommended	encourage	Recommended	approved
SHA2	SHA-256 (Category 2)	Recommended		Recommended	approved*
	SHA-512/256 (Category 2)	Recommended			
	SHA-384 (Category 4)	Recommended	encourage	Recommended	approved
	SHA-512 (Category 5)	Recommended	encourage	Recommended	approved
SHA3	SHA3-256 (Category 2)	Recommended		Recommended	PQC内部使用のみ可
	SHAKE128 (Category 2)			Acceptable	PQC内部使用のみ可
	SHA3-384 (Category 4)	Recommended	encourage	Recommended	PQC内部使用のみ可
	SHA3-512 (Category 5)	Recommended	encourage	Recommended	PQC内部使用のみ可
	SHAKE256 (Category 5)		encourage	Recommended	PQC内部使用のみ可

ドイツ（BSI）：“Technical Guideline TR-02102-2 Cryptographic Mechanisms: Recommendations and Key Lengths”
フランス（ANSSI）：“Avis de l’ANSSI sur la migration vers la cryptographie post-quantique“（PQC移行に関するANSSIの見解）
オランダ（AIVD）：“The PQC Migration Handbook”
オーストラリア（ACSC）：“Guidelines for cryptography”

＊ 2030年以降は非推奨

（参考）カテゴリ関連記載：海外政府機関による推奨状況②

ドイツ　情報セキュリティ庁（BSI）
“Technical Guideline TR-02102-2 Cryptographic Mechanisms: Recommendations and Key Lengths” (2025.1)
https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr02102/tr02102_node.html

2.4.3. ML-KEM Key Agreement（抜粋）　※[103]はFIPS203
ML-KEM with the parameter sets corresponding to NIST Security Strength Categories 3 and 5 (see Table 2.7) from [103] is considered cryptographically suitable for the long-term protection of confidential information at
the security level targeted in this Technical Guideline.

- ML-KEM-768, see [103],
- ML-KEM-1024, see [103].

Table 2.7: Recommended parameters for ML-KEM.

オーストラリア　サイバーセキュリティセンター（ACSC）
“Guidelines for cryptography” (2025.12時点)
https://www.cyber.gov.au/business-government/asds-cyber-security-frameworks/ism/cyber-security-guidelines/guidelines-for-cryptography

Using the Module-Lattice-Based Key Encapsulation Mechanism（抜粋）
The use of ML-KEM-768 and ML-KEM-1024 are approved. However, for interoperability and maintainability reasons, ML-KEM-768 will not be approved beyond 2030.

イギリス　国家サイバーセキュリティセンター（NCSC）
“Next steps in preparing for post-quantum cryptography” (2024.8)
https://www.ncsc.gov.uk/whitepaper/next-steps-preparing-for-post-quantum-cryptography

Key takeaways from this guidance（抜粋）
ML-KEM (Kyber) and ML-DSA (Dilithium) are algorithms standardised by NIST that are suitable for general purpose use. All proposed parameter sets provide an acceptable level of security for personal, enterprise and OFFICIAL-tier government information. The NCSC recommends ML-KEM-768 and ML-DSA-65 as providing appropriate levels of security and efficiency for most use cases.

フランス　国家情報システムセキュリティ庁（ANSSI）【仏語なので仮訳】
“Avis de l'ANSSI sur la migration vers la cryptographie post-quantique“（PQC移行に関するANSSIの見解）(2023.12)
https://messervices.cyber.gouv.fr/guides/avis-de-lanssi-sur-la-migration-vers-la-cryptographie-post-quantique-0

CRYSTALS-Kyber, aussi appelé ML-KEM（抜粋・仮訳）
このアルゴリズムが暗号製品への組み込みに選定された場合、ANSSIは以下の推奨事項を示します。
1. 標準化されたインスタンスのパラメータを変更しないことが重要です。
2. パラメータは複数のセキュリティレベルで拒否されます。可能な限り最高のNISTセキュリティレベル、できればレベル5（AES-256に相当）またはレベル3（AES-192に相当）を使用することをお勧めします。

オランダ　オランダ国家情報・安全保障局（AIVD）
“The PQC Migration Handbook” (2024.12)
https://english.aivd.nl/publications/publications/2024/12/3/the-pqc-migration-handbook

4.2) Recommended Cryptographic Primitives（抜粋）

Primitive	Recommended	Acceptable
ML-KEM[3]	ML-KEM-1024[1]	ML-KEM-768[1]
ML-DSA[3]	ML-DSA-87[2]	ML-DSA-65[2]
SLH-DSA	SLH-DSA-(SHA2/SHAKE)-256(s/f) SLH-DSA-(SHA2/SHAKE)-192(s/f)	SLH-DSA-(SHA2/SHAKE)-128(s/f)
AES	AES-256	AES-128
SHA-3	SHA-3-256 SHA-3-384 SHA-3-512 (c)SHAKE256	SHA-3-224 (c)SHAKE128
SHA-2	SHA-256 SHA-384 SHA-512	SHA-224

[1] Recommended to be deployed in a hybrid combination with ECDH. [2] Recommended to be deployed in a hybrid combination with either ECDSA or EdDSA. [3] BSI, ANSSI, NLNCSA recommend the use of NIST level 5 or 3 parameter sets. NSA CNSA 2.0 requires level 5.

Table 4.2 | Recommended parameter instantiations of the cryptographic primitives.

（参考）電子政府推奨暗号リスト～更新前

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（注18）
		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）
暗号利用モード	秘匿モード	CBC
		CFB
		CTR
		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

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/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。

（注18） FIPS PUB 186-5では廃止されているが、本リスト掲載時から安全性・利用実績の状況に大きな変化がないため、掲載を継続する。

CRYPTREC暗号リストの更新について

- これまでの検討ポイントを踏まえて、CRYPTREC暗号リストについて更新を行う。
- 「表２ 耐量子計算機暗号（PQC）リスト」の「公開鍵暗号」としては、
  暗号技術評価委員会において安全性評価等が行われた「ML-KEM」を追加する。

  ※ML-KEMの仕様書としては、次を用いる。
  NIST FIPS PUB 203
  https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf

→ **「電子政府推奨暗号リスト」の更新イメージは次ページのとおり。**
また、脚注４として参照する資料は次々ページのとおり。

電子政府推奨暗号リスト～更新イメージ

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（セキュリティ）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」[53]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。

<表１　現行暗号リスト>

技術分類		暗号技術
公開鍵暗号	署名	DSA（注~~18~~1）
		ECDSA
		EdDSA
		RSA-PSS ~~（注１）~~
		RSASSA-PKCS1-v1_5 ~~（注１）~~
	守秘	RSA-OAEP ~~（注１）~~
	鍵共有	DH
		ECDH（注２）
共通鍵暗号	64ビットブロック暗号（注~~2~~3）	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~~4）
		SHAKE256（注~~12~~4）
暗号利用モード	秘匿モード	CBC
		CFB
		CTR
		XTS（注~~17~~5）
	認証付き秘匿モード（注~~13~~6）	CCM
		GCM（注~~4~~7）
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

<表２　耐量子計算機暗号（PQC）リスト>

現行暗号の解読に利用可能な水準の量子計算機（CRQC: Cryptographically Relevant Quantum Computer）への耐性を有することが確認された暗号技術のリスト[4]。

技術分類		暗号技術	
		名称	パラメーターセット
公開鍵暗号	署名	–	–
	鍵共有	ML-KEM	ML-KEM-768 (Category 3) ML-KEM-1024 (Category 5)
共通鍵暗号		AES	AES-192 (Category 3) AES-256 (Category 5)
ハッシュ関数		SHA2	SHA-384 (Category 4) SHA-512 (Category 5)
		SHA3	SHA3-384 (Category 4) SHA3-512 (Category 5) SHAKE256（注８） (Category 5)

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。

~~53~~ 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
なお、当該設定基準の見直しの検討を行う予定であり、当面の間、表２（耐量子計算機暗号（PQC）リスト）の公開鍵暗号は、当該設定基準を適用しない。

4 暗号技術の耐量子計算機暗号（PQC）リストへの追加について検討中である。
https://www.cryptrec.go.jp/[継続検討内容に関する資料を掲載]

~~（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~~18~~1）　FIPS PUB 186-5では廃止されているが、本リスト掲載時から安全性・利用実績の状況に大きな変化がないため、掲載を継続する。

（注２）　使用するMACはHMAC又はCMACに限る。

（注~~2~~3）　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~~12~~4）　ハッシュ長は256ビット以上とすること。

（注~~17~~5）　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。

（注~~13~~6）　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~~4~~7）　初期化ベクトル長は96ビットを推奨する。

（注８）　ハッシュ長は256ビット以上とすること。

CRYPTREC暗号リストのPQC対応に関する検討課題

- 2025年度の暗号技術検討会において、CRYPTREC暗号リストの耐量子計算機暗号（PQC）対応に関する検討を行い、更なる検討が必要な課題について保留※した上でCRYPTREC暗号リストの改定を実施。
  ※現時点で、表２（耐量子計算機暗号（PQC）リスト）には、Category1・2の暗号技術や、暗号利用モードや認証暗号等の暗号技術は掲載していない。
- これらの課題については、安全性評価の観点のほか、利活用・普及促進の観点も含めた横断的な検討が必要。

→ 暗号技術評価委員会及び暗号技術活用委員会の協力も得ながら、暗号技術検討会の直下に、新たに**「耐量子計算機暗号（PQC）リスト検討タスクフォース」**を設け、2026年度末までを目途に検討を進める。

課題① Category 1・2（128ビットセキュリティ程度相当）の暗号の取扱い

- ✓ 暗号強度要件※では、128ビットセキュリティは2040年までは「利用可」だが、2041年以降は「移行完遂期間」とされ、2051年以降は順次「利用不可」となる。
  ※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（CRYPTREC LS-0003-2022R1）
- ✓ 政府システムのPQC移行は原則2035年が期限とされており、この時期を念頭においた際、128ビットセキュリティの暗号技術を「推奨」することとしてよいか検討が必要。
- ✓ 海外機関では、Category 3以上を想定する記載が多く見られるが、Category 1・2を排除するような記載はなく、相互運用性・国際調達の観点からも検討が必要。

＜参考：関係する暗号技術の例＞

暗号技術	
名称	パラメーターセット
ML-KEM	**ML-KEM-512 (Category 1)** ML-KEM-768 (Category 3) ML-KEM-1024 (Category 5)
AES	**AES-128 (Category 1)** AES-192 (Category 3) AES-256 (Category 5)
SHA2	**SHA-256 (Category 2)** **SHA-512/256 (Category 2)** SHA-384 (Category 4) SHA-512 (Category 5)

課題② 暗号利用モードや認証暗号等の取扱い

- ✓ 現行暗号のCRQCへの耐性について、AES、SHA2、SHA3については、CRYPTRECの外部評価報告書※等から明らか。
  ※量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価 2024年度版（CRYPTREC-EX-3401-2024）
- ✓ 一方で、暗号利用モード、メッセージ認証コード、認証暗号、エンティティ認証やCamellia、KCipher-2については検討が必要。

課題③ ハイブリッド構成の取扱い

- ✓ 現行暗号とPQCを組み合わせたハイブリッド構成について、CRYPTRECとしてどのように取り扱うか検討が必要。

課題④ 公開鍵暗号方式のPQCの安全性評価等の進め方

- ✓ 今後策定予定のFIPS標準等を始めとするPQCについて、どのような順序で安全性評価等を実施すべきか検討が必要。

※FIPS204、205は2026年度中に安全性評価等が終了予定。

資料６－２

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）

令和５年３月30日
デジタル庁・総務省・経済産業省
（最終更新：令和~~６年５月16日~~８年●月●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（セキュリティ）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~~[3]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。

＜表１　現行暗号リスト＞

技術分類		暗号技術
公開鍵暗号	署名	DSA（注~~1~~1）
		ECDSA
		EdDSA
		RSA-PSS~~（注１）~~
		RSASSA-PKCS1-v1_5~~（注１）~~
	守秘	RSA-OAEP~~（注１）~~
	鍵共有	DH
		ECDH（注２）
共通鍵暗号	64ビットブロック暗号（注~~2~~3）	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~~1~~2~~4~~）
		SHAKE256（注~~1~~2~~4~~）

（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。

~~3~~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
なお、当該設定基準の見直しの検討を行う予定であり、当面の間、表２（耐量子計算機暗号（PQC）リスト）の公開鍵暗号は、当該設定基準を適用しない。

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~~5）
	認証付き秘匿モード（注~~13~~6）	CCM
		GCM（注~~4~~7）
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~~18~~１）FIPS PUB 186-5では廃止されているが、本リスト掲載時から安全性・利用実績の状況に大きな変化がないため、掲載を継続する。

（注２）使用するMACはHMAC又はCMACに限る。

（注~~2~~３）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~~12~~４）ハッシュ長は256ビット以上とすること。

（注~~17~~５）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。

（注~~13~~６）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~~4~~７）初期化ベクトル長は96ビットを推奨する。

＜表２　耐量子計算機暗号（PQC）リスト＞

現行暗号の解読に利用可能な水準の量子計算機（CRQC: Cryptographically Relevant Quantum Computer）への耐性を有することが確認された暗号技術のリスト[4]。

技術分類		暗号技術	
		名称	パラメーターセット
公開鍵暗号	署名	ー	ー
	鍵共有	ML-KEM	ML-KEM-768（Category 3） ML-KEM-1024（Category 5）
共通鍵暗号		AES	AES-192（Category 3） AES-256（Category 5）
ハッシュ関数		SHA2	SHA-384（Category 4） SHA-512（Category 5）
		SHA3	SHA3-384（Category 4） SHA3-512（Category 5） SHAKE256（注８）（Category 5）

（注８）ハッシュ長は256ビット以上とすること。

4 暗号技術の耐量子計算機暗号（PQC）リストへの追加について検討中である。
https://www.cryptrec.go.jp/XXXXXXXX[継続検討内容に関する資料を掲載]

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~~5~~]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~~ 9）
共通鍵暗号	64ビットブロック暗号（注~~6~~10）	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~~11）
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~~12）	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~~ 9）KEM（Key Encapsulating Mechanism）－DEM（Data Encapsulating Mechanism）構成における利用を前提とする。

（注~~6~~10）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~~11）平文サイズは64ビットの倍数に限る。

（注~~14~~12）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~~5~~] 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[47]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[78]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」[89]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~~13）
	鍵共有	該当なし
共通鍵暗号	64ビットブロック暗号 （注~~15~~14）	3-key Triple DES （注~~19~~15）
	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~~17）
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~~13）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~~15~~14）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~~19~~15）SP 800-67 Revision 2では廃止されているが、本リスト掲載時から安全性・利用実績の状況に大きな変化がないため、掲載を継続する。

（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。

（注~~11~~17）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。

---

74 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。
~~87~~ 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
~~98~~ 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

更新履歴情報

更新日付	更新箇所	更新前の記述	更新後の記述
令和６年５月16日	~~（注18）~~注	［新規追加］	（注18）FIPS PUB 186-5では廃止されているが、本リスト掲載時から安全性・利用実績の状況に大きな変化がないため、掲載を継続する。
	~~（注19）~~注	［新規追加］	（注19）SP 800-67 Revision 2では廃止されているが、本リスト掲載時から安全性・利用実績の状況に大きな変化がないため、掲載を継続する。
令和８年●月●日	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（セキュリティ）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。
	電子政府推奨暗号リスト（表）	［表の件名付与］	＜表１　現行暗号リスト＞
	電子政府推奨暗号リスト（表）	［表の新設］	＜表２　耐量子計算機暗号（PQC）リスト＞ 現行暗号の解読に利用可能な水準の量子計算機（CRQC: Cryptographically Relevant Quantum Computer）への耐性を有することが確認された暗号技術のリスト。
	電子政府推奨暗号リスト（耐量子計算機暗号（PQC）リスト）	［技術分類の追加］	技術分類：公開鍵暗号－署名 名称／パラメーターセット：－ 技術分類：公開鍵暗号－鍵共有 名称：ML-KEM パラメーターセット： ML-KEM-768（Category 3） ML-KEM-1024（Category 5）
	電子政府推奨暗号リスト（耐量子計算機暗号（PQC）リスト）	［技術分類の追加］	技術分類：共通鍵暗号 名称：AES パラメーターセット： AES-192（Category 3） AES-256（Category 5）
	電子政府推奨暗号リスト（耐量子計算機暗号（PQC）リスト）	［技術分類の追加］	技術分類：ハッシュ関数 名称：SHA2 パラメーターセット： SHA-384（Category 4） SHA-512（Category 5） 名称：SHA3 パラメーターセット： SHA3-384（Category 4） SHA3-512（Category 5） SHAKE256（Category 5）
	脚注	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	３　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 なお、当該設定基準の見直しの検討を行う予定であり、当面の間、表２（耐量子計算機暗号（PQC）リスト）の公開鍵暗号は、当該設定基準を適用しない。
	脚注	［新規追加］	４　暗号技術の耐量子計算機暗号（PQC）リストへの追加について検討中である。 https://www.cryptrec.go.jp/XXXXXXXX［継続検討内容に関する資料を掲載］
	脚注	脚注３、４、７、８	脚注５、７、８、９

	注	（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）	［削除］
	注	（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）	［削除］
	注	［新規追加］	（注２）使用するMACはHMACまたはCMACに限る。
	注	［新規追加］	（注８）ハッシュ長は256ビット以上とすること。
	注	注18、２、12、17、13、４、５～７、14、９、15、19、11	注１、３、４、５、６、７、９～11、12、13、14、15、17

資料７

「耐量子計算機暗号（PQC）リスト検討タスクフォース」
開催要綱（案）

１．開催の趣旨・目的

量子計算機技術の進展に伴い、現在広く利用される公開鍵暗号の安全性が著しく低下すると想定され、耐量子計算機暗号（PQC）への移行は急を要する課題である。

CRYPTRECでは、CRYPTREC暗号リストのPQC対応を図るため、2025年度の暗号技術検討会において、電子政府推奨暗号リストに「耐量子計算機暗号（PQC）リスト」を加える改定を実施したところであるが、当該改定に関連して、更なる検討を要する課題が残されており、安全性評価の観点に加え、利活用及び普及促進の観点も踏まえた専門的かつ横断的な整理が必要となっている。

このため、CRYPTREC暗号リストのPQC対応に関する課題等の整理を行うことを目的として、暗号技術検討会の下に「耐量子計算機暗号（PQC）リスト検討タスクフォース」（以下「TF」という。）を開催する。

２．検討事項

(1) 128ビットセキュリティ相当の暗号技術の取扱いに関する検討
(2) 暗号利用モード、メッセージ認証コード、認証暗号、エンティティ認証等の取扱いに関する検討
(3) 現行暗号とPQCを組み合わせたハイブリッドモードの取扱いに関する検討
(4) 公開鍵暗号方式のPQCの安全性評価等の進め方に関する検討
(5) その他、耐量子計算機暗号（PQC）リストの運用に関し必要な事項

３．構成等

(1) TFの構成員は、別紙【別途検討】のとおりとする。
(2) TFの座長は、構成員の互選により定める。
(3) 座長は、TF構成員の中から座長代理を指名できる。
(4) 座長は、TFの議事を掌握する。
(5) 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。
(6) 座長が必要と認めた者は、オブザーバとしてTFに出席することができる。
(7) 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。
(8) その他、TFの運営に関し必要な事項は、座長が定めるところによる。

４．議事の公開

(1) TFは非公開とする。
(2) TFで使用した資料及びTFの議事概要については、次の場合を除き、公開する。
① 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と座長が認める場合
② その他、非公開とすることが必要と座長が認める場合

５．庶務

TFの庶務は、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の協力を得て、デジタル庁、総務省及び経済産業省において処理する。

資料８

2026 年度暗号技術評価委員会活動計画（案）

１．活動目的

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

２．活動概要

（１）暗号技術の安全性及び実装に係る監視及び評価

以下の通り、暗号技術の安全性及び実装に係る監視・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技術の削除に係る検討

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、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 CRYPTREC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。

（ア）米国NISTのPQC標準化において3件の標準化文書（FIPS 203から205）が公開され、今後も2件の標準化文書（FIPS 206及びFIPS 207）の公開が予定されるとともに、追加署名の選定プロジェクトが継続している。さらに、標準方式が世界各国で推奨暗号とされ、PQCへの移行が進められている。このような状況を鑑み、暗号技術調査ワーキンググループ（耐量子計算機暗号）において、PQCに関する技術動向を継続して調査・把握するとともに、ガイドライン及び調査報告書を作成する。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。

（イ）CRYPTREC暗号リスト掲載に向けたPQCの技術的検討に資するための外部評価を実施する。具体的には、FIPS 204及びFIPS 205の暗号技術（ML-DSA及びSLH-DSA）について、安全性・実装性能に関する調査及び評価を行う。

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

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

３．活動スケジュール

（１）暗号技術評価委員会は、下表のとおり２回の開催を予定する。

回	開催予定日	議案（予定）
第１回	2026 年 6 月上旬～6 月下旬	● 暗号技術評価委員会活動計画の具体的な進め方に関する審議 ● 暗号技術調査ワーキンググループ（耐量子計算機暗号）活動計画（案）に関する審議
第２回	2027 年 2 月中旬～3 月上旬	● 暗号技術評価委員会活動報告（案）に関する審議 ● 暗号技術調査ワーキンググループ（耐量子計算機暗号）活動報告（案）及びガイドライン・調査報告書に関する審議

（２）外部評価に関する審議のため、下表のとおり２回のメール審議を予定する。

回	予定日	議案（予定）
第１回	2026 年 9 月上旬～9 月下旬	● 外部評価（耐量子計算機暗号 ML-DSA の安全性・実装性能に関する評価及び調査）に関する審議
第２回	2026 年 11 月上旬～11 月下旬	● 外部評価（耐量子計算機暗号 SLH-DSA の安全性・実装性能に関する評価及び調査）に関する審議

以上

資料９

2026 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） 暗号強度要件ガイドラインの見直し

５年ごとの見直し時期にあたることに加え、PQC 移行方針が取りまとめられることを前提として、PQC の取り扱いを含めた形での見直しを実施する。2026 年度末での完成を目指す。

（2） TLS 暗号設定ガイドラインの見直し

PQC のサポートに向けた TLS1.3 への移行や PQC 関連技術の取込み、電子証明書の有効期限の短縮化、現行セキュリティ例外型の削除など、現在の TLS 暗号設定ガイドラインの前提となる条件が大きく変わりつつあるため、記載内容の見直しを実施する。委員任期を考慮しつつ、2027 年度末での完成を目指す。

（3） クラウドにおける鍵管理ガイダンスの作成

暗号鍵管理ガイドラインの拡充活動の一環として、クラウドサービスを利用したシステムにおける暗号鍵管理の適切な設計・構築・運用を目的に、クラウド鍵管理ガイダンスを作成する。クラウド鍵管理ガイダンス WG においてガイダンスの位置づけや内容を確定し、2026 年度末でのガイダンス完成を目指す。

（4） その他

その他、暗号技術の活用に係る状況の変化に応じ、暗号技術検討会で必要と位置づけられた活動の実施を検討する。

3. 活動スケジュール

暗号技術活用委員会の開催日程・議題については、以下のとおり、年 2 回の委員会開催を予定する。また、必要に応じて追加の委員会開催やメール審議を実施する。

回	開催日	議案（予定）
第1回	2026年6月～7月	■ 2026年度暗号技術活用委員会活動計画の確認 ■ クラウド鍵管理ガイダンスWG活動計画の審議 ■ 暗号強度要件ガイドライン見直しに関する検討 ■ クラウド鍵管理ガイダンスの進捗状況の確認
第2回	2027年2月下旬 ～3月上旬	■ 暗号強度要件ガイドライン見直しに関する審議 ■ TLS暗号設定ガイドライン見直しに関する検討 ■ クラウド鍵管理ガイダンスWG活動成果の審議 ■ 2026年度暗号技術活用委員会活動報告案について

以上

資料10

暗号技術検討会
２０２５年度　報告書
（案）

２０２６年３月

目次

１．はじめに

情報通信技術の急速な発展により、自動車、家電、医療、農業、工場等の様々な分野で、あらゆるモノがネットワークにつながる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、その影響が実空間にまで到達するリスクも高まっている。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り、IoT機器から得られる大量のデータの流通・連携を支える観点からも、その重要性は一層高まっている。

さらに近年は、量子計算機技術の進展に伴い、現在広く利用されている公開鍵暗号の安全性が低下することが懸念され、量子計算機への耐性を有する「耐量子計算機暗号（PQC）」への移行も急を要する課題となっている。米国では2024年８月に３方式の耐量子計算機暗号（PQC）が標準化され、実際にベンダー等での対応も広がってきているほか、我が国政府においても、原則として、2035年を目途に耐量子計算機暗号（PQC）へ移行する方針を昨年11月に発表するなど、耐量子計算機暗号（PQC）への移行に向けた関心が高まっている。

こうした中で、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CRYPTRECにおいても、耐量子計算機暗号（PQC）への対応が喫緊の課題であると認識している。そのため今年度は、暗号技術評価委員会において耐量子計算機暗号（PQC）の安全性・実装性能評価を行うとともに、暗号技術活用委員会において耐量子計算機暗号（PQC）に対応したCRYPTREC暗号リストの在り方について検討を行い、これらの結果を踏まえて暗号技術検討会においてCRYPTREC暗号リストを耐量子計算機暗号（PQC）に対応するための審議を行うなど、CRYPTRECのプロジェクト全体として耐量子計算機暗号（PQC）への対応に向けて精力的に活動を行った。

このほか、各委員会においては、暗号技術評価委員会では耐量子計算機暗号（PQC）に関する調査報告書とガイドラインの2026年度の作成に向けて調査を実施するとともに、「素因数分解の困難性に関する計算量評価」及び「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図を更新した。暗号技術活用委員会では、クラウドにおける鍵管理ガイダンスの検討を行った。これらの2025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 2025」を参照いただきたい。

今後も暗号技術を用いた情報システム及び情報社会全体のセキュリティ確保のために、成果物の検討や情報発信等を行っていく所存である。

末筆ではあるが、暗号技術検討会及び関係委員会等に御協力いただいた構成員、オブザーバの方々をはじめ、関係者の皆様に心から謝意を表する次第である。

2026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（セキュリティ）を暗号技術の専門家により技術的・専門的な見地から客観的に評価することが重要である。電子政府のセキュリティ確保のためには、安全性に優れた暗号技術を利用することが不可欠である。

このため、総務省及び経済産業省は、客観的な評価により安全性及び実装性に優れると判断される暗号技術をリスト化し、各府省に対してその利用を推奨することにより、高度な安全性と信頼性に支えられ、国民が安心して利用できる電子政府の構築に貢献することを目指し、2001年５月に最初の暗号技術検討会を開催した。

その後、2021年のデジタル庁発足に伴いデジタル庁が加わり、デジタル庁、総務省及び経済産業省は、電子政府推奨暗号等の安全性を評価・監視し、暗号技術の更なる普及促進を行うため、継続的に暗号技術検討会を開催している。

暗号技術検討会での検討を経て、2003年２月に策定された電子政府推奨暗号リストは、2013年３月にCRYPTREC暗号リストとして改定され、2023年３月に再改定（最終更新は2026年３月）された。

２．２．CRYPTRECの体制

CRYPTRECとは、Cryptography Research and Evaluation Committeesの略であり、デジタル庁、総務省及び経済産業省が共同で開催する暗号技術検討会（座長：松本勉国立研究開発法人産業技術総合研究所フェロー、横浜国立大学　先端科学高等研究院上席特別教授）と、国立研究開発法人情報通信研究機構（NICT）及び独立行政法人情報処理推進機構（IPA）が共同で開催する委員会から構成される暗号技術評価プロジェクトをいう。

委員会は、暗号技術評価委員会及び暗号技術活用委員会から構成され、それぞれ2025年度は「暗号技術調査WG（耐量子計算機暗号）」及び「クラウド鍵管理ガイダンスWG」を設置して検討を行った。

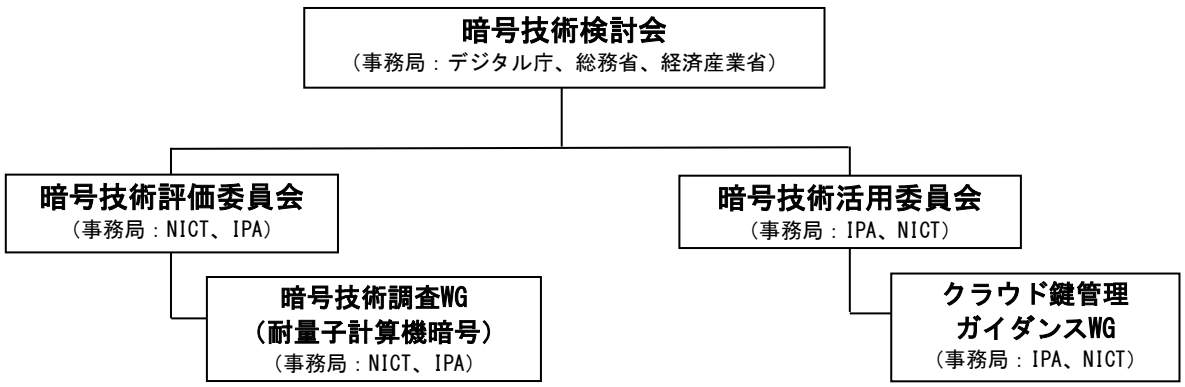


図2.2-1　CRYPTREC体制図（2025年度）

２．３．暗号技術検討会の開催実績

2025年度は、暗号技術検討会を１回開催し、各委員会の活動内容について報告・承認等を行うとともに、耐量子計算機暗号（PQC）に対応するためのCRYPTREC暗号リストの更新に関する審議等を行った。詳細は以下のとおり。

【第１回】2026年３月25日（水）9:00～11:00【P】

○ 暗号技術評価委員会の2025年度の活動についてNICTから報告が行われた。
○ CRYPTREC暗号リスト仕様書の参照先変更、並びに「耐量子計算機暗号ML-KEMの安全性・実装性能に関する評価及び調査」及び「耐量子計算機暗号の移行に関する技術動向調査」に関する外部評価報告書について、原案のとおり承認された。
○ 暗号技術活用委員会の2025年度の活動についてIPAから報告が行われた。
○ 政府機関等における耐量子計算機暗号（PQC）への移行について、国家サイバー統括室（NCO）から報告が行われた。
○ 耐量子計算機暗号（PQC）に対応したCRYPTREC暗号リストの在り方についてについて審議が行われ、CRYPTREC暗号リストについて原案のとおり更新することとされた。また、CRYPTREC暗号リストの耐量子計算機暗号（PQC）対応に関する課題等の整理を行うため、「耐量子計算機暗号（PQC）タスクフォース」を設置することについて審議され了承された。いずれも詳細は次節のとおり。
○ 暗号技術評価委員会及び暗号技術活用委員会における2026年度の活動計画案について、原案のとおり承認された。
○ 本報告書について、議論結果を反映することとした上で承認された。

２．４．CRYPTREC暗号リストの更新等

量子計算機技術の進展に伴い、現在広く利用されている公開鍵暗号の安全性低下（危殆化）や、解読可能となることを見越してデータを保存する攻撃（Harvest Now, Decrypt Later攻撃）が懸念されており、政府機関等における耐量子計算機暗号（PQC）への移行は急を要する課題である。

このため、内閣官房及びCRYPTREC事務局であるデジタル庁、総務省、経済産業省等の関係府省庁により、2025年６月に「政府機関等における耐量子計算機暗号（PQC）利用に関する関係府省庁連絡会議」が設置された。同年11月に中間取りまとめが行われ、政府機関等の情報システムについては、原則として、2035年を目途に耐量子計算機暗号（PQC）へ移行することとされた。

政府で利用可能な暗号は、CRYPTREC暗号リストに定められており[1]、CRYPTRECにおいても耐量子計算機暗号（PQC）への対応を速やかに進める必要があることから、暗号技術評価委員会では耐量子計算機暗号（PQC）の安全性・実装性能に関する評価を実施するとともに、暗号技術活用委員会では耐量子計算機暗号（PQC）に対応したCRYPTREC暗号リストの在り方について検討を行った。

CRYPTREC暗号リストの在り方については、論点が多岐にわたったことから、暗号技術活用委員会での検討を踏まえ、事務局、NICT、IPA等が連携して、論点やその考え方について整理した。当該整理については、暗号技術活用委員会及び暗号技術評価委員会での議論を経て、暗号技術検討会において審議された（暗号技術検討会の資料６－１）。審議結果を踏まえてCRYPTREC暗号リストの更新が行われ、CRYPTREC暗号リストの電子政府推奨暗号リストに「表２　耐量子計算機暗号（PQC）リスト」が新たに追加された。

なお、CRYPTREC暗号リストにおける耐量子計算機暗号（PQC）への対応に当たっては、引き続き検討すべき課題[2]もあり、当該課題の取扱いについては安全性評価の観点や利活用・普及促進の観点から横断した検討が必要であるため、暗号技術評価委員会及び暗号技術活用委員会の協力も得ながら、暗号技術検討会の直下に、新たに「耐量子計算機暗号（PQC）リスト検討タスクフォース」を立ち上げ、検討を行うこととした。

なお、「耐量子計算機暗号（PQC）」という用語には、量子計算機でも解読困難な数学的課題に基づいた公開鍵暗号方式（例えばFIPS 203（ML-KEM）等）を指す場合と、量子計算機への耐性を有する暗号技術を指す場合の双方の意味で用いられるため留意を要する。例えば、「耐量子計算機暗号（PQC）への移行」という文脈においては、主に後者の意味として、共通鍵暗号の強度向上（例：AES-128→AES-256）も含むものとして用いられる。

[1] 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群において規定。

[2] ① Category 1・2の暗号の取扱い／② 暗号利用モードや認証暗号等の取扱い／③ ハイブリッド構成の取扱い／④ 耐量子計算機暗号（PQC）の安全性評価等の進め方

３．各委員会の活動報告

３．１．暗号技術評価委員会

３．１．１．活動の概要

暗号技術評価委員会は、CRYPTREC暗号リストに掲載されている暗号技術や電子政府システム等で利用される暗号技術の安全性維持及び信頼性確保のために安全性及び実装に係る監視及び評価を行う。主要な検討課題は以下のとおりである。

- ○ 暗号技術の安全性及び実装に係る監視及び評価
- ○ 暗号技術の電子政府推奨暗号リストからの降格
- ○ 暗号技術に関する注意喚起レポートのCRYPTRECホームページでの公表
- ○ 推奨候補暗号リストへの新規暗号（事務局選出）の追加
- ○ 新技術暗号等に係る調査

また、CRYPTREC暗号リストとは別の文書として、「暗号技術ガイドライン（耐量子計算機暗号）2026年度版」及び「耐量子計算機暗号の研究動向調査報告書2026年度版」の作成を目指し、調査を実施した。基本方針は以下のとおりである。

- ○ 耐量子計算機暗号（PQC）に関するガイドライン（2026年度版）及び研究動向調査報告書（2026年度版）を作成するため、耐量子計算機暗号（PQC）に関するワーキンググループを2025年度も引き続き設置した。
- ○ 2025年度はガイドライン（2026年度版）及び研究動向調査報告書（2026年度版）を作成するための調査を実施した。

さらに、CRYPTREC暗号リストへの掲載に向けた耐量子計算機暗号（PQC）の技術的検討、及び耐量子計算機暗号（PQC）への移行に向けた技術的検討を実施した。基本方針は以下のとおりである。

- ○ 耐量子計算機暗号（PQC）であるML-KEMの安全性・実装性能に関する評価を外部評価により実施し、評価結果に基づき、外部評価報告書「耐量子計算機暗号ML-KEMの安全性に関する調査及び評価」及び「CRYPTREC: ML-KEM Evaluation Report」を作成した。
- ○ 耐量子計算機暗号（PQC）への移行に関する技術動向調査を外部評価により実施し、調査結果に基づき、外部評価報告書「耐量子計算機暗号への移行に関する技術動向調査」を作成した。

これらの課題について2025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 2025（暗号技術評価委員会報告）に掲載する。

３．１．３．CRYPTREC暗号リストにおける仕様書参照先の変更

公開鍵暗号（鍵共有）であるECDHに関し、CRYPTREC暗号の仕様書一覧に掲載されている仕様書参照先を更新した。さらに、CRYPTREC暗号リストにおける注釈として「（注２）使用するMACはHMAC又はCMACに限る。」を追加した。

３．１．４．外部評価：耐量子計算機暗号ML-KEMの安全性・実装性能に関する評価及び調査

３．１．４．１．背景

2022年７月５日にNISTから耐量子計算機暗号（PQC）の標準化方式として、公開鍵暗号１方式と電子署名３方式が発表された。これら４方式のうち、格子に基づく公開鍵暗号方式であるML-KEMはFIPS 203として、格子に基づく署名方式であるML-DSAはFIPS 204として、ハッシュ関数に基づく署名方式であるSLH-DSAはFIPS 205として、それぞれ2024年８月13日に標準化された。

2024年度暗号技術検討会において、耐量子計算機暗号（PQC）への対応について議論が行われ、CRYPTREC暗号リストへの掲載に向けた耐量子計算機暗号（PQC）の技術的検討と、耐量子計算機暗号（PQC）への移行方針の検討を両輪として並行に進めていくべきであるとの合意が得られた。これに伴い、ML-KEM、ML-DSA及びSLH-DSAの安全性・実装性能の評価を先行して実施することで合意された。

2025年度第１回暗号技術評価委員会において、ML-KEMの安全性・実装性能に関する評価及び調査を実施することが承認された。

３．１．４．２．評価・実施概要

ML-KEMの安全性評価に関する１件目の外部評価を安田雅哉様（立教大学）に依頼した。選出理由と依頼内容は次のとおり。

(1) 選出理由

ML-KEMの安全性を支える数学問題とその数学問題の求解アルゴリズムにおける計算量見積に関して広い知見をお持ちである。当該分野に関する数多くの実績があるとともに、「CRYPTREC暗号技術ガイドライン（耐量子計算機暗号）」の第３章「格子に基づく暗号技術」の執筆における主担当者としての実績がある。

(2)依頼内容

耐量子計算機暗号ML-KEMの安全性について、公開されている解析手法や評価結果の有無を調査し、存在する場合はその影響範囲等についてまとめるなど、安全性評価を実施した上で報告書を作成する。

ML-KEMの安全性評価に関する２件目の外部評価に加え、実装性能評価に関する外部評価を勝又秀一様（PQShield）に依頼した。選出理由と依頼内容は次のとおり。

(1) 選出理由

勝又様は格子暗号の安全性評価やその応用先である暗号プロトコルの安全性評価に関する

広い知見をお持ちであり、当該分野に関する数多くの実績がある。また、共同執筆者に関しては、実装性能評価に関する広い知見をお持ちであり、当該分野に関する数多くの実績がある。

(2) 依頼内容

耐量子計算機暗号ML-KEMの安全性・実装性能について、公開されている解析手法や評価結果の有無を調査し、存在する場合はその影響範囲等についてまとめるなど、安全性と実装性能を評価した上で報告書を作成する。

３．１．４．３．外部評価報告書の概要：ML-KEMの安全性に関する評価及び調査

ML-KEMの安全性に関する評価結果は、以下のとおり。

(1) 安全性証明

○ 古典ランダムオラクルモデルにおいて、使用される２つのハッシュ関数がランダムオラクルと仮定した場合、タイトなIND-CCA2安全性を持つことが確認された。

○ 量子ランダムオラクルモデルにおいて、IND-CCA2安全性はノンタイトであり漸近的な保証を与えるにとどまることが確認された。しかし、耐量子性の観点で十分に高い信頼性を有する結果とみなされている。

(2) Module-LWE問題に対する計算量見積

① BKZシミュレーション[3]とdimension-for-free技術に基づく計算量見積

Primal攻撃の計算量（攻撃に必要なゲートコストとメモリ量）を見積もった結果[4]、攻撃に必要なゲートコストは、ML-KEMの全てのパラメータセットにおいてNIST安全性水準を上回っており、十分な安全性を有することが確認された（表3.1-1）。

表3.1-1. BKZシミュレーションとdimension-for-free技術に基づくprimal攻撃の計算量見積

	ML-KEM-512	ML-KEM-768	ML-KEM-1024
NIST安全性レベル	レベル1 (AES-128相当)	レベル3 (AES-192相当)	レベル5 (AES-256相当)
要求されるゲートコスト（ビット）	**143**	**207**	**272**
攻撃に必要なゲートコスト（ビット）	**151.5**	**215.1**	**287.3**
攻撃に必要なメモリ量（ビット）	93.8	138.5	189.7

② 幾何級数仮定（GSA）とMATZOV計算量モデルに基づく計算量見積

Primal攻撃、dual攻撃及びhybrid攻撃の計算量を見積った結果、攻撃に必要なゲートコストは、ML-KEMの全てのパラメータセットにおいてNIST安全性レベルの水準を下回っていることが確認された（表3.1-2）。しかし、これらの見積では計算量に影響しうる重要な性質[5]が考慮されておらず、計算量見積が過小評価されている可能性がある。これらの性質を考慮した

[3] BKZはBlock-Korkine-Zolotarevの略であり、格子基底簡約を行うBKZアルゴリズムのシミュレーターを指す。

[4] 既存研究において、dual攻撃がprimal攻撃と比較して計算コストがかかると予想されているため、ここではPrimal攻撃の計算量見積のみ提供されている。

[5] 例えば、篩処理のProgressive化による格子基底の理想的な挙動からのずれ、BDGL型篩処理で発生するオーバーヘッド等がある。

場合には、攻撃者に最も有利な状況を仮定しても、NIST安全性水準を上回ることが議論されている。

表3.1-2. 幾何級数仮定（GSA）とMATZOV計算量モデルに基づく計算量見積

	ML-KEM-512	ML-KEM-768	ML-KEM-1024
NIST安全性レベル	レベル1 (AES-128相当)	レベル3 (AES-192相当)	レベル5 (AES-256相当)
要求されるゲートコスト（ビット）	**143**	**207**	**272**
Primal攻撃のゲートコスト（ビット）	**140.2**	**201.0**	**270.7**
Dual攻撃のゲートコスト（ビット）	149.9	214.3	288.5
Hybrid攻撃のゲートコスト（ビット）	**139.7**	**196.4**	**262.3**

(3) Module構造を考慮した攻撃

Module構造を考慮した攻撃は、現時点において、Module構造を考慮しない攻撃を上回るものではないということが確認された。

(4) 暗号強度に関する考察

① 安田様の見解

表3.1-1に示すとおり、攻撃に必要とされるゲートコストは、ML-KEMの全てのパラメータセットにおいてNIST安全性水準を上回っており、十分な安全性を有する。さらに、最新の技術動向を踏まえても、現時点では表3.1-1の評価結果が覆る可能性は低い。

② 勝又様の見解

調査対象とした全ての攻撃クラスにおいて、ML-KEM のいずれのパラメータセットに対しても、現実的な脅威とみなせる脆弱性は現在のところ発見されていない。さらに、具体的な計算量見積に基づく評価の結果、古典計算及び量子計算の双方について攻撃者側に極めて有利な仮定を置いた場合であっても、NIST 安全性レベルに対して十分な安全性マージンを有していると考えられる。

３．１．４．４．外部評価報告書の概要：ML-KEMの実装性能に関する評価及び調査

ML-KEMの実装性能に関する評価結果は、以下のとおり。

(1) サイドチャネル攻撃耐性

秘密鍵の復元につながるサイドチャネル攻撃の可能性が確認された。これに対する対策として、マスキング及びハイディングの適用が推奨される。

(2) ハードウェア実装性能

回路面積の最適化実装[XL21]、計算時間の最適化実装[DMG23]及びサイドチャネル攻撃対策が施された実装[Kam+22]について紹介された（表3.1-3）。

表3.1-3. ML-KEM（Kyber）のFPGA実装比較

参考文献	パラメータ	計算時間（μs）			回路面積		FPGA
		KeyGen	Encap	Decap	LUT(x1000)	FF(x1000)	
[XL21]	512	23.4	31.5	41.4	**7.4**	**4.6**	Artix-7
	768	39.2	49.2	62.4	**7.4**	**4.6**	
	1024	58.3	70.3	86.4	**7.4**	**4.6**	
[DMG23]	512	**10.0**	**14.7**	**20.5**	9.5	8.5	Artix-7
	768	**12.0**	**17.0**	**22.2**	10.5	9.8	
	1024	**16.2**	**21.7**	**26.4**	11.6	11.1	
[Kam+22] (M+H)	512	-	**88.1**	**137.7**	**163.6**	-	Virtex-7

※ M+Hはマスキング対策とハイディング対策の両方を施した実装を表す。

(3) ソフトウェア実装性能

① 計算時間

OpenSSL 3.6.0を使用して測定した結果、ML-KEMはECDHと同等以上の性能を発揮することが確認された（表3.1-4）。

表3.1-4. ECDHとML-KEMにおける計算時間の比較（ミリ秒）

鍵交換アルゴリズム		安全性レベル	KeyGen	Encap	Decap
古典	EC X25519	1†	0.027	0.058	0.029
	EC P-256	1†	0.008	0.058	0.047
	EC P-384	3†	0.088	0.327	0.229
	EC P-521	5†	0.098	0.341	0.226
量子	ML-KEM-512	1	**0.020**	**0.014**	**0.023**
	ML-KEM-768	3	**0.031**	**0.020**	**0.032**
	ML-KEM-1024	5	**0.047**	**0.028**	**0.043**
ハイブリッド	X25519 + ML-KEM-768	1† + 3	0.061	0.076	0.060
	P-256 + ML-KEM-768	1† + 3	0.044	0.076	0.076
	P-384 + ML-KEM-1024	3† + 5	0.143	0.344	0.256

※ †はECの古典安全性レベルを表しており、耐量子安全性は考慮していない。

② 帯域幅

ML-KEMとECDHの帯域幅（具体的には、鍵長と暗号文長）を比較した結果、最大で25倍の差が生じていることが確認された（表3.1-5）。なお、帯域幅の増加は対処可能であり、インターネット上でのML-KEMの使用を妨げるものではないことが実証された。

(4) 実装性能に関する考察

ML-KEMの計算時間は従来方式と比べて高速である一方、鍵長や暗号文長が増加するため、メモリ制約があるデバイス等においては実装上の課題が生じる可能性があるが、それ以外の用途においては問題なく利用できる。

サイドチャネル攻撃に対して厳密に保護されていることを前提として、政府及び重要インフラシステムへの広範な展開にも適していると考えられる。

表3.1-5. ECDHとML-KEMにおける帯域幅（鍵長と暗号文長）の比較（バイト）

鍵交換アルゴリズム		安全性レベル	カプセル化鍵	暗号文
古典	EC X25519	1†	**32**	**32**
	EC P-256	1†	**65**	**65**
	EC P-384	3†	**97**	**97**
	EC P-521	5†	**123**	**123**
量子	ML-KEM-512	1	**800**	**768**
	ML-KEM-768	3	**1184**	**1088**
	ML-KEM-1024	5	**1568**	**1568**
ハイブリッド	X25519 + ML-KEM-768	1† + 3	1216	1120
	P-256 + ML-KEM-768	1† + 3	1249	1153
	P-384 + ML-KEM-1024	3† + 5	1665	1617

※ †は古典的な安全性レベルを表しており、耐量子安全性は考慮していない。

３．１．４．５．外部評価報告書に対する暗号技術評価委員会の見解

2025年度外部評価報告書に基づき、以下の結論を得た。

- ○ ML-KEMは、全てのパラメータセット（ML-KEM-512/768/1024）において、米国NISTが規定する安全性レベル1/3/5を満たしている。
- ○ ML-KEMは、従来方式と比較して高速である一方、鍵長及び暗号文長が増加するが、メモリ制約が厳しいデバイスを除き、実用上問題なく利用できる。
- ○ ML-KEMは、サイドチャネル攻撃対策が不十分な場合に脆弱性が生じる可能性があるものの、適切な対策の実装を前提とすれば、電子政府システムを含む多様なシステムへの広範な展開が可能である。

また、2025年度外部評価報告書は、ML-KEMの安全性・実装性能に関する技術動向調査として十分な内容を含んでいると考えられる。このため、本報告書をCRYPTRECの技術調査報告書とすることが承認された。

３．１．５．外部評価：耐量子計算機暗号の移行に関する技術動向調査

３．１．５．１．背景

2022年７月５日にNISTから耐量子計算機暗号（PQC）の標準化方式として、公開鍵暗号１方式と電子署名３方式が発表された。これら４方式のうち、格子に基づく公開鍵暗号方式ML-KEMはFIPS 203として、格子に基づく署名方式ML-DSAはFIPS 204として、ハッシュ関数に基づく署名方式SLH-DSAはFIPS 205として、それぞれ2024年８月13日に標準化された。

2024年度暗号技術検討会において、耐量子計算機暗号（PQC）への対応について議論が行われ、CRYPTREC暗号リストへの掲載に向けた耐量子計算機暗号（PQC）の技術的検討と、耐量子計算機暗号（PQC）への移行方針の検討を両輪として並行に進めていくべきであるとの合意が得られた。

2025年度第１回暗号技術評価委員会において、耐量子計算機暗号（PQC）への移行に関する技術動向調査を外部評価により実施することが承認された。

３．１．５．２．評価・調査実施概要

鈴木茜様（日立製作所）に外部評価を依頼した。選出理由と依頼内容は次のとおりである。

(1) 選出理由

暗号の2010年問題における政府認証基盤の暗号移行に関する事業に携わるほか、近年は耐量子計算機暗号（PQC）の導入に向けた政府動向や標準技術仕様の調査を実施するとともに、今後の耐量子計算機暗号（PQC）への移行に向けて認証基盤システムへの影響を検討するなど、当該分野における知識・経験が豊富である。

(2) 依頼内容

耐量子計算機暗号（PQC）への移行に関する技術動向を調査し、公開情報を基にまとめ、考察等を行い、報告書を作成する。

３．１．５．３．外部評価報告書の概要

(1) 耐量子計算機暗号（PQC）への移行時・導入時における課題

耐量子計算機暗号（PQC）への移行時及び導入時に直面する技術的課題について、用途別（署名用途／守秘用途／鍵共有用途）に整理された。

(2) 耐量子計算機暗号（PQC）導入へのアプローチ

耐量子計算機暗号（PQC）の導入に伴う全体プロセス、移行計画策定の検討事項、用途別の導入アプローチ、及び段階的な移行モデルであるハイブリッド構成の位置付けについて整理された。特に、ハイブリッド構成は、既存システムとの連続性を確保しつつ、新たな暗号方式を段階的に導入するための現実的な構成例として位置付けられていると整理された。

(3) ハイブリッド構成に関する解説

ハイブリッド構成は単一の技術要素ではなく、複数のレイヤーにまたがる設計課題を内包していることが確認された。この認識に基づき、ハイブリッド構成を目的（後方互換性の確保及び安全性の維持）と運用主体（アルゴリズム／プロトコル／システム）の各レイヤーから体系的に整理された。

(4) ハイブリッド構成の安全性に関する解説

標準化文書に記載された内容を基に、ハイブリッド構成における安全性について整理された。特に、安全性が成立するための条件と、それらの条件が実際の運用においてどのような構成要素に依存して具体化されるのかという点に着目された。

(5) ハイブリッド構成の実装・運用に関する解説

耐量子計算機暗号（PQC）及びハイブリッド構成の実装に関する主要なOSSの整備状況と、実運用環境における実装・移行事例が整理された。特に、国際会議PQC Conferenceで報告された以下の事例について紹介された。

① 実運用を想定した耐量子計算機暗号（PQC）への移行及び性能評価の事例
② Web PKIにおける段階的な耐量子計算機暗号（PQC）導入の事例
③ PKI階層設計におけるハイブリッド構成の活用事例
④ S/MIME電子メールにおけるハイブリッド構成の実装事例

(6) 耐量子計算機暗号（PQC）への移行に関わる標準化動向の調査

耐量子計算機暗号（PQC）への移行期におけるハイブリッド方式の取扱いに着目し、国際的な標準化団体及び関連組織における検討状況が整理された。

表3.1-6. 調査対象組織の一覧

No.	組織名	URL
1	NIST	https://www.nist.gov/
2	IETF	https://www.ietf.org/
3	ITU	https://www.itu.int/
4	ETSI	https://www.etsi.org/
5	IEEE	https://www.ieee.org/
6	ISO	https://www.iso.org/
7	ASC X9	https://x9.org/
8	NSA	https://www.nsa.gov/
9	CSA	https://cloudsecurityalliance.org/
10	PQCRYPTO	https://pqcrypto.eu.org/
11	PQCC	https://pqcc.org/

(7) 調査結果に関する考察

2020年度外部評価報告書「ハイブリッドモードの技術動向調査」の公開時点[6]では、ハイブリッドモードは概念レベルにとどまり、標準化活動も議論が始まったばかりの段階であった。一方、2025年以降は主要な標準化機関において本格的な標準化活動が進展しており、その動きが活発化していることが確認された。

特に、2026年１月時点では、ハイブリッド構成は主要な標準化機関において技術仕様として確立されており、TLS、X.509、VPN等の各種プロトコルやPKI基盤に関しても、具体的な実装指針が整備されていることが確認された。

これらの具体的な整備により、ハイブリッド構成は単なる概念段階を超えて、既存プロトコル・証明書・運用基盤の中で「移行期に採用可能な実装構成」として体系的に確立されたと評価できる。

３．１．５．４．外部評価報告書に対する暗号技術評価委員会の見解

2025年度外部評価報告書は、耐量子計算機暗号（PQC）への移行に関する技術動向調査として十分な内容を含んでいると考えられる。このため、本報告書をCRYPTRECの技術調査報告書とすることが承認された。

[6] 2020年度の報告書では、ハイブリッド構成のことをハイブリッドモードと称していた。

３．１．６．暗号技術調査WG（耐量子計算機暗号）

2021年度から活動を継続している暗号技術調査WG(耐量子計算機暗号)（以下「PQC WG」という。）の活動背景と2025年度の活動報告を記述する。

３．１．６．１．活動の背景

2020年度第２回暗号技術検討会において、大規模な量子コンピュータが実用化された後でも量子攻撃に対して安全性を確保でき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活動が各国で進められていることから、PQC WGを設置することが承認された。活動内容として、２年間かけて調査活動を行い、耐量子計算機暗号（PQC）に関するガイドライン・調査報告書を作成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」及び「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新をWGで実施することが承認された。2023年３月に2022年度版のガイドライン及び調査報告書を公開した。

2023年度第１回暗号技術評価委員会において、耐量子計算機暗号（PQC）関連の技術開発、標準化活動が世界的に活発であることから、引き続き、PQC WGを設置することが承認された。２年間かけて耐量子計算機暗号（PQC）に関する技術動向調査を行い、耐量子計算機暗号（PQC）に関するガイドライン・調査報告書（2024年度版）を作成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」及び「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新をWGで実施することが承認された。2025年３月に2024年度版のガイドライン及び調査報告書を公開した。

2025年度暗号技術評価委員会において、引き続き、PQC WGの設置及び以下の活動を行う事が承認された。

- ○ NISTにおける耐量子計算機暗号（PQC）の標準化において３件の標準化文書（FIPS 203から205まで）が公開され、今後も２件の標準化（FALCON及びHQC)、並びに追加署名の選定プロジェクトが進行中であることをはじめ、世界各国の機関において技術開発、標準化活動が引き続き活発であり、情勢が流動的であることから、2024年度版の調査報告書・ガイドラインが出版された以降の研究技術動向を2026年度末までに調査・把握し、調査報告書・ガイドラインを作成する。
- ○ 「素因数分解の困難性に関する計算量評価」及び「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新についても検討し、更新する。

上記活動の成果は以下のとおりであり、2025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。

３．１．６．２．耐量子計算機暗号（PQC）に関する調査報告書・ガイドライン

調査報告書・ガイドライン執筆方針の基本的な方針は2024年度版調査報告書・ガイドラインを踏襲し、以下のとおりである。

- ○ 取り扱う耐量子計算機暗号（PQC）は2024年度版調査報告書にあるとおり「古典アルゴリズムの組み合わせにより定式化され，かつ耐量子計算機性を持つことを技術的に判断できる暗号方式」とし、特に公開鍵暗号である公開鍵暗号方式（Public-key encryption)、署名方式

（Digital signature）及び鍵共有（Key exchange）に関して調査を行う。

○ 調査対象を安全性の根拠となる計算問題に応じて分類し、格子に基づく暗号、符号に基づく暗号、多変数多項式に基づく暗号、同種写像に基づく暗号及びハッシュ関数に基づく署名に分けて調査・執筆を行う。

○ 耐量子計算機暗号（PQC）の研究成果が発表される主要な国際会議Crypto、Eurocrypt、Asiacrypt及びPQCryptoを中心に、開発・標準化（ISO, IETF, NIST, ETSI等）の動向に関しても2026年９月30日までの情報を可能な限り調査する。その他主要な動向があれば可能な限り取り上げる。

○ 調査報告書には可能な限り詳細な情報を記載し、ガイドラインでは暗号初学者を対象とし技術的な詳細を省き簡略化する。

○ 調査報告書・ガイドラインの章立ては以下を予定する。第８章が追加されるかどうかは来年度の議論とする。また、A.2、A.3の節タイトルに関しては計算問題の名称との組み合わせにより読みやすい形に各章担当者が調整することで合意された。

表3.1-7　ガイドラインの章立て

章	タイトル
１	はじめに
２	耐量子計算機暗号（PQC）の活用方法
３	格子に基づく暗号技術
４	符号に基づく暗号技術
５	多変数多項式に基づく暗号技術
６	同種写像に基づく暗号技術
７	ハッシュ関数に基づく署名技術
８	総括（追加される場合）
３章以降の構成（Ａ章の場合：Ａは３～７を表す）	
Ａ．１．	安全性の根拠となる問題（例：LWE問題、シンドローム復号問題）
Ａ．２．	暗号方式の基本設計
Ａ．３． Ａ．３．１． Ａ．３．２． Ａ．３．３． ・・・	実用的な暗号方式 暗号方式１（例：CRYSTALS-KYBER, Classic McEliece） 暗号方式２ 暗号方式３ ・・・
Ａ．４．	まとめ

３．１．６．３．「素因数分解の困難性に関する計算量評価」及び「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図の更新

「素因数分解の困難性に関する計算量評価」及び「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図（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」及び「今後の公開鍵暗号のパラメータ選択」）を決定した。2025年度において、対応方針は以下のとおりとなっている。

予測図の取扱い対応方針

＜今後の予測図の取扱い＞

(1) いわゆるムーアの法則を仮定して外挿線を年度末からプラス20年後まで従来どおり直線で引き、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。

(2) 予測図における500位のプロットに係る削除について検討した、TOP500.Orgにおける500位のプロットは2024年度までとし、以降の外挿線を削除する。

(3) 暗号強度要件（アルゴリズム及び鍵長）に関する設定基準[7]における基本設定方針に沿ったパラメータ（利用可・移行完遂期間、2022年３月策定以降）を黄色部分で示した。

(4) セキュリティ強度が112ビットセキュリティ相当のBinary Fields及びKoblitz Curves （群位数233ビット）を追記した（図3.1-2）。

＜今後の公開鍵暗号のパラメータ選択＞

(5) 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等、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、暗号技術検討会及び暗号技術活用委員会や関係各所等を含めて検討する。

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。

予測図の更新について

素因数分解問題の困難性及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2025年６月と11月のベンチマーク結果を追加して予測図の更新を行った（図3.1-1及び図3.1-2）。

[7] https://www.cryptrec.go.jp/list/cryptrec-ls-0003-2022r1.pdf

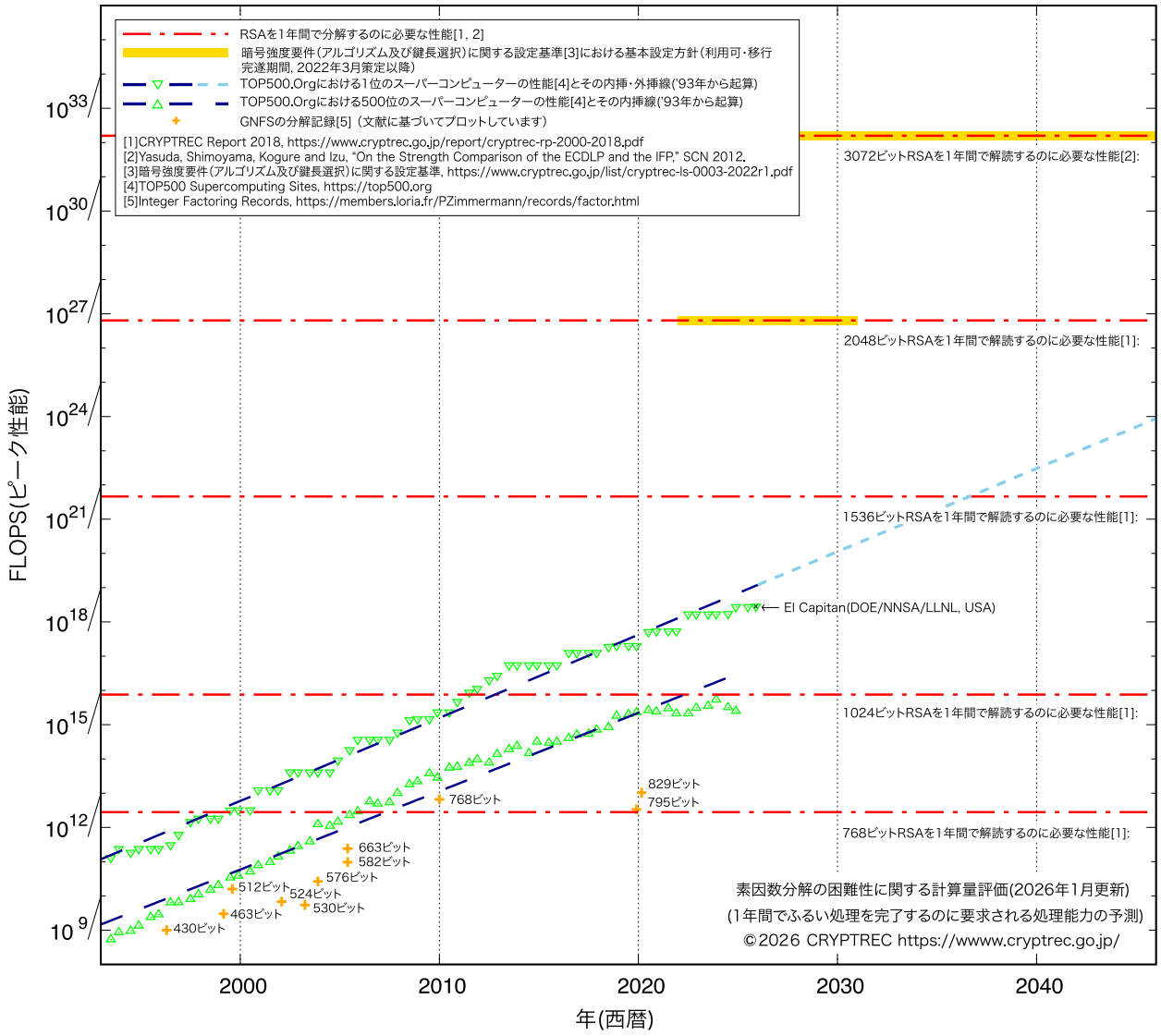


図3. 1-1：素因数分解の困難性に関する計算量評価（2026年１月更新）[8]

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

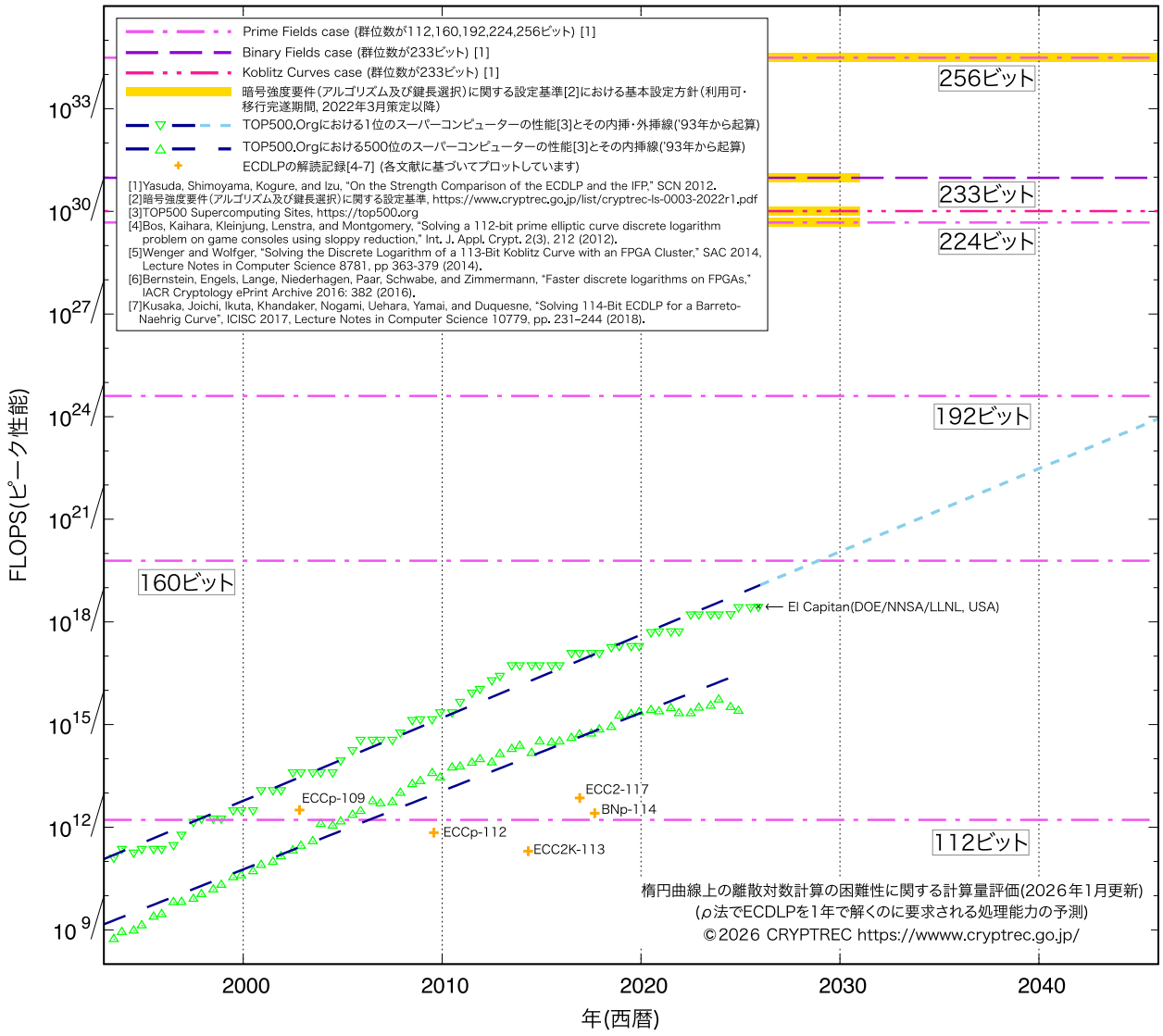


図3.1-2：楕円曲線上の離散対数計算の困難性に関する計算量評価（2026年１月更新）[9]

[9] 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。

３．１．７．暗号技術評価委員会の開催実績

暗号技術評価委員会は計２回開催された。各回会合の概要は表3.1-8のとおりである。

表3.1-8　暗号技術評価委員会の開催状況

回	開催日	議案
第１回	2025年７月１日	○ 暗号技術調査 WG（耐量子計算機暗号）の活動計画案に関する審議 ○ 耐量子計算機暗号（PQC）への対応方針と 2025 年度外部評価「ML-KEM の安全性・実装性能に関する評価及び調査」及び「耐量子計算機暗号への移行に関する技術動向調査」の実施に関する審議 ○ CRYPTREC 暗号リストにおける仕様書参照先の変更に関する審議 ○ 監視状況報告
第２回	2026年３月３日	○ 暗号技術調査 WG（耐量子計算機暗号）の活動内容に関する報告 ○ メール審議「ECDH における仕様書参照先の変更」及び「外部評価スケジュールの変更と ML-DSA 外部評価の実施」に関する報告 ○ 外部評価スケジュールの再変更と SLH-DSA 外部評価の実施に関する審議 ○ 外部評価報告書「ML-KEM の安全性・実装性能に関する評価及び調査」及び「耐量子計算機暗号への移行に関する技術動向調査」に関する概要報告、並びに本外部評価に関する審議 ○ 監視状況報告 ○ 2025 年度暗号技術評価委員会活動報告案の確認 ○ 2026 年度暗号技術評価委員会活動計画案の確認 ○ CRYPTREC Report 2025 の目次案の確認

また、PQC WGは計２回開催した。2025年度のPQC WG各回の概要は表3.1-9のとおりである。

表3.1-9　PQC WGの開催状況

回	開催日	ガイドラインの議論・決定・報告
第１回	2025年８月４日	○ 追記・改定の方針について議論 ○ 執筆担当者を議論・決定
第２回	2026年１月22日	○ 追記・改定すべき項目及びその章立ての決定 ○ 調査の中間報告

３．２．暗号技術活用委員会

３．２．１．活動の概要

2025年度の活動概要は以下のとおりである。詳細については、CRYPTREC Report 2025（暗号技術活用委員会報告）を参照されたい。

(1) 耐量子計算機暗号（PQC）の取扱いに係る検討

耐量子計算機暗号（PQC）をめぐる社会的動向を踏まえ、耐量子計算機暗号（PQC）の取扱基準や位置付け・記載内容等についての検討を行った。具体的には、日本における「耐量子計算機暗号（PQC）への移行（方針）」を検討する際の素材としてもらうため、「耐量子計算機暗号（PQC）への移行（方針）」の政策的側面からの整理、及び「CRYPTREC暗号リスト」における耐量子計算機暗号（PQC）の位置付けや移行ルールを変更すべきかどうかを検討し、暗号技術活用委員会としての見解を取りまとめた。

(2) クラウドにおける鍵管理ガイダンスの作成

クラウド鍵管理ガイダンスWGを設置し、クラウドサービスを利用した情報システムにおける暗号鍵管理のガイダンス作成を開始した。2026年度末での完成に向けて、今年度はガイダンスの骨子を整理した。

(3) 「暗号鍵管理システム設計指針（基本編）」の改訂

暗号鍵管理システムの設計に関わる中心となる解説書である「暗号鍵管理システム設計指針（基本編）（以下「設計指針」という。）」の作成から約５年が経過し、記載に古い箇所がある、誤記がある等の問題が見つかったため、改訂方針を議論した。

３．２．２．耐量子計算機暗号（PQC）の取扱いに係る検討

(1) 「耐量子計算機暗号（PQC）への移行（方針）」の政策的側面からの取りまとめ

各国政府・公的機関等が発行している耐量子計算機暗号（PQC）への移行に関する政策やガイドラインの情報について時系列的観点での整理を行った。

G7での国際協調方針が打ち出されたこともあり、欧米各国とも国家安全保障システムや高セキュリティシステム等は2030年頃を、その他のシステムは2035年を耐量子計算機暗号（PQC）への移行完了時期に設定した移行方針が打ち出されている。併せて、耐量子計算機暗号（PQC）ではない従来の暗号技術、特に公開鍵暗号についての取扱いについても言及されつつあることが分かった。

これらの結果を踏まえて、2026年度は「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」の改定を進めることとする。

(2) 「CRYPTREC暗号リストでの取扱いルール」を変更すべきかどうかの検討

以下の観点を中心に耐量子計算機暗号（PQC）に対応したCRYPTREC暗号リストの在り方について議論し、暗号技術活用委員会としての見解をまとめた。また、これらの見解を踏まえつつ作成

された「耐量子計算機暗号（PQC）に対応したCRYPTREC暗号リストの在り方（案）」に対してその内容を確認し、暗号技術活用委員会としても同意した。

① 耐量子計算機暗号（PQC）リストの形式

現行のCRYPTREC暗号リスト（「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」）の技術分類をそのまま使うか、「耐量子計算機暗号（PQC）」のカテゴリを追加するか、あるいは、それらのリストとは別の「量子計算機耐性を備えた暗号方式に関わる独立したリスト（耐量子計算機暗号（PQC）リスト）」を作るべきかを議論した。

その結果、現時点においては既存のCRYPTREC暗号リストに耐量子計算機暗号（PQC）を加えるのではなく、新たに量子コンピュータに耐性のある暗号のみを記載したリストを作成したほうがよいとの見解で一致した。また、新たなリストに掲載するアルゴリズムについて、量子計算機耐性を備えた公開鍵暗号だけではなく、共通鍵暗号やハッシュ関数等も含めて、量子計算機耐性を備えた暗号方式全般を含むリストとするのがよいとの見解で一致した。さらに、アルゴリズムの強度を規定するパラメータもリスト中に記載するのがよいとの意見もあった。

これは、耐量子計算機暗号（PQC）リストを参照することで量子計算機耐性を備えた一通りの暗号方式を選択可能であり、利用者にとって利便性が高いと考えられるためである。

② 移行ルール・選定ルールについての検討

「耐量子計算機暗号（PQC）リストを新たに作成すること自体はよく、そのためのルールが設けられることも理解できる。ただし、電子政府推奨暗号リストとの関係は整理すべき」との指摘があった。例えば、両リストの使い分けをどうするのか、両リストは将来的に一本化されていくのか、耐量子計算機暗号（PQC）リストと推奨暗号リストの両方に記載されるアルゴリズムはあるのか、耐量子計算機暗号（PQC）リストでも将来は監視暗号リストが必要である、などである。

これらの意見は、「耐量子計算機暗号（PQC）に対応した CRYPTREC 暗号リストの在り方（案）」を作成する際の参考としてもらうこととした。

③「耐量子計算機暗号（PQC）に対応したCRYPTREC暗号リストの在り方について（案）」に対する意見

「耐量子計算機暗号（PQC）に対応した CRYPTREC 暗号リストの在り方（案）」に対して次のような意見があった。

- ○ 「耐量子計算機暗号（PQC）リスト」の説明文に関連して、例えば、現行の電子政府推奨暗号リストにのみ掲載されている方式について、いずれ耐量子計算機暗号（PQC）リストにも掲載される可能性があるのか、あるいは耐量子計算機暗号（PQC）リストには掲載されない方式であるのかが明確でない等の課題があるため、誤解が生じないような工夫がいる。
- ○ 耐量子計算機暗号（PQC）リストは、プリミティブとなるアルゴリズムだけの記載とするのがよい。ハイブリッド構成を含めると暗号リストとして複雑になることが予想され、ハイブリッド構成に関しては、例えばTLS暗号設定ガイドラインのようなプロトコルやアプリケーションを対象に作成するガイドラインの中で対応するのがよいのではないか。

３．２．３．クラウドにおける鍵管理ガイダンス

今年度はクラウド鍵管理ガイダンスの骨子となる資料を作成した。ここでは、ガイダンスの骨格となる目次構成を中心にまとめる。

クラウド鍵管理ガイダンスの目次案は以下のとおりである。

１．はじめに
２．基礎知識
３．クラウド鍵管理サービスについて
４．クラウド鍵管理サービスに関わる責任分界
５．暗号鍵管理システムのフレームワーク要求からの整理
６．その他
Appendix. 参考資料

１章ではイントロダクションとして、本ガイダンスの位置付けや想定読者、スコープをまとめる。２章では、クラウド鍵管理に関わる基礎技術や政府システムにおけるクラウドサービス利用時の要件等の基礎知識をまとめる。３章では、クラウドサービスプロバイダ（CSP）が提供するクラウド鍵管理サービスを体系化し、比較する。４章では、クラウド鍵管理サービスにおけるCSPと利用者の責任分界の原則を説明する。５章では、NIST SP 800-130を基に鍵管理における管理策を抽出した管理策一覧表を示し、この管理策一覧表を利用してクラウド鍵管理サービス利用時の責任分界を整理した例を説明する。さらに、利用者側に実施責任がある事項における注意点について、より詳細に説明する。６章では、本ガイダンスで取り上げていない周辺事項に触れる。

３．２．４．「暗号鍵管理システム設計指針（基本編）」の改訂

2020年に発行した「暗号鍵管理システム設計指針（基本編）」の修正について検討した。検討の背景は、2023年度から2024年度までに実施した「暗号鍵管理ガイダンスPart 2」の作成過程で、同ガイダンスの親文書に相当する「設計指針」に対する修正の意見があったことである。「設計指針」の執筆から約５年が経過し、記載内容の最新化や明確化を行うのが適当な箇所や誤記も多数見つかったため、修正すべき箇所と修正方針、修正案を議論した。結果として、「設計指針v1.1」として改訂を行う方針とした。

３．２．５．暗号技術活用委員会の開催状況

2025年度の暗号技術活用委員会での審議概要は表3.2-1のとおりである。

表3.2-1　暗号技術活用委員会の開催状況

回	開催日	議案
第１回	2025年７月18日	○ 2025 年度暗号技術活用委員会活動計画の確認 ○ 2025 年度クラウド鍵管理ガイダンス WG 活動計画の審議 ○ 耐量子計算機暗号（PQC）の取扱いに関する検討 ○「暗号鍵管理システム設計指針（基本編）」の修正について
第２回	2026年２月24日	○ 耐量子計算機暗号（PQC）の取扱いに関する検討 ○ 2025 年度クラウド鍵管理ガイダンス WG 活動状況及び WG 活動報告の審議 ○ 「暗号鍵管理システム設計指針（基本編）」の修正案の審議 ○ 2025 年度暗号技術活用委員会活動報告案について

４．2026年度のCRYPTRECの活動について

CRYPTRECでは、2026年度も、電子政府推奨暗号等の安全性を評価・監視するとともに、暗号技術の更なる普及促進を行うべく検討を進める。

暗号技術検討会においては、「耐量子計算機暗号（PQC）リスト検討タスクフォース」を設置し、CRYPTREC暗号リストにおける耐量子計算機暗号（PQC）への対応に関する課題等の整理を行う。

例年、暗号技術検討会は年度末に開催していたが、同タスクフォースにおける検討状況や暗号技術評価委員会におけるFIPS 204（ML-DSA）及びFIPS 205（SLH-DSA）に係る安全性・実装性能に関する調査及び評価の状況を踏まえながら、CRYPTREC暗号リストの速やかな改定に必要があれば、年度途中の開催も含めて開催することとする。

暗号技術評価委員会においては、「暗号技術調査WG（耐量子計算機暗号）」において、耐量子計算機暗号（PQC）に関する技術動向を継続して調査・把握するとともに、ガイドライン及び調査報告書を作成する。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同WGで検討し、更新を行う。また、CRYPTREC暗号リスト掲載に向けた耐量子計算機暗号（PQC）の技術的検討に資するための外部評価を実施する。具体的には、FIPS 204（ML-DSA）及びFIPS 205（SLH-DSA）の暗号技術について、安全性・実装性能に関する調査及び評価を行う。

暗号技術活用委員会においては、「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」について、耐量子計算機暗号（PQC）の取扱いを含めた形での見直しを実施するとともに、「TLS暗号設定ガイドライン」について耐量子計算機暗号（PQC）のサポートに向けたTLS1.3への移行や電子証明書の有効期限の短縮化等に対応するため、2027年度の見直しに向けた検討を実施する。また、「クラウド鍵管理ガイダンスWG」において、クラウドサービスを利用したシステムにおける暗号鍵管理の適切な設計・構築・運用のために、クラウド鍵管理ガイダンスを作成する。

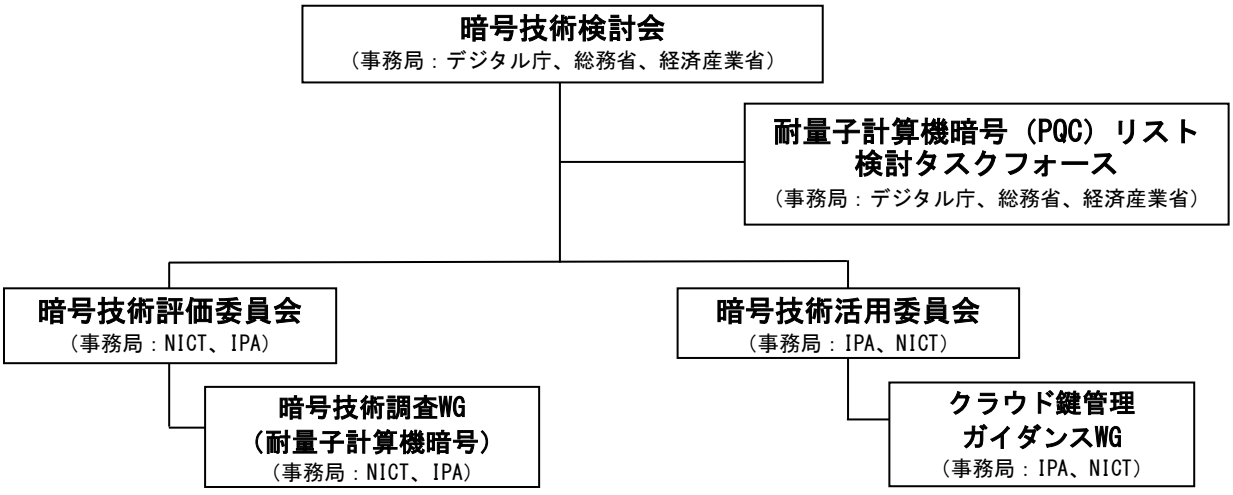


図4-1 CRYPTREC体制図（2026年度）（予定）