CRYPTREC
耐量子計算機暗号の研究動向調査報告書

2023 年 3 月

CRYPTREC
暗号技術調査ワーキンググループ（耐量子計算機暗号）

目次

第 1 章

はじめに

現在広く使用されている公開鍵暗号方式として RSA 暗号と楕円曲線暗号が挙げられる．RSA 暗号と楕円曲線暗号が安全であるためには，素因数分解問題や楕円曲線上の離散対数問題が計算量的に困難であることが必要である．これらの問題は現在普及しているコンピュータでは効率的に解くことはできないと信じられている．ただし，量子コンピュータの開発が十分に進むと Shor のアルゴリズム [40, 41] により整数の素因数分解や離散対数を高速に計算できるため，RSA 暗号と楕円曲線暗号の安全性は大きく低下する．そのため，古典コンピュータ[*1]上で効率的な実装が可能であり，かつ古典・量子双方のコンピュータを用いた攻撃に対しても安全性を確保できる公開鍵暗号方式が必要とされている．この安全性を確保した暗号方式が耐量子計算機暗号 (Post-Quantum Cryptography: PQC) と呼ばれている．同様に，共通鍵暗号方式においても量子コンピュータによって安全性は低下することが知られているが [15]，公開鍵暗号に比べるとその影響は小さいと考えられている．この影響について多くの報告があるが，その一つとして，電子政府推奨暗号の安全性を評価・監視し，暗号技術の適切な実装法・運用法を調査・検討するプロジェクトである CRYPTREC （図 1.1）による，量子コンピュータに対する共通鍵暗号の安全性の影響に関する 2019 年度の調査 [17] がある．これらの状況を踏まえ，本ガイドライン [11] 及び調査報告書では，PQC は共通鍵暗号方式を含まず，公開鍵暗号方式のみを示す言葉とする．本ガイドライン及び調査報告書は PQC に関する内容をまとめたものである．

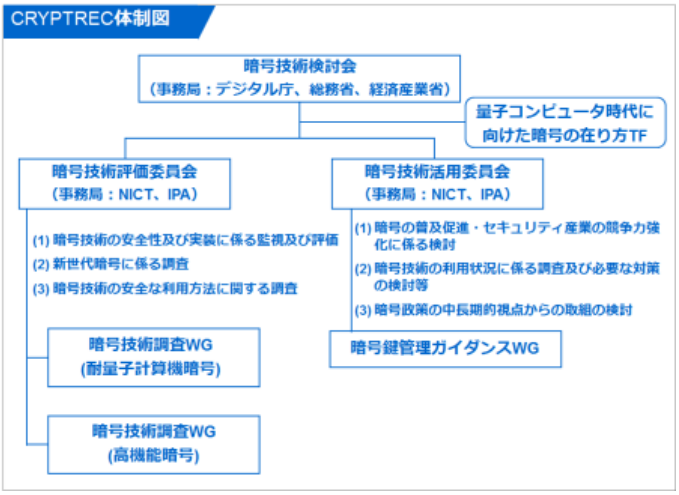


図 1.1: 2022 年度 CRYPTREC 体制図

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

近年の世界的な量子コンピュータの開発にともない，PQC に関する研究及びその標準化に向けた活動も世界各国の組織で実施されており，国内でも PQC の研究動向を把握する必要性が高まっている．2020 年度第 2 回暗号技術検討会において，2021 年度から暗号技術評価委員会の活動計画として 2 年をかけて PQC の研究動向を調査し，ガイドラインを作成することが決定された．暗号技術評価委員会は暗号技術調査ワーキンググループ (耐量子計算機暗号) を設置し，2021 年度及び 2022 年度において本ガイドライン及び調査報告書を作成した．

本ワーキンググループでは PQC の代表的な候補である 5 種類の分類（格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術，同種写像に基づく暗号技術，ハッシュ関数に基づく署名技術）について調査し，主に 2022 年 9 月 30 日までの調査結果をガイドラインと調査報告書にまとめた．ガイドラインは暗号初学者を対象としており，調査報告書は暗号についての知見のある技術者や専門家を対象としている．ガイドラインと調査報告書の第 1 章は共通であり，全ての読者にむけた内容をまとめた．ガイドラインの第 2 章には暗号初学者向けに PQC の活用方法に関する内容，特に守秘・鍵共有・署名のための PQC の利用などについて記載している．この章は調査報告書には記載していない．ガイドラインの第 3 章以降，そして調査報告書の第 2 章以降は，暗号技術に携わる研究者及び技術者を読者として想定し，PQC の代表的な候補である 5 種類の分類をまとめた．ただし，これらの章ではガイドラインの記載内容は調査報告書の簡略版となっており，ガイドラインでは専門的な内容を省略し，暗号初学者が代表的な PQC 方式を把握するために最小限の内容のみを記載した．

1.1 耐量子計算機暗号 (PQC) の必要性について

量子コンピュータは重ね合わせ・エンタングルメント等の量子的な物理現象を用いて計算を行うコンピュータの総称である [50, 22]．基本的な計算操作と物理的操作の対応関係を表すモデルにより，量子回路型計算，測定型量子計算，断熱型量子計算，トポロジカル量子計算，ホロノミック量子計算等に大別できる[*2]．この分類とは別に，量子アニーリング (Quantum Annealing: QA) と呼ばれる計算フレームワークが存在する．[*3] これらのうち，超伝導，イオントラップによる量子回路型コンピュータおよび，超伝導磁束量子ビットによる量子アニーリングを行うコンピュータは物理的なハードウェアの進化とプログラミング環境の進化により商用レベルにまで達し，非専門の技術者レベルでもクラウドを通じた利用が容易となったことから注目されている．そのため，先に述べた基本的操作の種類による分類の他に，応用目的から量子回路型と量子アニーリング型のみを取り上げそれぞれ量子ゲート型と量子アニーリング型という名称で対比されることもある [50, 第 2 章, p.11]．

超伝導を用いた量子回路型コンピュータの開発は米国の民間企業を中心にここ数年で急速に進展しており，2019 年 10 月に Google が 53qubits[*4] [1]，2021 年 12 月には Rigetti が 80qubits [37]，2022 年 11 月には IBM が 433qubits のプロセッサ [20] を発表している．また，中国においても中国科学技術大学が 66qubits の祖沖之 2.1 [53] を，百度 (Baidu) が 2022 年 8 月に 10qubits コンピュータ [2] を発表している．日本でも富士通が 2023 年度中の 64qubits 量子コンピュータの公開を目指している [38]．超伝導方式以外の開発も進んでおり，イオントラップ式では 2020 年 10 月に IonQ が 32qubits [7]，2022 年 6 月には Quantinuum が 20qubits [36] を発表している．シリコン量子ビット式では Intel，日立製作所が開発を進めている [45]．

2020 年代終盤までのロードマップとして 10-100 万 qubits を搭載し，量子誤り訂正を組み込むことによりほぼノイ

[*2] 分類に関しては [22, 48] および [24, Sect 1.6] を参照．

[*3] シミュレーテッドアニーリング (Simulated Annealing: SA) に類似したアルゴリズムを量子的に構成した Apolloni ら [13] によりこの名前が付けられたが，現在では断熱計算のモデル [3, Def. 1] における条件を開放系，有限時間に緩め，ハミルトニアンをイジングモデルに制限したものと見なされている [34, § 3]．

[*4] 以下，qubits は搭載されている量子ビット数を表現するものとする．

ズの無い量子回路型コンピュータの開発目標が Google, IBM 等から公表されている．なお，量子コンピュータの性能を十分に引き出す強力なアルゴリズムを実現するためには量子ビット数のみではなく，量子誤り訂正，量子ランダムアクセスメモリ等，2022 年現在では実用化されていない技術を用いる必要があり，それらの開発スピードの予測困難性が，量子コンピュータが暗号に与える影響の将来予測を困難なものとしている．

量子コンピュータによる現代暗号への脅威: Peter Shor による素因数分解問題と離散対数問題に対する量子多項式時間アルゴリズム [41] が発表されて以降，数千 bits の RSA 暗号を危殆化させる量子コンピュータの規模，実現時期の見積りに関する研究が進められている [14, 16, 43, 4, 28, 30, 31]．特に，現在標準的に用いられている，2048bits の合成数を公開鍵とする RSA 暗号 RSA-2048 の危殆化時期に関して様々な予測が存在する．学術的な研究に基づいたものでは 2039 年以降 [43]，2050 年前後 [4] と少なくとも 20 年程度は実現に時間がかかるとされている．量子コンピューティングの専門家へのアンケートを元にした 2021 年時点での予測 [31] では 24 時間で RSA-2048 を解読可能な量子コンピュータが 15 年以内に出現する可能性が 50% 程度であると考える専門家が半数程度存在する．文部科学省 科学技術・学術政策研究所（NISTEP）による科学技術予測調査 [33, p. (II-4)48,52] ではある程度コヒーレンス時間の長い数百 qubits 規模の量子回路コンピュータの登場は 2033 年頃としているため，現代暗号に対して脅威となる量子コンピュータが出現するのはそれ以降と解釈できる．一方で，セキュリティ分野の専門家の予測では，PQCrypto2014 の招待講演における Mariantoni の予測では 2029 年 [28]，Workshop on Cybersecurity in a Post-Quantum World (2015 年) における Mosca の予測では 2026-2031 年 [30] と若干早めの時期を想定している．

将来的に RSA 暗号が危殆化すると考える専門家が多数存在する一方で，量子コンピュータ実機を用いた素因数分解問題及び離散対数計算の実験は小規模なものに留まっている．量子回路型コンピュータ実機を用いた Shor のアルゴリズムの実験に関しては，CRYPTREC 外部調査報告書「Shor のアルゴリズム実装動向調査」[46] に挙げられているもの及びその後の [25, 44, 49] を含めて 15，21，35 の素因数分解実験および離散対数問題 $2^z \equiv 1 \pmod 3$ の離散対数の計算実験を行ったもののみである．Shor のアルゴリズムを用いた初期の報告は $N = 15$ の素因数分解回路の量子フーリエ変換部分を除いた部分回路を実装する予備実験的なもの，位数や N の情報を用いて過度な簡略化を行ったものが多かった．しかし，近年では IBM Quantum を用いてほぼ完全な回路を実装した実験報告 [5] や離散対数問題の実装実験報告 [4] が出版されるなど，実際に素因数分解できた合成数の大きさには表れない量子回路規模の拡大は着実に続いている．

Shor のアルゴリズムを用いない素因数分解の計算手法として，2 進数乗算の筆算形式で式展開したものを，組み合わせ最適化問題として定式化するものがある．特に，量子アニーリングを用いた実験がこの 10 年間で多数報告されている．初期にはハミルトニアンに合わせて有機化合物を合成し，最適化問題の変数に対応する原子のスピンを核磁気共鳴 (Nuclear Magnetic Resonance: NMR) を用いた分析により結果を取り出すという手法で計算を行っていたためスケーリングが困難であったが，D-Wave 社の量子アニーリングマシンがオンライン上で比較的手軽に利用可能になって以降は実験報告が相次いでいる [51]．素因数分解のターゲットとなる数は着実に大型化しており，実機を用いた最も大きな実験 [23] では 19bits の合成数 376289=571×659 の分解に成功しているが，その要因は主に最適化問題を表現する際の方程式における変数の省略によるものである．また，同様の組み合わせ最適化問題を量子回路型コンピュータの上で Quantum Approximate Optimization Algorithm (QAOA) を用いて解く実験 [35](143，291311 を分解)，Digitized adiabatic quantum computation を用いて解く実験 [19](2479 を分解) も報告されている．量子回路型コンピュータ上で QAOA を用いた素因数分解問題へのアプローチとして，Schnorr アルゴリズム [39] の部分的な量子化の研究が存在する．Schnorr アルゴリズムは数体篩法の関係探索を係数制限付きの近似最近ベクトル問題に変換して行うが，[52] ではこれをさらに最適化問題に落とし込み，QAOA を 10qubits 回路上で実行することで 48bits の素因数分解実験を行ったという報告がされている．

PQC の必要性について: 上述のように，現代暗号に対する量子コンピュータの直接的な脅威は現時点では生じていない．しかし，民間企業のロードマップが予定通りに達成された場合には，今後数十年で現代暗号の解読を行うのに十分な大きさの量子計算を実行可能な量子コンピュータが開発される可能性がある．そのような量子コンピュータが出現した場合，守秘・鍵共有に用いられる現代の暗号方式の中で素因数分解問題や離散対数問題の計算困難性に基づいた RSA 暗号・楕円曲線暗号が危殆化するリスクがある．暗号方式の提案から社会的な普及までは RSA 暗号・楕円曲線暗号で 20 年ほどの期間が必要とされたことから，PQC の場合でも同程度の期間が必要と想定されるため，長期間の移行スケジュールを策定し，準備を行う必要がある．

1.2 PQC の研究及び標準化等に関する動向

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

PQC の標準化に関する近年の動向については，まず 2015 年 8 月，アメリカ国家安全保障局 (NSA) は PQC への将来的な移行計画を発表している．また，アメリカ国立標準技術研究所 (NIST) は 2016 年から PQC の公募を開始し，その締切である 2017 年 11 月 30 日までに 82 件の暗号方式が提案され，そのうち公募条件を満たした暗号方式は 69 件あり，その後 5 件の取り下げがあった．2019 年 1 月 30 日には，NIST から PQC の標準化の第 2 ラウンドへ進む方式として 26 件が発表された．2020 年 7 月 22 日には，NIST から PQC の標準化の第 3 ラウンドへ進む方式として，Finalists の 7 件，Alternate Candidates の 8 件が発表された．そして，2022 年 7 月 5 日には，NIST から標準化方式として公開鍵暗号方式 1 件と電子署名方式 3 件が発表された．同時に，第 4 ラウンドへ進む方式として，公開鍵暗号方式の 4 件が発表され，電子署名方式については再公募を行うこととした．欧州では ETSI が PQC の調査活動を行い [12]，ISO/IEC でも標準化に向けた議論が始まっている [21].

NIST の標準化において，安全性をレベル 1 から 5 で定義しており，各暗号方式は提案パラメータとそれによって達成される安全性レベルを示す必要があった．レベル 1, 3, 5 はそれぞれ AES128, AES192, AES256 などの 128, 192, 256bits の秘密鍵を持つブロック暗号の秘密鍵探索と同等かそれ以上の計算量であり，レベル 2 と 4 はそれぞれ SHA256/SHA3-256 と SHA384/SHA3-384 などの 256bits と 384bits の暗号学的ハッシュ関数の衝突探索と同等かそれ以上の計算量とされている．ここで，公開鍵暗号では，適応的選択暗号文攻撃に対する識別不可能性（IND-CCA2 安全性）を考える際には 2^{64} 個以下の選択暗号文を復号オラクルに古典的にクエリできるとし，電子署名では，適応的選択文書攻撃に対する存在的偽造困難性（EUF-CMA 安全性）を考える際には 2^{64} 個以下のメッセージを署名オラクルに古典的にクエリできるとしている．計算時間を制限するために，量子コンピュータを利用可能な攻撃者に対しては量子回路の最大の深さによって，古典コンピュータを利用可能な攻撃者に対しては古典論理ゲート数によってレベル 1 から 5 の安全性における計算量を評価しており，それぞれ表 1.1 のようになると見積もっている [32].

CRYPTREC の暗号技術調査ワーキンググループにおいても 2014 年度に PQC の代表的な候補である格子に基づく暗号技術について調査を行い，報告書「格子問題等の困難性に関する調査」を公開している [9]．さらに 2017 年度から 2018 年度にかけて，PQC の代表的な候補である 4 種類の分類（格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術，同種写像に基づく暗号技術）について調査し，報告書にまとめた [10].

表 1.1: NIST 耐量子計算機暗号 Call for proposal[32] における安全性レベルと計算量の対応表

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

1.3 本調査で対象とした PQC の種類

この節では本調査の対象として格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術, 同種写像に基づく暗号技術，ハッシュ関数に基づく署名技術を選択した理由及びその説明に必要な事項について述べる．

代表的な公開鍵暗号方式は，その安全性が数学的な計算問題の困難性と関わりがある．例えば RSA 暗号では，二つの同程度の大きさでかつ異なる素数 p, q が秘密鍵，それらの積 $N = pq$ が公開鍵として使用される．N が素因数分解されると秘密鍵 p, q が計算されてしまい，RSA 暗号は解読されてしまう．楕円曲線暗号の場合も楕円曲線暗号の公開鍵から楕円曲線上の離散対数問題が定義され，それを解くことでその秘密鍵が計算できてしまう．本ガイドライン・報告書で扱う代表的な 5 種類の PQC (格子に基づく暗号技術，符号に基づく暗号技術，多変数多項式に基づく暗号技術, 同種写像に基づく暗号技術，ハッシュ関数に基づく署名技術) も RSA 暗号と同様に，それらの安全性はそれぞれで利用される数学的な計算問題の困難性と関わりがある．そして，これらの問題を量子コンピュータを利用して効率よく解くアルゴリズムはまだ発見されていないことが，それら 5 種類の暗号方式が PQC とされている理由である．（本調査の対象である暗号方式と数学的な計算問題の関係は各章の第 1 節で説明する．）

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

1.4　耐量子計算機暗号調査報告書執筆者リスト

主査	國廣 昇	筑波大学
委員	青木 和麻呂	文教大学
委員	伊藤 忠彦	セコム株式会社
委員	草川 恵太	日本電信電話株式会社
委員	下山 武司	国立情報学研究所
委員	高木 剛	東京大学
委員	高島 克幸	早稲田大学
委員	廣瀬 勝一	福井大学
委員	安田 貴徳	岡山理科大学
委員	安田 雅哉	立教大学
事務局	野島 良	国立研究開発法人情報通信研究機構
事務局	青野 良範	国立研究開発法人情報通信研究機構
事務局	五十部 孝典	国立研究開発法人情報通信研究機構
事務局	伊藤 竜馬	国立研究開発法人情報通信研究機構
事務局	大久保 美也子	国立研究開発法人情報通信研究機構
事務局	大東 俊博	国立研究開発法人情報通信研究機構
事務局	小川 一人	国立研究開発法人情報通信研究機構
事務局	金森 祥子	国立研究開発法人情報通信研究機構
事務局	黒川 貴司	国立研究開発法人情報通信研究機構
事務局	高安 敦	国立研究開発法人情報通信研究機構
事務局	横山 和弘	国立研究開発法人情報通信研究機構
事務局	吉田 真紀	国立研究開発法人情報通信研究機構
事務局	篠原 直行	国立研究開発法人情報通信研究機構

第 1 章の参照文献

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. Bardin, R. Barends, R. Biswas, S. Boixo et al. Quantum supremacy using a programmable superconducting processor. *Nature*, volume 574, number 7779, pp. 505–510, Springer Science and Business Media, 2019.

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

[3] T. Albash, D. A. Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, volume 90, issue 1, 015002, American Physical Society, 2018.

[4] Y. Aono, S. Liu, T. Tanaka, S. Uno, R. Van Meter, N. Shinohara, R. Nojima. The present and future of discrete logarithm problems on noisy quantum computers. *IEEE Transactions on Quantum Engineering*, volume 3, pp. 1–21, IEEE, 2022.

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

[6] J. Buchmann, E. Dahmen, M. Szydlo. Hash-based digital signature schemes. *Post-Quantum Cryptography*, pp. 35–93, Springer, 2009.

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

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

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

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

[11] CRYPTREC 暗号技術調査ワーキンググループ（耐量子計算機暗号）．暗号技術ガイドライン（耐量子計算機暗号）．CRYPTREC GL-2004-2022, https://www.cryptrec.go.jp/report/cryptrec-gl-2004-2022.pdf, 2023.

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

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

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

[15] L. K. Grover. A fast quantum mechanical algorithm for database search. *STOC '96*, pp. 212–219, ACM, 1996.

[16] É. Gouzien, N. Sangouard. Factoring 2048-bit RSA integers in 177 days with 13 436 qubits and a multimode memory. *Physical Review Letters*, volume 127, 140503, American Physical Society, 2021.

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

[18] A. Hülsing, D. Butin, S. Gazdag, J. Rijneveld, A. Mohaisen. XMSS: eXtended Merkle Signature Scheme. *IRTF RFC* 8391, https://www.rfc-editor.org/rfc/rfc8391, 2018-05. (2023-04-11 閲覧)

[19] N. N. Hegade, K. Paul, F. Albarrán-Arriagada, X. Chen, E. Solano. Digitized adiabatic quantum factorization. *Physical Review A*, volume 104, l050403, American Physical Society, 2021.

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

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

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

[23] S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble, S. Kais. Quantum annealing for prime factorization. *Scientific Reports*, volume 8, article number 17667, 2018.

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

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

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

[27] N. Kunihiro. Quantum factoring algorithm: Resource estimation and survey of experiments. *International Symposium on Mathematics, Quantum Theory, and Cryptography*, volume 33 of Mathematics for Industry, pp. 39–55, Springer, 2021.

[28] M. Mariantoni, Building a superconducting quantum computer. *PQCrypto 2014*, Invited talk, https://www.youtube.com/watch?v=wWHAs--HA1c, 2014-10-01. (2023-04-11 閲覧)

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

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

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

[32] NIST. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf, 2016.

[33] 文部科学省 科学技術・学術政策研究所科学技術予測センター. 第 11 回科学技術予測調査 デルファイ調査. https://nistep.repo.nii.ac.jp/?action=repository_uri&item_id=6692&file_id=13&file_no=3, 2020-06.（2023-04-10 閲覧）

[34] 大関 真之. 量子アニーリングが拓く機械学習と計算技術の新時代. **数理解析研究所講究録**, volume 2059（量子システム推定の数理）, pp. 13–23, 2017.

[35] L. Qiu, M. Alam, A. Ash-Saki, S. Ghosh. Resiliency analysis and improvement of variational quantum factoring in superconducting qubit. *ISLPED'20*, pp. 229–234, ACM, 2020.

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

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

[38] 佐藤 信太郎. 量子コンピューティング：現状と産業化への課題 (量子技術の実用化推進 WG 第 3 回資料）. https://www8.cao.go.jp/cstp/ryoshigijutsu/jitsuyo_wg/3kai/siryo2-3.pdf, 2022-12-06.（2023-04-12 閲覧）

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

[40] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *SFCS'94*, pp. 124–134, ACM, 1994.

[41] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, volume 26, number 5, pp. 1484–1509, SIAM, 1997.

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

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

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

[45] 鈴木 教洋. 日立の量子コンピュータ研究開発戦略. https://www8.cao.go.jp/cstp/ryoshigijutsu/jitsuyo_wg/3kai/siryo2-2.pdf, 2022-12-06.（2023-04-10 閲覧）

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

[47] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, volume 414, pp. 883–887, Nature Publishing, 2001.

[48] D.-S. Wang. A comparative study of universal quantum computing models: Toward a physical unification. *Quantum Engineering*, volume 3, issue 4, article e85, Wiley, 2021.

[49] W. Wang, Z. You, S. Wang, Z. Tang, H. Ian. Computing Shor's algorithmic steps with classical light beams.

Scientific Reports, volume 12, article number 16599, Nature Publishing, 2022.

[50] 山本 俊. 量子情報技術（令和 3 年度　科学技術に関する調査プロジェクト）. **国立国会図書館調査及び立法考査局**, `https://www.ndl.go.jp/jp/diet/publication/document/2022/index.html`. （2023-04-11 閲覧）

[51] 山口 純平, 伊豆 哲也. イジング計算を用いた暗号解析について. **オペレーションズ・リサーチ**. 第 67 巻 6 号, pp. 290–296, 日本オペレーションズ・リサーチ学会, 2022.

[52] B. Yan, Z. Tan, S. Wei, H. Jiang, W. Wang, H. Wang, L. Luo, Q. Duan et al. Factoring integers with sublinear resources on a superconducting quantum processor. *arXiv*: 2212.12372, 2022.

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

第 2 章

格子に基づく暗号技術

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

2.1 格子に基づく暗号技術の安全性の根拠となる問題

2.1.1 LWE 問題と代表的な求解法

本節では, 2005 年 Regev が提案した LWE 問題 [89] を紹介すると共に, 格子を利用した LWE 問題に対する求解法を紹介する. また, LWE 問題のいくつかの変種についても言及する.

2.1.1.1 LWE 問題の紹介

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

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

が与えられたとする. (この数値例は [91] から引用した.) ただし, 各線形方程式の値は近似値であり, その誤差はこの例では ± 1 以内と仮定する. このとき, この連立線形近似方程式の解 $\mathbf{s}$ を求めるのが LWE 問題である. ここに示した数値例では $\mathbf{s} = (0, 13, 9, 11) \in \mathbb{Z}_{17}^4$ が解となる. LWE 問題で注意すべきことは, 連立線形近似方程式に誤差がない場合は, Gauss の消去法により効率的に解を求めることができる点である. 逆に言うと, 連立線形近似方程式で与えられる誤差の大きさが LWE 問題の求解を困難にする.

■ **離散 Gauss 分布**　一般に, LWE 問題における連立線形近似方程式の誤差は, 平均 0, パラメータ $\sigma > 0$ の $\mathbb{Z}$ 上の離散 Gauss 分布 $\chi = D_{\mathbb{Z},\sigma}$ から生成される[*1]. より正確には, χ は各整数 x がサンプルされる確率が $\exp\left(-\dfrac{\pi x^2}{\sigma^2}\right)$ に比

[*1] 本章では, 記号 σ をガウス分布のパラメータ (標準偏差とは異なる) の意味で使い, 署名を表すときには sig を用いる.

例する $\mathbb{Z}$ 上の離散確率分布である．この分布は, 数学的な正規分布[*2] とは異なるが, 絶対値の大きな値が生成される確率が非常に小さいという性質は共通している．例えば, 絶対値が 3σ より大きな整数がサンプルされる確率は非常に小さい．離散 Gauss 分布の詳細については [73] などを参照.

離散 Gauss 分布を厳密に実装するのは容易ではなく, timing attack などの脆弱性 [26] が生まれてしまう．現実の方式 (2.3 節参照) においては, 誤差 (ノイズ) として離散 Gauss 分布との統計距離が小さい分布を用いている．それらと区別するため, 方式 Scheme 内で用いられるノイズの分布を $D_{\mathbb{Z},s}^{\mathsf{Scheme}}$ と表現する．ここで, s はパラメータである, また, 記号 $D_{\mathbb{Z}^n,s}^{\mathsf{Scheme}}$, $D_{\mathbb{Z}^{n\times m},s}^{\mathsf{Scheme}}$ によってそれぞれ, 成分を $D_{\mathbb{Z},s}^{\mathsf{Scheme}}$ から独立に生成した n 次元ベクトル, $n\times m$ 行列とする.

■ **LWE 問題の定式化**　以下は，定式化された LWE 問題である：

定義 2.1 (LWE 問題 [89]) n を正の整数とし, q を奇素数とする．平均 0, パラメータ σ の $\mathbb{Z}$ 上の離散 Gauss 分布を $\chi = D_{\mathbb{Z},\sigma}$ とする．秘密ベクトル $\mathbf{s}\in\mathbb{Z}_q^n$ を固定する．一様ランダムに選ばれた $\mathbf{a}\in\mathbb{Z}_q^n$ と離散 Gauss 分布 χ からサンプルされた $e\in\mathbb{Z}$ に対して, $(\mathbf{a},b)\in\mathbb{Z}_q^n\times\mathbb{Z}_q$ の組を出力する確率分布を $L_{\mathbf{s},\chi}$ とする．ただし, $b\equiv\langle\mathbf{a},\mathbf{s}\rangle+e \pmod q$ とする．(2 つのベクトル $\mathbf{v}$ と $\mathbf{w}$ の内積を $\langle\mathbf{v},\mathbf{w}\rangle$ で表す．) このとき, 次の 2 つの問題を考える：

1. **判定** LWE (Decision-LWE)　与えられた組 $(\mathbf{a},b)\in\mathbb{Z}_q^n\times\mathbb{Z}_q$ が, 確率分布 $L_{\mathbf{s},\chi}$ からサンプルされた元か, $\mathbb{Z}_q^n\times\mathbb{Z}_q$ 上一様ランダムに生成された元かを決定する問題.
2. **探索** LWE (Search-LWE)　確率分布 $L_{\mathbf{s},\chi}$ からサンプルされた組 $(\mathbf{a},b)$ から秘密ベクトル $\mathbf{s}$ を復元する問題.

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

$$\begin{cases}(\mathbf{a}_1,b_1), & b_1\equiv\langle\mathbf{a}_1,\mathbf{s}\rangle+e_1 \pmod q\\ (\mathbf{a}_2,b_2), & b_2\equiv\langle\mathbf{a}_2,\mathbf{s}\rangle+e_2 \pmod q\\ \quad\vdots & \\ (\mathbf{a}_m,b_m), & b_m\equiv\langle\mathbf{a}_m,\mathbf{s}\rangle+e_m \pmod q\end{cases}$$

から LWE 問題を解くことを考える．(解読に要する計算時間が最も短くなるような m を攻撃者が選べることを想定する．) 第 i 行ベクトルを $\mathbf{a}_i$ とする $m\times n$ 行列を $\mathbf{A}$ とし, $\mathbf{b}=(b_1,b_2,\ldots,b_m)$ とおく．このとき, ここに示した m 個の LWE サンプルの組は $(\mathbf{A},\mathbf{b})\in\mathbb{Z}_q^{m\times n}\times\mathbb{Z}_q^n$ と簡潔に表せて, 関係式 $\mathbf{b}\equiv\mathbf{s}\mathbf{A}^\top+\mathbf{e} \pmod q$ を満たす．ただし, $\mathbf{e}=(e_1,e_2,\ldots,e_m)\in\mathbb{Z}^m$ をノイズベクトルとする．(各 e_i は χ からサンプルされた元であることに注意する．)

■ **LWE 問題の変種**　LWE 問題の変種として，多項式環 $R_q=\mathbb{Z}_q[x]/(\phi)$ 上の LWE である Ring-LWE [99, 74][*3] や Module-LWE [76] がある．Ring-LWE では, 3 つの多項式 $s,a_i,e_i\in R_q$ に対する Ring-LWE サンプルとして $\{(a_i,a_i\cdot s+e_i)\}_{i=1}^m$ を考える．(特に, 通常の LWE 問題と同じように, ランダムな s と, 係数が小さい多項式の集合からサンプリングされた e_i が用いられる．) Ring-LWE の基礎環 R_q を定める多項式として，2 のべき乗の形をした整数 n に対し $\phi=x^n+1$ がよく用いられる．また, Module-LWE では, 多項式ベクトル $\mathbf{s},\mathbf{a}_i\in R_q^k$ と多項式 $e_i\in R_q$ に対する Module-LWE サンプルとして $\{(\mathbf{a}_i,\langle\mathbf{a}_i,\mathbf{s}\rangle+e_i)\}_{i=1}^m$ を考える．Module-LWE の基礎環 R_q を定める多項式としては $\phi=x^{n/k}+1$ がよく用いられる．さらに, 環上の LWE 以外の LWE 問題の変種として, 丸め込み（rounding）

[*2] 分散 t^2 に対して数学的な正規分布 $N(0,t^2)$ は, 確率密度関数が $\frac{1}{\sqrt{2\pi}t}e^{-z^2/(2t^2)}$ により定義されるため, $\sqrt{2\pi}$ 倍のずれがある．暗号の安全性を議論する際に格子上のフーリエ変換が用いられることが多く [89], 本文中の定義を用いることで, 数式の表現が簡潔となる.

[*3] 文献 [74] ではより一般的に整数環とイデアルを用いて定義されているが, 後の文献 [32] ではそれの簡略化として, 多項式環 R_q を用いた表現である “polynomial-LWE assumption” が提案された．2019 年現在では後者の表現の方が Ring-LWE と呼ばれている.

でノイズベクトルを生成する LWR[30] や middle-product と呼ばれる多項式演算を用いる Middle-product LWE [92] など数多くの変種が提案されている.

2.1.1.2 格子の基本事項と q-ary 格子の紹介

■ **格子の基本事項** m 次元実ベクトル空間 $\mathbb{R}^m$ の一次独立な m 個のベクトル $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$ の整数係数の線形結合全体 $L = \{\sum_{i=1}^{m} x_i \mathbf{b}_i : x_i \in \mathbb{Z}, 1 \leq i \leq m\}$ を (完全階数の) m 次元**格子**と呼ぶ. 特に, 格子 L はベクトル空間 $\mathbb{R}^m$ の (離散) 加法部分群である. また, 格子 L を生成する一次独立な m 個のベクトルの組 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ を**基底**と呼び, 各 $\mathbf{b}_i$ を**基底ベクトル**と呼ぶ. さらに, 行ベクトルで表した基底ベクトル $\mathbf{b}_i \in \mathbb{R}^m$ を行として持つ $m \times m$ 行列 $\mathbf{B} = (\mathbf{b}_i)_{i=1}^{m}$ を格子 L の**基底行列**と呼ぶ. 2 次元以上の格子を生成する異なる基底は無限に存在し, 同じ格子を生成する 2 つの基底行列 $\mathbf{B}_1$ と $\mathbf{B}_2$ に対し $\mathbf{B}_2 = \mathbf{V}\mathbf{B}_1$ を満たす $m \times m$ のユニモジュラ行列 $\mathbf{V}$ が存在する. また, 基底行列 $\mathbf{B}$ を用いて, 格子 L の**体積**を $\mathrm{vol}(L) = |\det(\mathbf{B})|$ と定める. (体積は基底の取り方に依存しない.) 格子 L の**第 1 逐次最小**は L 上の最短な非零ベクトルの Euclid ノルムを指し, $\lambda_1(L)$ と表す. ベクトル空間 $\mathbb{R}^m$ の完全階数の格子 L に対し, 集合 $\widehat{L} = \{\mathbf{x} \in \mathbb{R}^m : \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z} \quad (\forall \mathbf{y} \in L)\}$ を格子 L の**双対格子**と呼ぶ. また, 格子 L の基底行列 $\mathbf{B}$ に対して, $\widehat{\mathbf{B}} = \left(\mathbf{B}^{-1}\right)^{\top}$ は双対格子 $\widehat{L}$ の基底行列となり, この $\widehat{\mathbf{B}}$ を**双対基底行列**と呼ぶ. 単位行列 $\mathbf{I}_m$ に対し $\mathbf{B}\widehat{\mathbf{B}}^{\top} = \mathbf{I}_m$ を満たすので, $\mathrm{vol}(L) \times \mathrm{vol}(\widehat{L}) = 1$ が成り立つ.

■ q-ary **格子** ここでは，LWE 問題の求解で利用する特殊な格子を紹介する. 正の整数 q に対して, $q\mathbb{Z}^m \subseteq L \subseteq \mathbb{Z}^m$ を満たす完全階数の m 次元格子 L を **q-ary 格子**と呼ぶ. 2 つの自然数 $m > n$ に対し, 任意の正の整数 q と $n \times m$ 整数行列 $\mathbf{X}$ に対する 2 つの m 次元 q-ary 格子を

$$\Lambda_q(\mathbf{X}) = \{\mathbf{y} \in \mathbb{Z}^m : \exists \mathbf{s} \in \mathbb{Z}^n \text{ s.t. } \mathbf{y} \equiv \mathbf{s}\mathbf{X} \pmod{q}\}, \quad \Lambda_q^{\perp}(\mathbf{X}) = \left\{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y}\mathbf{X}^{\top} \equiv \mathbf{0} \pmod{q}\right\}$$

と定義する. (これらの集合は $\mathbb{R}^m$ の離散加法部分群なので格子である.) 正規化の差を除き, これら 2 つの q-ary 格子は互いに双対の関係にある. 正確には $\Lambda_q^{\perp}(\mathbf{X}) = q\widehat{\Lambda_q(\mathbf{X})}$ と $\Lambda_q(\mathbf{X}) = q\widehat{\Lambda_q^{\perp}(\mathbf{X})}$ が成り立つ. また, 群準同型写像 $f : \mathbb{Z}^m \longrightarrow (\mathbb{Z}/q\mathbb{Z})^n, \quad \mathbf{y} \longmapsto \mathbf{y}\mathbf{X}^{\top} \pmod{q}$ の核は q-ary 格子 $\Lambda_q^{\perp}(\mathbf{X})$ なので, 群の準同型定理から $\mathrm{vol}(\Lambda_q^{\perp}(\mathbf{X})) = [\mathbb{Z}^m : \Lambda_q^{\perp}(\mathbf{X})] = \#\mathrm{Im}(f)$ が成り立つ. (群の指数 $[\mathbb{Z}^m : \Lambda_q^{\perp}(\mathbf{X})]$ は格子の体積の比 $\dfrac{\mathrm{vol}(\Lambda_q^{\perp}(\mathbf{X}))}{\mathrm{vol}(\mathbb{Z}^m)}$ に一致することに注意する.) これより, 体積 $\mathrm{vol}\left(\Lambda_q^{\perp}(\mathbf{X})\right)$ は q^n を割る. さらに, 元の格子と双対格子の体積の関係から, q^{m-n} は体積 $\mathrm{vol}\left(\Lambda_q(\mathbf{X})\right)$ を割ることが分かる. (ただし, ほとんどの行列 $\mathbf{X}$ に対して写像 f は全射で, その時 $\mathrm{vol}\left(\Lambda_q^{\perp}(\mathbf{X})\right) = q^n$ と $\mathrm{vol}\left(\Lambda_q(\mathbf{X})\right) = q^{m-n}$ が成り立つ.) q-ary 格子 $\Lambda_q(\mathbf{X})$ 上のベクトルは $\mathbf{y} = \mathbf{s}\mathbf{X} + q\mathbf{z}$ $(\mathbf{s} \in \mathbb{Z}^n, \mathbf{z} \in \mathbb{Z}^m)$ とかけるので, その格子は $(n+m) \times m$ 整数行列 $\begin{pmatrix} \mathbf{X} \\ q\mathbf{I}_m \end{pmatrix}$ の一次従属な $(n+m)$ 個の行ベクトルで生成される. この生成行列の Hermite Normal Form を計算することで, m 次元 q-ary 格子 $\Lambda_q(\mathbf{X})$ の基底行列 $\mathbf{B} \in \mathbb{Z}^{m \times m}$ が得られる. また, 双対基底の性質から, もう片方の q-ary 格子 $\Lambda_q^{\perp}(\mathbf{X})$ の基底行列は $(q\mathbf{B}^{-1})^{\top} \in \mathbb{Z}^{m \times m}$ で得られる.

2.1.1.3 LWE 問題の代表的な求解法

格子上の計算問題である格子問題として，最短ベクトル問題（Shortest Vector Problem, SVP）や最近ベクトル問題（Closest Vector Problem, CVP）などが代表的である. ここでは，LWE 問題の格子問題への帰着を述べる.

■ **判定 LWE 問題に対する求解** 判定 LWE 問題を SIS (Short Integer Solution) 問題に帰着して解く方法を紹介する：正の整数 q と, $0 < \beta < q$ を満たす実数 β を固定する. 各成分が剰余環 $\mathbb{Z}/q\mathbb{Z}$ 上一様ランダムに選ばれた $n \times m$ 整数行列 $\mathbf{X}$ に対して, $\|\mathbf{v}\| \leq \beta$ かつ $\mathbf{v}\mathbf{X}^{\top} \equiv \mathbf{0} \pmod{q}$ を満たす非零ベクトル $\mathbf{v} \in \mathbb{Z}^m$ を見つける問題を **SIS 問題**と呼

ぶ. つまり, これは q-ary 格子 $\Lambda_q^\perp(\mathbf{X})$ 上の短い非零ベクトルを見つける問題である. 剰余パラメータ q における LWE 問題のサンプル数を m とし, m 個の LWE サンプルの組を $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ とする. ここで, $n \times m$ の転置行列 $\mathbf{A}^\top$ に対する SIS 問題の短い解ベクトル $\mathbf{v} \in \Lambda_q^\perp(\mathbf{A}^\top)$ が得られたとする ($0 < \|\mathbf{v}\| \le \beta$ と仮定). このとき, LWE サンプルの組 $(\mathbf{A}, \mathbf{b})$ は関係式 $\mathbf{b} \equiv \mathbf{s}\mathbf{A}^\top + \mathbf{e} \pmod q$ を満たすので, $\langle \mathbf{v}, \mathbf{b}\rangle \equiv \langle \mathbf{v}, \mathbf{s}\mathbf{A}^\top + \mathbf{e}\rangle \equiv \langle \mathbf{v}\mathbf{A}, \mathbf{s}\rangle + \langle \mathbf{v}, \mathbf{e}\rangle \equiv \langle \mathbf{v}, \mathbf{e}\rangle \pmod q$ が成り立つ. ($\mathbf{v}\mathbf{A} \equiv \mathbf{0} \pmod q$ に注意) さらに, ノイズベクトル $\mathbf{e}$ のすべての成分 e_i は離散 Gauss 分布 χ からサンプルされた元なので, $|\langle \mathbf{v}, \mathbf{e}\rangle| \lesssim \sigma\sqrt{m}\|\mathbf{v}\| \le \sigma\beta\sqrt{m}$ が期待できる. (離散 Gauss 分布 $\chi = D_{\mathbb{Z},\sigma}$ のサンプル元 e_i の絶対値はおおよそ σ 未満で, 多めに見積もって $\|\mathbf{e}\| \lesssim \sigma\sqrt{m}$ とした.) ゆえに, $\sigma\beta\sqrt{m} \ll q$ ならば, $|\langle \mathbf{v}, \mathbf{b}\rangle| \pmod q$ の値の大きさから LWE サンプルの組 $(\mathbf{A}, \mathbf{b})$ は確率分布 $L_{\mathbf{s},\chi}$ からサンプルされたものか判定できる.

■ **探索 LWE 問題に対する求解法** 探索 LWE 問題を BDD (Bounded Distance Decoding) 問題に帰着して解く方法を紹介する：格子 L と目標ベクトル $\mathbf{w}$ に対し, ある $0 < \mu \le \frac{1}{2}$ が存在し $\mathrm{dist}(\mathbf{w}, L) = \min_{\mathbf{v}\in L}\|\mathbf{w} - \mathbf{v}\| < \mu\lambda_1(L)$ を満たすと仮定する. 格子 L の基底が与えられたとき, 目標ベクトル $\mathbf{w}$ に最も近い格子ベクトル $\mathbf{v} \in L$ を見つける問題を **BDD 問題**と呼ぶ. m 個の LWE サンプルの組 $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ は関係式 $\mathbf{b} \equiv \mathbf{s}\mathbf{A}^\top + \mathbf{e} \pmod q$ を満たすので, 探索 LWE 問題は $\mathbf{b}$ を目標ベクトルとする q-ary 格子 $\Lambda_q(\mathbf{A}^\top)$ 上の BDD 問題とみなせる. 実際, 目標ベクトル $\mathbf{b} = \mathbf{s}\mathbf{A}^\top + \mathbf{e} + q\mathbf{z}$ ($\exists \mathbf{z} \in \mathbb{Z}^m$) に対して, 格子ベクトルを $\mathbf{v} = \mathbf{s}\mathbf{A}^\top + q\mathbf{z} \in \Lambda_q(\mathbf{A}^\top)$ とおくと, $\mathbf{b} - \mathbf{v} = \mathbf{e}$ が成り立つ. ノイズベクトル $\mathbf{e}$ のすべての成分 e_i は離散 Gauss 分布 χ からサンプルされた元であるため, 分散と次元が大きい場合にはおおよそスケーリングされたカイ二乗分布に従い, 高い確率で $\|\mathbf{e}\| \approx \frac{\sigma}{\sqrt{2\pi}} \cdot \sqrt{m}$ となる. ゆえに, 目標ベクトル $\mathbf{b}$ との距離が $\sigma\sqrt{m}$ 以下となる q-ary 格子 $\Lambda_q(\mathbf{A}^\top)$ 上の格子ベクトル $\mathbf{v}$ を見つけることで, ノイズベクトル $\mathbf{e}$ を復元することができる. 実用的には，Kannan や Bai-Galbraith らの埋め込み法 [68, 25] により，BDD 問題を unique-SVP 問題に帰着してから，ノイズベクトル $\mathbf{e}$ を復元する.

注意 2.2 (LWE 問題の変種に対する求解) LWE 問題の代表的な変種である Ring-LWE や Module-LWE では，上述したように多項式環 $R_q = \mathbb{Z}_q[x]/(\phi)$ を基礎環として利用する. n 次多項式 ϕ に対して，基礎環 $R_q = \mathbb{Z}_q[x]/(\phi)$ の任意の元は $n-1$ 次以下の多項式 $f = f_0 + f_1 x + \cdots + f_{n-1}x^{n-1}$ ($f_i \in \mathbb{Z}_q$) と表せ，その係数ベクトル $\mathbf{f} = (f_0, f_1, \ldots, f_{n-1}) \in \mathbb{Z}_q^n$ と一対一に対応する. このように，基礎環 R_q の元をその係数ベクトルに対応させることで，Ring-LWE や Module-LWE 問題は通常の LWE 問題と同じようにベクトル・行列の形で表現できる. （詳細は [15] を参照. また，ベクトル・行列の形の表現については，次で説明する NTRU 問題も参照. ）ベクトル・行列の形で表現した Ring-LWE や Module-LWE 問題に対して，上述で説明した通常の LWE 問題の求解法が適用できる.

2.1.2 NTRU 問題と代表的な求解法

ここでは，NTRU 問題とその代表的な求解法を紹介する. まず以下で, NTRU 問題について述べる：

定義 2.3 (NTRU 問題 [65]) 2 つの正の整数 n と q に対し, $\phi \in \mathbb{Z}[x]$ を次数 n の多項式とし, $R_q = \mathbb{Z}_q[x]/(\phi)$ とする. 係数が小さい 2 つの多項式 $f \in R_q^\times, g \in R_q$ に対して, $h = g \cdot f^{-1} \in R_q$ とする. (特に，f は環 R_q の可逆元に注意) このとき, 与えられた多項式 h から, f または g の多項式を復元する問題を (探索) NTRU 問題という.

NTRU 問題における多項式 ϕ の選び方として, $\phi = x^n \pm 1$, $x^n - x - 1$, $x^n - x^{n/2} + 1$, $\sum_{i=0}^{n-1} x^i$ などがある [12, Table 1]. (最後の ϕ のみ，次数は $n-1$ である.) また, 多項式 f (または g) の選び方として, $\{-1, 0, 1\}$ などの小さい係数を持つ多項式や, 小さい素数 p と係数が小さい多項式 F に対し $f = pF$ または $f = pF + 1$ と選ぶことが多い.

次に, NTRU 問題の代表的な求解法を紹介する．まず, 与えられた多項式 $h \in R_q$ に対して, h の回転行列を $\mathbf{H} \in \mathbb{Z}^{n\times n}$ とする. (具体的には，$n \times n$ 整数行列 $\mathbf{H}$ の i 行ベクトルを多項式 $x^{i-1}h \in R_q$ の係数ベクトルとする．) このとき

$$\mathbf{H}\begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix} = \begin{pmatrix} h \\ xh \\ \vdots \\ x^{n-1}h \end{pmatrix} \in R_q^n$$

が成り立つ．ここで, $2n \times 2n$ 行列 $\mathbf{B} = \begin{pmatrix} \mathbf{I}_n & \mathbf{H} \\ \mathbf{0} & q\mathbf{I}_n \end{pmatrix}$ の行ベクトルで生成される **NTRU 格子**を L とする．このとき, $2n$ 次元の NTRU 格子 L は短いベクトル $(\mathbf{f} \mid \mathbf{g}) \in \mathbb{Z}^{2n}$ を含む. (ただし, $\mathbf{f}, \mathbf{g} \in \mathbb{Z}^n$ を多項式 $f, g \in R_q$ の係数ベクトルとする．) 実際, $hf = g \pmod q$ より, $g = hf + qr$ を満たす多項式 $r \in R_q$ が存在する．また, 多項式 r の係数ベクトルを $\mathbf{r} \in \mathbb{Z}^n$ とすると,

$$\mathbf{g}\begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix} = g = hf + qr = \mathbf{f}\begin{pmatrix} h \\ xh \\ \vdots \\ x^{n-1}h \end{pmatrix} + q\mathbf{r}\begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix} = (\mathbf{fH} + q\mathbf{r})\begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix} \in R_q$$

となるので, $\mathbf{g} = \mathbf{fH} + q\mathbf{r}$ が成り立つ．これより, $(\mathbf{f} \mid \mathbf{g}) = (\mathbf{f} \mid \mathbf{fH} + q\mathbf{r}) = (\mathbf{f} \mid \mathbf{r})\mathbf{B} \in L$ が成り立つ．（つまり, ベクトル $(\mathbf{f} \mid \mathbf{g})$ が NTRU 格子 L に含まれる．）ベクトル $(\mathbf{f} \mid \mathbf{g}) \in \mathbb{Z}^{2n}$ が十分小さく NTRU 格子 L 上の最短ベクトルと仮定すると, これは NTRU 問題を SVP に帰着できることを示している．（最短ベクトル $(\mathbf{f} \mid \mathbf{g})$ の各ブロックにおける回転で得られるベクトルも NTRU 格子 L に含まれるので，一般に NTRU 格子 L は複数の最短ベクトルを含む．）

2.1.3 格子問題を解くアルゴリズムとその計算量について

SVP・CVP の格子問題やここまでに紹介した LWE 問題・NTRU 問題などの格子問題を解くのに有用な技術として**格子基底簡約**がある．格子基底簡約は, 与えられた格子 L の基底から, 各ベクトル $\mathbf{b}_i$ が短く・互いのベクトルが直交に近い格子 L の新しい基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ を見つける操作である. (明確な定義はないが, このような基底を「簡約基底」または「良い基底」と呼ぶ．)

2.1.3.1 代表的な格子基底簡約アルゴリズムの紹介

基底簡約アルゴリズムを紹介するために, Gram-Schmidt の直交化を説明する：基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ の Gram-Schmidt ベクトル $\mathbf{b}_i^*$ は次のように再帰的に定まる：$\mathbf{b}_1^* = \mathbf{b}_1$, $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*$, $\mu_{i,j} = \dfrac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\|\mathbf{b}_j^*\|^2}$．また, 各 $2 \le \ell \le m$ に対し $\mathbb{R}^m$ から $\mathbb{R}$-ベクトル空間 $\langle \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{\ell-1} \rangle_{\mathbb{R}}$ の直交補空間への直交射影を π_ℓ とかく. (便宜上，π_1 を恒等写像とする．) 以下で, 代表的な 2 つの格子基底簡約アルゴリズムを紹介する：

■ **LLL [71]** 簡約パラメータ $\dfrac{1}{4} < \delta < 1$ に対し, LLL 基底簡約は次の 2 条件を満たす基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ を見つける （次元 m に関する）多項式時間アルゴリズムである： (i) 基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ はサイズ簡約されている：Gram-Schmidt 係数が $|\mu_{i,j}| \le \dfrac{1}{2}$ $(\forall i > \forall j)$ を満たす． (ii) 基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ は Lovász 条件を満たす：$\delta\|\mathbf{b}_{k-1}^*\|^2 \le \|\pi_{k-1}(\mathbf{b}_k)\|^2$ $(2 \le \forall k \le m)$ を満たす．入力基底に対して, Lovász 条件が成り立たないとき LLL 基底簡約アルゴリズム内で隣り合う基底ベクトル $\mathbf{b}_{k-1}$ と $\mathbf{b}_k$ の交換を行い, (i) と (ii) の 2 条件を満たす基底を見つける．

■ BKZ [95]　BKZ 基底簡約アルゴリズムは，ブロックサイズ β による LLL 基底簡約アルゴリズムの一般化である．LLL に比べ, BKZ 基底簡約アルゴリズムでより良い簡約基底を見つけることが可能であるが, その計算量は β に関して指数時間である．特に, BKZ 基底簡約アルゴリズムに入力するブロックサイズ β を増やすごとに, 実行時間が非常に遅くなるが, より短い基底ベクトルを出力する．具体的には, ブロックサイズ $2 \leq \beta \leq m$ に対して, BKZ 基底簡約アルゴリズムは次の 2 つの条件を満たす格子 L の基底 $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$ を見つける： (i) 基底はサイズ簡約されている．(ii) すべての $1 \leq j \leq m$ に対し $\|\mathbf{b}_j^*\| = \lambda_1(L_{[j,k]})$ を満たす．ただし, $k = \min(j + \beta - 1, m)$ とし, 射影ベクトル $\pi_j(\mathbf{b}_j), \ldots, \pi_j(\mathbf{b}_k)$ で生成されるブロック射影格子を $L_{[j,k]}$ とする．入力基底に対して, BKZ 基底簡約アルゴリズム内ではブロック射影格子 $L_{[j:k]}$ 上の SVP オラクルを繰り返し呼びだし, (i) と (ii) の 2 条件を満たす基底を見つける．

2.1.3.2　BKZ 基底簡約アルゴリズムの出力基底と計算量

これまで BKZ2.0 [37] などの効率的な BKZ の改良アルゴリズムが提案され, 格子に基づく暗号技術の安全性評価で頻繁に利用されている．以下で，BKZ の出力基底と計算量評価の見積もりについて紹介する (詳細は [12] を参照)：

■ BKZ の出力基底の見積もり　格子基底簡約アルゴリズムが出力する簡約基底の「良さ」を測る指標として Hermite 因子がある．m 次元格子 L の基底が与えられたとき, アルゴリズムが出力する最短な基底ベクトルを $\mathbf{b} \in L$ とする．このとき, その基底簡約アルゴリズムの Hermite 因子は $\gamma = \dfrac{\|\mathbf{b}\|}{\mathrm{vol}(L)^{1/m}}$ で定義される．(つまり, Hermite 因子が小さいほど, より短い基底ベクトルの出力を意味する．) 100 以上の高次元のランダム格子に対し, LLL や BKZ などの基底簡約アルゴリズムの Hermite 因子の m 乗根 $\gamma^{1/m}$ は定数に収束することが実験的に知られている．高い次元 m のランダム格子において, ブロックサイズ $\beta \geq 50$ に対する BKZ 基底簡約アルゴリズムの root Hermite 因子はおおよそ

$$\gamma^{\frac{1}{m}} \approx \left(\nu_\beta^{-\frac{1}{\beta}}\right)^{\frac{1}{\beta-1}} \approx \left(\frac{\beta}{2\pi e}(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}}$$

に従うことが実験的に知られている [37, 104]．ただし, ν_β は β-次元の単位超球の体積とする．(例えば，$\beta = 85$ で $\gamma^{1/m} \approx 1.01$ となる．) この root Hermite 因子の見積もりを用いて, 格子に基づく暗号技術の安全性評価対象の格子問題の求解で必要となる BKZ のブロックサイズ β を求めることができる．

■ BKZ の計算量の見積もり　BKZ 基底簡約アルゴリズムの計算量は, β 次元格子上の「SVP オラクルの計算量」と「呼び出し回数」の積で見積もることができる．β 次元格子上の SVP オラクルに適したアルゴリズムとして篩 (sieving) と数え上げ (enumeration) があり, 篩の方が漸近計算量が小さい．(ただし, 数え上げの空間計算量が β に関して多項式的であるのに対し，篩の空間計算量は β に関して指数関数的である．) 具体的には，β 次元格子上の篩の時間計算量は $2^{c\beta+o(\beta)}$ で, 古典計算機上では $c = 0.292$ で, Grover アルゴリズムによって量子計算機上で $c = 0.265$ と見積もられている．一方, 数え上げの時間計算量は古典計算機上で $2^{c_1\beta\log\beta+c_2\beta+c_3}$ または $2^{c_1\beta^2+c_2\beta+c_3}$ で, Grover アルゴリズムにより量子計算機上ではその指数部分が半分になると見積もられている．(定数 c_1, c_2, c_3 に関しては様々な評価値があり, 具体的な値については [12, Table 4] を参照．) また, BKZ 内の SVP オラクルの呼び出し回数については, β または $8m$ と見積もることが多い．(β は BKZ のブロックサイズで, m は格子の次元とする．)

■ Core-SVP による安全性レベルの見積もり　上述の LWE や NTRU の探索問題の求解において，秘密情報に対応するベクトル $\mathbf{v}$ を帰着先の格子 L の最短ベクトルとして埋め込み，L の基底に BKZ 基底簡約アルゴリズムを適用することで目的の $\mathbf{v}$ を見つけることを考える．（具体的には，目的の $\mathbf{v}$ は，探索 LWE 問題では q-ary 格子上のノイズベクトル $\mathbf{e}$ で，NTRU 問題では NTRU 格子上のベクトル $(\mathbf{f} \mid \mathbf{g})$ を想定する．）ここで，$\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$ を格子 L の β-BKZ 簡約基底とし，$\{\mathbf{b}_1^*, \ldots, \mathbf{b}_m^*\}$ をその Gram-Schmidt ベクトルとする．Gaussian Heuristic と Geometric Series

Assumption（GSA）の仮定の下で，目的ベクトル $\mathbf{v}$ の $m-\beta$ の位置における射影ベクトル $\pi_{m-\beta}(\mathbf{v}) \in \pi_{m-\beta}(L)$ の長さが

$$\|\pi_{m-\beta}(\mathbf{v})\| < \|\mathbf{b}^*_{m-\beta}\| \approx \delta_\beta^{2\beta-m-1}\mathrm{vol}(L)^{\frac{1}{m}}, \quad \delta_\beta = \left(\frac{\beta}{2\pi e}(\pi\beta)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}}$$

を満たせば，BKZ 基底簡約の基底ベクトルとして目的の $\mathbf{v}$ を見つけることができる．（探索 LWE 問題に対する BKZ による求解実験については，[19, 87] を参照．）この不等式を満たす BKZ のブロックサイズ β に対して，BKZ 基底簡約アルゴリズムのサブルーチンである β-次元 SVP アルゴリズムの 1 回の計算困難性を Core-SVP 困難性と呼ぶ [17]. 格子に基づく暗号方式の具体的な安全性レベルは，Core-SVP 困難性で評価・比較されることが多い.

2.1.3.3 格子問題の公開チャレンジの求解状況

SVP や LWE に対する求解アルゴリズムをテストする目的で，ドイツ・ダルムシュタット大学によって「SVP チャレンジ」・「LWE チャレンジ」と呼ばれる求解コンテストがインターネット上で開催されている [101]. 2018 年に，篩（ふるい）をベースとした高速な格子アルゴリズム群である General Sieve Kernel (G6K)[16] が提案され，SVP チャレンジ・LWE チャレンジの求解記録が飛躍的に更新された. 具体的には，SVP チャレンジにおいては，G6K 内の篩アルゴリズムを GPU 実装することで，180 次元の SVP インスタンスが 4 NVIDIA Turing GPUs の計算機（1.5TB RAM）を用いて 51.6 日で求解されたと 2021 年 2 月に報告されている [47].（ただし，本報告では Gaussian Heuristic で期待される最短ベクトル長に対する近似因子が 1.04002 なので，今回見つかった格子ベクトルは 180 次元 SVP インスタンスの厳密解ではなく近似解である．）また，LWE チャレンジにおいては，$(n, \alpha) = (45, 0.030), (50, 0.025), (55, 0.020), (90, 0.005)$ の数多くの LWE インスタンスが G6K 内の progressive-BKZ の改良により求解されたと 2022 年 6, 7 月に報告されている．（ただし，n は LWE の秘密ベクトル長で，α はノイズの大きさに関するパラメータで，組 (n, α) のバランスで LWE インスタンスの難しさが大きく変化する．）例えば，$(n, \alpha) = (50, 0.025)$ の LWE インスタンスに対して，次のスペックを持つ計算システムで約 592 時間で求解されている：

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

2.2 格子に基づく代表的な暗号方式

本節では, 格子に基づく代表的な暗号方式として, LWE 問題に基づく Regev による暗号化方式 [89] および Lindner, Peikert らによる暗号化方式 [73], Ring-LWE 問題に基づく Brakerski らによる暗号化方式 [32], NTRU 問題に基づく Hoffstein らによる暗号化方式 [65], Hash-and-Sign に基づく署名方式の格子問題への拡張, ならびに Fiat-Shamir 署名方式の格子問題への拡張について述べる.

2.2.1 LWE に基づく Regev による暗号化方式

Regev による暗号化方式 [89] の構成には, 以下の 4 つのパラメータが必要である.

- n: 安全性パラメータ
- m: LWE サンプルの個数 ($m \geq 1.1 \cdot n\log q$ となる最小の整数を選ぶ)

- q: 剰余パラメータ (q として $n^2 \leq q \leq 2n^2$ を満たす素数を選ぶ)
- $\alpha > 0$: ノイズパラメータ ($\alpha = 1/(\sqrt{n} \cdot \log^2 n)$)

以下に具体的な暗号方式の構成を示す.

秘密鍵の生成 一様ランダムに $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ を選ぶ

公開鍵の生成 秘密鍵 $\mathbf{s}$, 剰余パラメータ q ノイズパラメータ α を持つ LWE 分布から生成した m 個のサンプル $(\mathbf{a}_i, b_i)_{i=1}^m \leftarrow A_{\mathbf{s},\chi}^m$ を公開鍵とする. ただし各 i について, $a \leftarrow \mathbb{Z}_q^n$, $e_i \leftarrow \chi = D_{\mathbb{Z},\alpha q}$ とした時, $b_i = \langle \mathbf{a}_i, s \rangle + e_i \in \mathbb{Z}_q$ とする.

暗号化 集合 S を $\{1, 2, \ldots, m\}$ の中から一様ランダムに選んだ部分集合とする. このとき, 平文ビットが 0 の暗号文を $\left(\sum_{i \in S} \mathbf{a}_i, \sum_{i \in S} b_i\right)$ とし, 平文ビットが 1 の暗号文を $\left(\sum_{i \in S} \mathbf{a}_i, \left\lfloor \frac{q}{2} \right\rfloor + \sum_{i \in S} b_i\right)$ とする.

復号 暗号文 $(\mathbf{a}, b)$ に対し, $b - \langle \mathbf{a}, \mathbf{s} \rangle \in \mathbb{Z}_q$ が $\left\lfloor \frac{q}{2} \right\rfloor$ より 0 に近い場合, 復号結果として 0 を出力し, それ以外の場合は 1 を出力する.

復号の正当性について. 平文 0 を暗号化した暗号文 $(\mathbf{a}, b) = \left(\sum_{i \in S} \mathbf{a}, \sum_{i \in S} b_i\right)$ の場合, $b - \langle \mathbf{a}, \mathbf{s} \rangle \in \mathbb{Z}_q = \sum_{i \in S} (b_i - \langle \mathbf{a}_i, \mathbf{s} \rangle) = \sum_{i \in S} e_i$ なので, $-\frac{q}{4} < \sum_{i \in S} e_i < \frac{q}{4}$ であれば復号に成功し, 0 が出力される.

各ノイズ e_i は Gauss 分布 $\chi = D_{\mathbb{Z},\alpha q}$ から選ばれているので, $\sum_{i \in S} e_i$ の標準偏差は高々 $\sqrt{m}\alpha q$ となる.

ここで, 各パラメータの選択方法から $\sqrt{m}\alpha q < q/\log n$ であり, 非常に高い確率で復号に成功することが分かる. また平文 1 を暗号化した暗号文に対しても同様の議論が成り立つ. この暗号方式の安全性については, LWE 仮定の下で CPA 安全であることが証明されている [45].

ここで紹介した [89] による暗号方式は, 公開鍵のサイズが $O(mn\log q) = \widetilde{O}(n^2)$ で, 暗号文サイズも平文サイズの $O(n\log q) = \widetilde{O}(n)$ 倍に増加するため, 決して効率的ではない. より効率的な方式としては [88] などを参照.

2.2.2 LWE に基づく Lindner, Peikert らによる暗号化方式

Lindner, Peikert らによる暗号化方式 [73] の構成には, 以下のパラメータが必要である.

- n_1, n_2 LWE 問題の安全性パラメータ
- s_k, s_e : 鍵生成時, 暗号化時のノイズ付与のためのガウス分布パラメータ
- q : q-ary 格子を構成する剰余パラメータ ($q > 2$)
- l : 平文空間の次元, $\{0,1\}^l$ を平文の対象空間とする

以下に具体的な暗号方式の構成を示す.

鍵生成 $\mathbb{Z}_q^{n_1 \times n_2}$ からランダムな行列 A を選択する. $n_2 \times l$ 次元ガウス分布 $D_{\mathbb{Z},s_k}^{n_2 \times l}$ から要素 S を選択し秘密鍵とする. $n_1 \times l$ 次元ガウス分布 $D_{\mathbb{Z},s_k}^{n_1 \times l}$ から要素 E を選択し, $P = E - AS \in \mathbb{Z}_q^{n_1 \times l}$ を求め, (A, P) を公開鍵とする.

暗号化 メッセージ $m \in \{0,1\}^l$ に対し, 成分が小さいベクトル $(e_1, e_2, e_3) \in (\mathbb{Z}^{n_1}, \mathbb{Z}^{n_2}, \mathbb{Z}^l)$ の各要素をそれぞれ $D_{\mathbb{Z},s_e}$ から選択する. $c = (e_1 A + e_2, e_1 P + e_3 + m \left\lfloor \frac{q}{2} \right\rfloor)$ とし c を暗号文とする.

復号 $v = c_1 S + c_2$ を求め, 各要素毎に m_i を $|v_i| < \frac{q}{4}$ であれば 0, それ以外であれば 1 として $m = \{m_1, ..., m_l\}$ を復号文とする.

復号の正当性について．平文の 0 に対応する暗号文 $c = (e_1A + e_2, e_1P + e_3)$ に対し，$v = c_1S + c_2 = e_1AS + e_2S + e_1P + e_3 = e_2S + e_1E + e_3$ となる．秘密鍵 S ならびに E の各成分は，Gauss 分布 $D_{\mathbb{Z},s_k}$ から，各ノイズ e_i の成分は Gauss 分布 $D_{\mathbb{Z},s_e}$ から選ばれており，また s_k, s_e の値が一定以下に抑えられることから，高い確率で $|v| < s_e s_k(n_1 + n_2) < \frac{q}{4}$ を満たし，復号に成功することが分かる．また平文ビットが 1 の暗号文に対しても同様の議論が成り立つ．この暗号方式の安全性については LWE 仮定の下で CPA 安全であることが証明されている [73].

2.2.3 Ring-LWE に基づく Brakerski らによる暗号化方式

Brakerski らによる Ring-LWE 問題にもとづく暗号化方式は，暗号化したまま限定回の加算と乗算が可能な somewhat 準同型暗号として提案されているものである．この暗号方式には，以下の 4 つのパラメータが必要である．

- n: 2 のべき乗の整数で，暗号方式を構成する基礎的な環 $R = \mathbb{Z}[x]/(x^n + 1)$ を定義する (n が 2 べきであることから，多項式 $x^n + 1$ が $\mathbb{Z}$ 上既約となることに注意).
- q: $q \equiv 1 \pmod{2n}$ を満たす素数で，暗号文空間の基礎環 $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ を定義する．
- t: 条件 $t < q$ を満たす整数で，暗号方式の平文空間 $R_t = \mathbb{Z}_t[x]/(x^n + 1)$ を定義する．
- $\sigma > 0$: ノイズを与えるための離散ガウス分布のパラメータ．

以下に具体的な暗号方式の構成を示す．なお，$a_0 + a_1x + \cdots + a_{n-1}x^{n-1} \to (a_0, a_1, \ldots, a_{n-1})$ によって，環 R を $\mathbb{Z}^n$ と同一視する．また同様に R_q と $\mathbb{Z}_q^n$ を同一視する．

鍵生成 $s \in R \leftarrow D_{\mathbb{Z}^n,\sigma}$ を選び，一様ランダムに $p_i \in R_q$ を取り，小さなエラー $e \leftarrow \chi$ を固定する．([32] では $s \leftarrow \chi$ を一様ランダムに選択するのに対し，[72] では一様ランダムには選択しない点だけが異なる)．そこで，公開鍵を $\mathsf{pk} = (p_0, p_1)$ とし (ただし，$p_0 = -(p_1s + te)$ とする)，秘密鍵を $\mathsf{sk} = s$ とする．

暗号化 平文情報 $m \in R_t$ と公開鍵 $\mathsf{pk} = (p_0, p_1)$ に対し，まず χ から $u, f, g \in R$ をサンプリングし，暗号文を

$$\mathsf{Enc}(m, \mathsf{pk}) = (c_0, c_1) = (p_0u + tg + m, p_1u + tf),$$

と定義する．ただし，条件 $t < q$ より，この数式では元 $m \in R_t$ を環 R_q の元として見なして計算する．つまり，暗号文は $(R_q)^2$ の元として表現される．

復号 任意の長さの暗号文 $\mathsf{ct} = (c_0, c_1, \ldots, c_\xi)$ に対して，復号は

$$\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}) = [\tilde{m}]_q \bmod t \in R_t,$$

で計算できる．ただし，$\tilde{m} = \sum_{i=0}^{\xi} c_i s^i \in R_q$ であり，$[\tilde{m}]_q$ は元 $\tilde{m}$ の各係数の $[-q/2, q/2)$ への剰余とする．また，$\mathbf{s} = (1, s, s^2, \ldots, s^\xi)$ としたとき，この復号処理を $\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}) = [\langle \mathsf{ct}, \mathbf{s} \rangle]_q \bmod t$ と書き直すこともできる．

この復号アルゴリズムの正当性については，暗号アルゴリズムで得られる暗号文 $\mathsf{ct} = (c_0, c_1)$ に対し，関係式 $p_0 + p_1s = -te$ が成り立つことから，$\langle \mathsf{ct}, \mathbf{s} \rangle = (p_0u + tg + m) + s \cdot (p_1u + tf) = m + t \cdot (g + sf - ue)$ が環 R_q 上で成り立つ．ここで，元 $m + t \cdot (g + sf - ue)$ を環 R の元と見なしたとき，その各係数が $[-q/2, q/2)$ 内に収まっている限り，$[\langle \mathsf{ct}, \mathbf{s} \rangle]_q = m + t \cdot (g + sf - ue)$ が環 R 上で成立する (元 $e, f, g, u \leftarrow \chi$ が十分小さなノイズとして選択されていることに注意)．この場合，剰余 $\bmod t$ の操作で正しい復号結果 $m \in R_t$ が得られる．

また，この暗号方式の安全性については，Ring-LWE 問題の計算量困難性仮定の下で KDM 安全 (key dependent message security) であることが証明されている [32].

2.2.4 NTRU 問題に基づく Hoffstein らによる暗号化方式

Hoffstein らによる NTRU 問題に基づく暗号化方式 NTRUEncrypt [65] の構成には次のパラメータが必要である.

- n : 正の整数 (セキュリティパラメータ)
- q : 正の整数 (素数である必要はない)
- p : q と互に素で $p \ll q$ である正の整数
- ϕ: 次数 n の多項式であり環 $R_p = \mathbb{Z}_p[x]/(\phi)$, $R_q = \mathbb{Z}_q[x]/(\phi)$ を定義する (ϕ としては例えば $x^n \pm 1$, $x^n - x - 1$ 等)

以下に具体的な暗号方式の構成を示す.

鍵生成 すべての係数の絶対値が小さい二つの多項式 $f \in R_q, g \in R_q$ を選ぶ. ただし, f は R_p, R_q の両方において可逆な要素とする. すなわち, ある f_p, f_q が存在し, 以下を満たす.

$$f_p \cdot f = 1 \in R_p, f_q \cdot f = 1 \in R_q$$

ここで f, f_p を秘密鍵とし, $h = pf_q \cdot g \in R_q$ を公開鍵とする. なお f_p, f_q ならびに g は f と h を用いて復元可能であることに注意する.

暗号化 平文情報として, すべての係数の絶対値が p より小さい (例えば $-1, 0, 1$ のいずれかである) 要素 $m \in R_q$ とし, 公開鍵 $\mathsf{pk} = h$ に対し, $r \in R_q$ を係数が小さい多項式からランダムに選び, 暗号文を

$$\mathsf{Enc}(m, \mathsf{pk}) = r \cdot h + m \in R_q$$

と定義する.

復号 暗号文 $c \in R_q$ に対し, 復号は

$$\mathsf{Dec}(m, \mathsf{sk}) = [f_p \cdot [f \cdot c]_q]_p$$

で求められる. ただし $[a]_q$, $[a]_p$ は元 $a \in R_q$ の各係数をそれぞれ $[-q/2, q/2)$, $[-p/2, p/2)$ に収めたものとする.

復号の正当性については, 次のように示される. $[f \cdot c]_q$ は, $f \cdot c = f \cdot (r \cdot h + m) = f(r \cdot pf_q \cdot g + m) = pr \cdot g + f \cdot m \in R_q$ と変形されるが, r, g, f, m 共に, 係数が小さいものから抽出しており, また $p \ll q$ であること, 更に係数が $[-q/2, q/2)$ に収められていることから適切なパラメータ選択により, $f \cdot c$ は q による剰余を伴わない等式, すなわち $f \cdot c = pr \cdot g + f \cdot m \in Z[x]/(\phi)$ が満たされる. また右辺第一項は p 倍項であることから, 続く p による剰余で消去され, $f_p \cdot (pr \cdot g + f \cdot m) = f_p \cdot f \cdot m = m \in R_p$ となり正しい復号結果 m が得られる.

この NTRU Encrypt 暗号の安全性についてはアルゴリズム提案当初格子問題への安全性帰着がついていなかったが, Stehlé ら [98] により, standard model の CPA 仮定に基づくイデアル格子上の Ring-SIS 問題, ならびに Ring-LWE 問題に帰着されることが示されている.

2.2.5 Hash-and-Sign に基づく署名方式の格子問題への拡張

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

- M : メッセージ

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

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

この署名方式は, さまざまな改良が提案されているが, 格子問題の困難性に基づく落とし戸つき関数を用いた Hash-and-Sign 署名方式が, Gentry らによって提案されている [60]. 以下にその方式を示す. 次のパラメータを準備する.

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

以下に具体的な署名方式を示す.

鍵生成 $A \in \mathbb{Z}_q^{n \times m}$ をランダムな行列, $S \in \Lambda_q^{\perp}(\mathbf{A}, \mathbf{q}), ||S|| < L$ を短いベクトルとし, $SA = 0 \mod q$ を満たす行列の組 (A, S) を生成する (具体的な手法は [21] 参照). 秘密鍵を S, 公開鍵を A とする.

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

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

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

2.2.6 Fiat-Shamir 署名方式の格子問題への拡張

Fiat, Shamir らによって提示された Fiat-Shamir 変換 [56] に基づく署名方式を総称して Fiat-Shamir 署名と呼ばれており, 現在までさまざまな方式が提案されている. 以下に基本となる方式の一つである素因数分解問題をベースとする方式を記す. 合成数 $n = pq$ (p, q は素数) を法とするべき乗剰余演算 $g(x) = g^x \mod n$ を一方向性関数として利用し, 秘密鍵 s, 公開鍵 $a = g(s)$ を準備する.

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

Lyubashevsky によって, Fiat-Shamir with Aborts 型の格子ベースの署名方式が提案されている [56]. 以下にその具体的な署名方式について述べる. 次のパラメータを準備する.

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

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

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

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

署名生成 メッセージを M とする.

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

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

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

2.3 格子に基づく主要な暗号方式

本節では，格子に基づく主要な暗号方式として，5 つの公開鍵暗号と 2 つの署名を取り上げ，その概要と設計原理を説明する.

格子を用いた主な公開鍵暗号の構成として，最初期の Ajtai-Dwork 型 [13]，GGH 型 [58] から近年の [89] による LWE 型 (Regev 型)，[60, 73] に代表される dual-LWE 型，[65] に代表される NTRU 型が存在する. 格子を用いた署名の構成としては主に GGH/NTRUSign 型 [58, 65]，Fiat-Shamir with abort 型 [77, 78]，Hash-and-Sign 型 [60, Sect.6]，Plantard-Susilo-Win 型 [86] 等が知られている[*4].

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

これらの構成は安全性，実装時の性能，使いやすさなどの面から様々な長所・短所を持つが，できる限り幅広くそれらを解説するため，以下の表 2.1 に挙げる 7 つの方式を紹介する.

- FrodoKEM は dual-LWE 型の公開鍵暗号であり，安全性の根拠に LWE 問題の計算困難性を仮定している. 保

[*4] この分類に関しては例えば [64, Sect. 3]，[49, Sect. 5.5] 等を参照.

表 2.1: 格子に基づく暗号の分類

文献	暗号化	鍵交換	署名
NewHope [4]	○	○	
FrodoKEM [11]	○	○	
NTRU [33]	○	○	
SABER [29]	○	○	
CRYSTALS-Kyber [9]	○	○	
CRYSTALS-Dilithium [23]			○
FALCON [54]			○

守的な構成を設計指針としており，将来 Ring 型や Module 型の構造を持つ格子問題に効率的な解法が発見された場合でも安全性が確保されると期待される．保守的な構成という観点から取り上げる．

- NewHope は dual-LWE 型の公開鍵暗号であり，安全性の根拠に $x^n + 1, n = 2^k$ の形の多項式により定義される環上の Ring-LWE 問題の困難性を置いている．環の構造を活用した数論変換による高速実装，サイズの低減という観点から取り上げる．
- NTRU は NTRU 型の公開鍵暗号であり，NTRU 格子上の格子問題の計算困難性を安全性の根拠としている．NTRU 暗号は 1996 年の提案依頼改良が続けられてきたが，本方式は近年の研究成果が数多く盛り込まれた形となるため取り上げる．
- SABER は LWE 型の公開鍵暗号であり[*5]，$x^{256} + 1$ を定義多項式とする環上の Module-LWR 問題の計算困難性を安全性の根拠としている．LWR 問題を用いることで実装時のサンプリングを極力負担の少ないものとしている．この観点から取り上げる．
- CRYSTALS-Kyber は dual-LWE 型の公開鍵暗号であり，安全性の根拠に $x^n + 1, n = 2^k$ の形の多項式により定義される環上の Module-LWE 問題の困難性を置いている．NIST PQC 標準化の Selected Algorithm となったことから取り上げる．
- CRYSTALS-Dilithium は Fiat-Shamir 型の署名方式であり，$x^{256} + 1$ を定義多項式とする環上の Module-LWE 問題の計算困難性を安全性の根拠としている．環の性質を用いた数論変換による高速処理とサイズの圧縮が可能であり，公開鍵サイズと署名サイズの和を最小化することを目的としてパラメータ設計を行っている．環の性質を用いた処理の効率化の観点から取り上げる．
- FALCON は Hash-and-Sign 型の署名方式であり，$x^n + 1$ を定義多項式とする NTRU 格子上の SIS 問題の困難性を安全性の根拠としている．格子上の高速フーリエサンプリングを用いた高速な署名生成を特徴とし，方式提案後も数多くの改良が提案されていることから取り上げる．

[*5] 仕様書 [29] の設計原理の項には “Encryption: we use a simple LWR version of Regev’s LWE encryption scheme [35], where the encryption part is compressed (using the parameter T) to save on bandwith.” とあるが, Second PQC Standardization Conference の発表スライド [39, p.5] においてはひな型を dual-LWE 型暗号としている．数式の比較から，どちらも原型と考えることが可能であるため，本報告書では仕様書に従い LWE 型であるとした．

2.3.1 FrodoKEM

歴史: FrodoKEM は 2016 年の国際会議 CCS において Joppe Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, Douglas Stebila の 8 名の連名で公表された [22].

2017 年 11 月の NIST PQC 公募に提出された版 [10] では [22] の著者 8 名から Craig Costello が抜け, Erdem Alkim, Patrick Longa, Christopher Peikert の 3 名を加えた 10 名が inventors，Karen Easterbrook, Brian LaMacchia を Additional submitters とした合計 12 名での提案となっている.

NIST PQC 提出後のディスカッションを通じて修正が加えられ，現在の最新版は 2021 年 6 月に公表された [11] である．本節の記述はこの仕様書に従う.

参照 URL: 開発者による公式ページ `https://frodokem.org/` およびリファレンスコード `https://github.com/Microsoft/PQCrypto-LWEKE` を参照した.

設計原理: FrodoKEM は LWE 問題を安全性の根拠とする公開鍵暗号方式であり，将来 Ring 型や Module 型の構造を持つ格子問題を用いた暗号に致命的な脆弱性が発見された場合でも安全性が確保されると期待されることを特徴としている[*6]．LWE 問題自体の単純さからパラメータ設定の小回りが利くことも長所の一つとしている．形式的には Gentry-Peikert-Vaikuntanathan の [60, Sect. 7.1]，Lindner-Peikert [73] をひな型とする dual-LWE 暗号に分類される．IND-CPA 安全な公開鍵暗号方式を構成した後に，Hofheinz ら [62] のモジュール化藤崎-岡本変換 $\mathsf{FO}^{\not\perp}$ に Bos らの [24] の修正を施した変換手法 [11, Def. 2.19] を適用し IND-CCA2 安全な KEM を構成している.

アルゴリズムの詳細: 表 2.2，2.3，2.4 に Lindner-Peikert[73] による格子ベース公開鍵暗号と FrodoKEM の鍵生成，暗号化，復号アルゴリズムを並置する.

パブリックパラメータは以下で与えられる.

- q: 計算の剰余環 $\mathbb{Z}_q$ を指定する．ここでは $q = 2^D$ の形とし，$D = 15, 16$ に固定される.
- $n, \overline{m}, \overline{n}$: 行列のサイズを指定する．$n$ は 8 の倍数とする．また，平文は $\overline{m} \times \overline{n} = 8 \times 8$ 行列に符号化される.
- B, ℓ: 平文行列に符号化する情報量を指定する．行列の各成分は $0, \ldots, 2^B - 1$ の整数で表現され，合計で $\ell = B \cdot \overline{m} \cdot \overline{n}$ ビットの情報を埋め込むことができる.
- $\mathsf{len}_{\mathsf{seed}_A}, \mathsf{len}_{\mathsf{seed}_{SE}}$: 擬似ランダム行列 A, S, E を生成するためのシードとなるビット列の長さで，提案パラメータセットでは $\mathsf{len}_{\mathsf{seed}_A}$ は 128 に固定され，$\mathsf{len}_{\mathsf{seed}_{SE}}$ はセキュリティレベルに合わせて $128, 192, 256$ の値を取る.
- T_χ: SampleMatrix 関数で用いられる確率分布の表で，セキュリティレベルごとに離散ガウス分布からの Rényi divergence が小さくなるように設計されている．具体的な値は [11, Table 3] を参照.

関数内で用いられるサブルーチン群を以下に記述する.

- Frodo.Gen 関数はシードとなるビット列 seed と SHAKE ハッシュ関数を用いて擬似ランダム行列 $A \in \mathbb{Z}_q^{n \times n}$ を生成する関数である．i 行目を生成する際に整数 i を 16 ビットのビット列にエンコードした $\langle i \rangle$ を用いて $\mathsf{SHAKE}(\langle i \rangle || \mathsf{seed}, 16n)$ を呼び出し，得られた $16n$ ビットを 16 ビットごとに分割することで n 個の整数 $c_0, \ldots, c_{n-1} \in \{0, \ldots, 2^{16} - 1\}$ として，$A_{i,j} = c_j \mod q$ の形で各成分に振り分けている．このように関数を構成することで，各 i 行目を生成する操作がハードウェアによる並列実装に適した形となる．また，q は 2 べきの形で取られるため，各 $A_{i,j}$ の分布に偏りは生じない.

[*6] FrodoKEM の仕様書 [11] では構造を持たないことを “algebraically unstructured” と表現している.

- Frodo.SampleMatrix$((\boldsymbol{r}^{(0)},\ldots,\boldsymbol{r}^{(nm-1)}),n,m,T)$ 関数は行列のサイズ $n\times m$ と (i,j) 成分の生成に用いるシード $\boldsymbol{r}^{(in+j)}$ の列，乱数の確率分布を示すテーブル T を入力とする．それぞれのシードの長さは 16 ビットに固定されている．T は整数上の中心対称な確率分布が $\Pr[|T|=t]$ のテーブルとして与えられており，シード $\boldsymbol{r}^{(in+j)}$ の先頭 15 ビットで (i,j) 成分の絶対値を，残りの 1 ビットで符号を決定しサンプリングを行う．
- Frodo.Encode,Frodo.Decode 関数は $\ell=B\cdot\overline{m}\cdot\overline{n}$ ビットの平文を $\overline{m}\times\overline{n}$ 行列に埋め込む関数とその逆演算を行う関数である．B ビットの整数 k を $\mod q$ に埋め込むため，行列の成分を $k\cdot\lfloor q/2^B\rfloor$ とする．

表 2.2: Lindner-Peikert 格子ベース暗号および FrodoKEM における鍵生成関数の比較

	Lindner-Peikert[73, Sect. 3.1] $\mathsf{KeyGen}(1^\lambda)\to(pk,sk)$	FrodoKEM [11, Algorithm 9] $\mathsf{KeyGen}(1^\lambda)\to(pk,sk)$
1:	A: $n_1\times n_2$ ランダム行列	$\mathsf{seed}_A\xleftarrow{\$}\{0,1\}^{\mathsf{len}_{\mathsf{seed}_A}}$ $A\leftarrow\mathsf{Frodo.Gen}(\mathsf{seed}_A)$
2:	S: 成分の小さい $n_2\times\ell$ 行列	$\mathsf{seed}_{SE}\xleftarrow{\$}\{0,1\}^{\mathsf{len}_{\mathsf{seed}_{SE}}}$ //擬似乱数ビットの生成 $(\boldsymbol{r}^{(0)},\ldots,\boldsymbol{r}^{(2n\overline{n}-1)})\leftarrow\mathsf{SHAKE}(\mathsf{0x5F}\|\mathsf{seed}_{SE},2n\overline{n}\cdot\mathsf{len}_\chi)$ $S^T\leftarrow\mathsf{Frodo.SampleMatrix}((\boldsymbol{r}^{(0)},\ldots,\boldsymbol{r}^{(n\overline{n}-1)}),\overline{n},n,T_\chi)$
3:	E: 成分の小さい $n_1\times\ell$ 行列	$E\leftarrow\mathsf{Frodo.SampleMatrix}((\boldsymbol{r}^{(n\overline{n})},\ldots,\boldsymbol{r}^{(2n\overline{n}-1)}),\overline{n},n,T_\chi)$
4:	$B=AS+E$	$B=AS+E$
return	$pk=(A,B),sk=S$	$pk=(\mathsf{seed}_A,B),sk=S^T$

FrodoKEM の鍵生成関数（表 2.2 右）を説明する．鍵生成のためのシード seed_A と seed_{SE} を生成した後，Frodo.Gen 関数と Frodo.SampleMatrix 関数を用いて A,S,E を生成する．このとき，行列乗算時のメモリアクセスの順序を考えて S は転置の形で格納される．公開鍵行列 A は鍵サイズ圧縮のために成分ではなくシード seed_A の形で格納される．

FrodoKEM の暗号化関数（表 2.3 右）を説明する．seed_A から行列 A を復元した後，暗号化用の乱数行列 S',E',E'' を擬似乱数列 $\boldsymbol{r}^{(i)}$ から生成する．擬似乱数列の生成には SHAKE ハッシュ関数を用いるが，パディング値 0x96 が鍵生成で用いられた 0x5F と異なるため鍵生成の S,E とは異なる行列が得られることに注意．残りの処理はひな型の Lindner-Peikert 暗号を行列化したものである．

FrodoKEM の復号関数（表 2.4 右）は Lindner-Peikert 暗号の復号処理を行列化したものである．

安全性とパラメータ: ベースとなる IND-CPA 安全な公開鍵暗号の安全性は判定版 LWE 問題に帰着される．実装上の効率化のため，鍵生成，暗号化処理において離散ガウス分布を近似した確率分布 T_χ を用いているが，その際の安全性の低下は Rényi divergence を用いた議論により評価されている [11, Sect. 5.1]．n は格子の次元で，大きくとることで安全性レベルが上がるが処理コストも上がる．q は環を定義する法で，大きく取ることで平文空間も大きくなるが，格子が疎になり安全性が下がる．σ は離散ガウス分布の大きさを決定するパラメータで，大きくとることで安全性が上がるが，復号エラー率が上がる．

FrodoKEM のパラメータは LWE 問題の Primal 攻撃，Dual 攻撃双方での BKZ アルゴリズムを用いた計算量評価から求められている．保守的なパラメータ設定のため，Core-SVP, BKZ の計算量評価を，計算量の上界を示す既存の攻撃手法のみではなく，下界からの議論が行われていることも方式の特徴である．

変種: 行列 A の生成に SHAKE-256 ではなく，AES-128 を使ったバージョンも提案されている．AES-NI 命令を用いた

表 2.3: Lindner-Peikert 格子ベース暗号および FrodoKEM における暗号化関数の比較

	Lindner-Peikert[73, Sect. 3.1] $\mathsf{Enc}(pk=(A,B),$ $\boldsymbol{m}\in\{0,1\}^{\ell})\to\mathsf{ct}$	[11, Algorithm 10] $\mathsf{Enc}(pk=(A,B),$ $\mu\in\{0,1\}^{\ell}))\to\mathsf{ct}$
0:		$A\leftarrow\mathsf{Frodo.Gen}(\mathsf{seed}_A)$ // A の復元
1:	$\boldsymbol{s}',\boldsymbol{e}',\boldsymbol{e}''$: 成分の小さいベクトル	$\mathsf{seed}_{SE}\xleftarrow{\$}\{0,1\}^{\mathsf{len}_{\mathsf{seed}_{SE}}}$ $(\boldsymbol{r}^{(0)},\ldots,\boldsymbol{r}^{(2n\overline{n}-1)})\leftarrow\mathsf{SHAKE}(\mathsf{0x96}\|\mathsf{seed}_{SE},$ $(2\overline{m}\cdot n+\overline{m}\cdot\overline{n})\cdot\mathsf{len}_{\chi})$ //擬似乱数ビットの生成 $S'\leftarrow\mathsf{Frodo.SampleMatrix}((\boldsymbol{r}^{(0)},\ldots,\boldsymbol{r}^{(\overline{m}\cdot n-1)}),\overline{m},n,T_{\chi})$ $E'\leftarrow\mathsf{Frodo.SampleMatrix}((\boldsymbol{r}^{(\overline{m}\cdot n)},\ldots,\boldsymbol{r}^{(2\overline{m}\cdot n-1)}),\overline{m},n,T_{\chi})$ $E''\leftarrow\mathsf{Frodo.SampleMatrix}((\boldsymbol{r}^{(2\overline{m}\cdot n)},\ldots,\boldsymbol{r}^{(2\overline{m}\cdot n+\overline{m}\cdot\overline{n}-1)}),\overline{m},\overline{n},T_{\chi})$
2:	$\boldsymbol{u}=\boldsymbol{s}'A+\boldsymbol{e}'$ $\boldsymbol{v}=\boldsymbol{s}'B+\boldsymbol{e}''+\boldsymbol{m}\cdot\left\lfloor\frac{q}{2}\right\rfloor$	$B'=S'A+E'$; $V=S'B+E''$ $C_1=B'$; $C_2=S'B+E''+\mathsf{Frodo.Encode}(\mu)$
return	$\mathsf{ct}=(\boldsymbol{u},\boldsymbol{v})$	$\mathsf{ct}=(C_1,C_2)$

表 2.4: Lindner-Peikert 格子ベース暗号および FrodoKEM における復号関数の比較

	Lindner-Peikert[73, Sect. 3.1] $\mathsf{Dec}(sk,\mathsf{ct})\to\boldsymbol{m}'$	FrodoKEM [11, Algorithm 11] $\mathsf{Dec}(sk,\mathsf{ct})\to\boldsymbol{m}'$
1:	$\overline{\boldsymbol{m}}=\boldsymbol{v}-\boldsymbol{u}S$ $m'_i=\begin{cases}0 & \lvert m_i\rvert\le\lfloor q/4\rfloor\\ 1 & \text{それ以外}\end{cases}$	$M=C_1-C_2S$ $\boldsymbol{m}'=\mathsf{Frodo.Decode}(M)$
return	$\boldsymbol{m}'=(m'_1,\ldots,m'_\ell)$	$\boldsymbol{m}'$

Intel CPU による実装では SHAKE を用いたものよりも 2.5 倍程度高速である [11, Sect. 3.2].

また，Encode Decode 関数に誤り訂正符号を用いて暗号文サイズを 1 割ほど削減したバージョンが提案されている [96, 93].

補足情報: 構造を持つ格子 (structured lattice) でないという理由から 3rd Round での候補として残っていたが，標準化から漏れた理由として，構造を持つ格子でないという性質を持つ符号ベース暗号の BIKE,HQC や同種写像ベース暗号の SIKE と比較すると，パフォーマンスの観点から不利であったと NIST の標準化レポート [5, p.17] に述べられている．

NIST PQC 第 2 ラウンドのバージョンでは，藤崎-岡本変換を行った IND-CCA2 KEM の実装において再暗号化後のチェックが定数時間ではないことから鍵復元攻撃が可能であることが示され [59]，修正されている．実装にかかわる攻撃として，ロウハンマー (Rowhammer) 攻撃による鍵復元攻撃が新たに発見された [55].

表 2.5: FrodoKEM CCA のパラメータ [11, Table 5]. σ の値は T_χ の元となる離散ガウス分布の標準偏差を示す．秘密鍵サイズはデカプセル化時に用いられる鍵情報の中から，公開鍵に相当するものを除いた分である．公開鍵，秘密鍵，平文，暗号文サイズの単位はそれぞれ Byte である．

(n, q, σ)	安全性レベル	公開鍵サイズ	秘密鍵サイズ	平文サイズ	暗号文サイズ
$(\ 640, 2^{15}, 2.8)$	レベル 1	9,616	10,272	16	9,720
$(\ 976, 2^{16}, 2.3)$	レベル 3	15,632	15,664	24	15,744
$(1344, 2^{16}, 1.4)$	レベル 5	21,520	21,568	32	21,632

2.3.2 NewHope

歴史: NewHope の最初のバージョンは 2016 年に国際会議 USENIX Security において Alkim, Ducas, Pöppelmann, Schwabe により鍵共有プロトコルとして発表された [17]．また，直後に reconciliation によるエラー訂正プロセスを省略し簡略化した [18] が ePrint Archive において発表されている．

2017 年 11 月の NIST PQC 公募に応募された Version 1.0[2] は新たに Roberto Avanzi, Joppe Bos, Antonio de la Piedra, Douglas Stebila の 4 人が加わった合計 8 人での提案とし，[18] をベースとして公開鍵暗号方式を構成している．NIST PQC 提出後のディスカッションを通じて何回かの修正が加えられ，現在の最新版は 2020 年 4 月に公表された Version 1.1[4] である．

2nd round に提出された Version 1.02[3] では，Martin R. Albrecht, Emmanuela Orsini, Valery Osheter, Kenneth G. Paterson, Guy Peer, Nigel P. Smart の 6 人が Contributor として列挙されている．

本節の記述は最新版の仕様書 [4] に従う．

参照 URL: 開発者による公式ページ `https://newhopecrypto.org/` および GitHub 上のリファレンス実装 `https://github.com/newhopecrypto/newhope` を参照した．

設計原理: NewHope は環 $\mathbb{Z}[x]/(x^n + 1), n = 2^k$ 上の Ring-LWE 問題の計算困難性を安全性の根拠とする公開鍵暗号方式であり，形式的には Gentry-Peikert-Vaikuntanathan の [60, Sect. 7.1]，Lindner-Peikert [73] をひな型とする dual-LWE 暗号に分類される．ベースとなる暗号方式におけるベクトルと行列の演算を多項式環の要素に置き換え IND-CPA 安全な公開鍵暗号方式を提案している．その際，数論変換を用いた乗算などの実装テクニックを用いて処理を高速化している．環の定義多項式を $x^n + 1, n = 2^k$ の形としている点も高速化に寄与しているが，その一方でパラメータ選択の自由度に制限があり，NIST PQC の提案方式では安全性レベル 1 およびレベル 5 のパラメータセットのみが提案されている．

IND-CPA 安全な公開鍵暗号から IND-CCA 安全な KEM の構成には [62] のモジュール化された藤崎-岡本変換 $\mathsf{QFO}^{\not\perp}_m$ を用いているが，その際に公開鍵暗号 CRYSTALS-Kyber（本報告書の 2.3.5 節も参照）の IND-CCA 安全な KEM の構成 [24, Sect. 4] に用いられた手法を取り入れ微修正を施している．結果として，構成された KEM が ROM, QROM の双方のモデルにおいて CCA 安全であることが保証されている．

アルゴリズムの詳細: 表 2.6，2.7，2.8 に Lindner-Peikert[73] による格子ベース公開鍵暗号と NewHope の鍵生成，暗号化，復号アルゴリズムを並置する．

パブリックパラメータは以下で与えられる．

- n, q: 演算を行う環 $R_q := \mathbb{Z}_q[x]/(x^n+1)$ を定義する．特に指定のない場合には多項式の係数は自動的に区間 $[-q/2, q/2)$ 内に収められるものとする．高速数論変換のため，n は 2 のべき乗の形であり，さらにアルゴリズム中で用いられる ω, γ が存在するために q は $q \equiv 1 \pmod{2n}$ を満たす素数として選ばれる．NewHope のパラメータ設定では $n = 512, 1024, q = 12289$ が選ばれている．
- k: ノイズの大きさを設定する．
- ω, γ: 数論変換で用いる．$\mathbb{Z}_q^\times$ における 1 の原始 n 乗根を ω, その平方根を $\gamma := \sqrt{\omega} \mod q$ とする．NewHope のパラメータ設定では $n = 512$ に対して $(\omega, \gamma) = (3, 10968)$，$n = 1024$ に対して $(\omega, \gamma) = (49, 7)$ が取られている．

関数内で用いられるサブルーチン群を以下に記述する．

- $\mathsf{Sample}(\mathsf{seed}, \mathsf{nonce})$ 関数は，各係数を平均ゼロに調整した二項分布 ψ_8 から独立にサンプリングした n 次多項式を 32Bytes の seed と非負整数値 nonce から生成する．自然数 k に対して, ψ_k の出力は独立にサンプリングした $2k$ 個のビット $b_i, b'_i \xleftarrow{\$} \{0,1\}$ $(k = 1, \ldots, k)$ に対する $\sum_{i=1}^{k}(b_i - b'_i)$ として定義される．
- $\mathsf{PolyBitRev}(a \in R_q)$: 高速数論変換を用いた乗算の場合，結果のインデックス順序が入れ替わるため配列の要素 $c[i]$ を $x^{\mathsf{BitRev}}(i)$ として解釈しなければならない．ただし，ビット反転は $h = \log_2(n)$, $i = \sum_{j=0}^{h-1} b_j 2^j$ と 2 進数展開したときに, $\mathsf{BitRev}(i) := \sum_{j=0}^{h-1} b_j 2^{h-j-1}$ 計算される．関数は多項式 $a(x) = \sum_{i=0}^{n-1} a_i x^i$ に対して, 指数部分をビット反転した多項式 $\sum_{i=0}^{n-1} a_i x^{\mathsf{BitRev}(i)}$ を出力する．
- $\mathsf{NTT}(a), \mathsf{NTT}^{-1}(\hat{a})$: R_q の多項式同士の乗算を高速化するため，数論変換 (NTT: Number Theoretic Transform)[4, p. 7-9] を用い，鍵と暗号文の処理を極力 NTT 空間で行う工夫がなされている．パラメータ n, q, ω, γ を固定したとき，多項式 $a(x) = \sum_{i=0}^{n-1} a_i x^i \in R_q$ の数論変換

$$\hat{a} = \mathsf{NTT}(a) := \sum_{i=0}^{n-1} \hat{a}_i x^i,\ \hat{a}_i := \sum_{j=0}^{n-1} \gamma^j a_j \omega^{ij} \pmod q$$

および逆変換を

$$a = \mathsf{NTT}^{-1}(\hat{a}) := \sum_{i=0}^{n-1} a_i x^i,\ a_i := \left(n^{-1}\gamma^{-i} \sum_{j=0}^{n-1} \hat{a}_j \omega^{-ij} \right) \pmod q$$

とする．これらは線形変換であるので，$a, b \in R_q$ に対して $\mathsf{NTT}(a) + \mathsf{NTT}(b) = \mathsf{NTT}(a+b)$ 等の性質がなりたつ．また，$a * b := \sum_{i=0}^{n-1} c_i x^i, c_i = \sum_{j=0}^{n-1} a_j b_{i-j \mod n} \mod q$ は $a * b = \mathsf{NTT}^{-1}(\mathsf{NTT}(a) \circ \mathsf{NTT}(b))$ を満たす．ただし，記号 $\circ$ は多項式の係数同士の積を取ることを表す．
$a * b$ を定義式通りに計算すると mod q での演算が $O(n^2)$ 回必要であるのに対して，高速数論変換を用いた方法では $O(n\log n)$ 回の演算で可能である．

数論変換を用いた高速乗算ではどこかのタイミングで添え字のビット反転を行う必要がある．単純な IND-CPA PKE の実装における最適なのみを考えるのであればこのような置換は必要ないが，IND-CCA KEM のデカプセル化

処理内での再暗号化まで含めて実装を最適化した結果，NewHope では最初の KeyGen, Enc 関数の中で置換が行われている [4, p.8].

なお，仕様書 [4] には多項式とバイト列の相互変換を行う関数 EncodePK, EncodePolynomial, EncodeC, Compress, DecodePK, DecodePolynomial, DecodeC, Decompress が定義されているが，何れもデータの再配置を行う関数であり方式を説明する上では本質的ではないため省略した.

表 2.6: Lindner-Peikert 格子ベース暗号および NewHope における鍵生成関数の比較

	Lindner-Peikert[73, Sect. 3.1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	NewHope[4, Algorithm 1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
1:	A: $n_1 \times n_2$ ランダム行列	$\mathsf{seed} \xleftarrow{\$} \{0,1,\ldots,255\}^{32}$, $z = \mathsf{SHAKE256}(64, \mathsf{seed})$ //32Bytes の seed を 64Bytes に伸長 $\hat{a} = \mathsf{GenA}(\mathsf{z}[0:31]) \in R_q$
2:	S: 成分の小さい $n_2 \times \ell$ 行列	$s = \mathsf{PolyBitRev}(\mathsf{Sample}(\mathsf{z}[32:63], 0)) \in R_d$; $\hat{s} = \mathsf{NTT}(s)$
3:	E: 成分の小さい $n_1 \times \ell$ 行列	$e = \mathsf{PolyBitRev}(\mathsf{Sample}(\mathsf{z}[32:63], 1)) \in R_d$; $\hat{e} = \mathsf{NTT}(e)$
4:	$B = E - AS$	$\hat{b} = \hat{a} \circ \hat{s} + \hat{e}$ //$b = \mathsf{NTT}(a * s + e)$
return	$pk = (A, B), sk = S$	$pk = (\hat{a}, \hat{b}), sk = \hat{s}$

NewHope の鍵生成関数（表 2.6 右）を説明する．ステップ 1 では 32Bytes(= 256bits) の乱数のシードを SHAKE-256 ハッシュ関数を用いて 64Bytes の擬似乱数列 z を生成し，それを前半の $\mathsf{z}[0:31]$ と後半の $\mathsf{z}[32:63]$ に分割する．Lindner-Peikert 型暗号（左側）におけるランダム行列 A の生成に対応して，$\mathsf{GenA}(\cdot)$ 関数は 32Bytes の列をシードとして，ランダムな R_q の元を生成する．各係数が一様独立に Z_q の元からサンプリングされる．実装は [4, Algorithm 5] を参照．出力された $\hat{a}$ がランダムな多項式 a の数論変換であることは，ランダム多項式の数論変換がまたランダム多項式となることから従う.

ステップ 2,3 ではそれぞれ係数の小さい多項式 s, e の数論変換を計算する．ステップ 1 で生成した擬似乱数の後半 $z[32:63]$ をシードに用いて小さい値を係数に持つ多項式のサンプリングを行い，数論変換のためのビット反転処理をしたものを s，その数論変換を $\hat{s}$ とする．$e, \hat{e}$ についても同様.

ステップ 4 では数論変換後の多項式から公開鍵 $\hat{b}$ を計算する．数論変換の性質より，これは $a * b + e$ の NTT 表現となる.

表 2.7: Lindner-Peikert 格子ベース暗号および NewHope における暗号化関数の比較

	Lindner-Peikert[73, Sect. 3.1] $\mathsf{Enc}(pk = (A, B), \boldsymbol{m} \in \{0,1\}^\ell) \to \mathsf{ct}$	NewHope[4, Algorithm 1] $\mathsf{Enc}(pk = (A, B), M \in \{0,1,\ldots,255\}^{32}) \to \mathsf{ct}$
1:	$\boldsymbol{t}, \boldsymbol{e}', \boldsymbol{e}''$: 成分の小さいベクトル	$\mathsf{coin} \xleftarrow{\$} \{0,1,\ldots,255\}^{32}$ // ランダムシード $s' = \mathsf{PolyBitRev}(\mathsf{Sample}(\mathsf{coin}, 0)) \in R_d$; $\hat{t} = \mathsf{NTT}(s')$ $e' = \mathsf{PolyBitRev}(\mathsf{Sample}(\mathsf{coin}, 1)) \in R_d$ $e'' = \mathsf{Sample}(\mathsf{coin}, 2) \in R_d$
2:	$\boldsymbol{u} = \boldsymbol{t}A + \boldsymbol{e}'$ $\boldsymbol{v} = \boldsymbol{t}B + \boldsymbol{e}'' + \boldsymbol{m} \cdot \lfloor \frac{q}{2} \rfloor$	$\hat{u} = \hat{a} \circ \hat{t} + \mathsf{NTT}(e')$ $v' = \mathsf{NTT}^{-1}(\hat{b} \circ \hat{t}) + e'' + \mathsf{Encode}(M)$
return	$\mathsf{ct} = (\boldsymbol{u}, \boldsymbol{v})$	$\mathsf{ct} = (\hat{u}, v')$

NewHope の暗号化関数（表 2.7 右）を説明する．暗号化のためのランダム多項式 $\hat{t}, e', e''$ を生成するため，鍵生成のステップ 2-3 と同様の処理を行う．$\hat{t}$ は数論変換後の形式であるが，e', e'' は通常の形式のまま用いる．

Encode 関数は 32Bytes(=256bits) の平文を n 次多項式の係数として埋め込む．NewHope のパラメータでは $n = 512, 1024$ が用いられるため，1bit の情報が複数箇所に埋め込まれることになる．具体的には平文の $i$bit 目を b_i，多項式の j 次の係数を v_j としたときに $j = 0, \ldots, n-1$ に対して

$$v_j = \begin{cases} 0 & (b_{j \bmod 256} = 0) \\ \left\lfloor \frac{q}{2} \right\rfloor & (b_{j \bmod 256} = 1) \end{cases}$$

とする．

表 2.8: Lindner-Peikert 格子ベース暗号および NewHope における復号関数の比較

	Lindner-Peikert[73, Sect. 3.1] $\mathsf{Dec}(sk, \mathsf{ct}) \to m'$	NewHope[4, Algorithm 1] $\mathsf{Dec}(sk, \mathsf{ct}) \to M'$
1:	$\overline{\boldsymbol{m}} = \boldsymbol{v} + \boldsymbol{u}S$ $m'_i = \begin{cases} 0 & \lvert m_i \rvert \le \lfloor q/4 \rceil \\ 1 & \text{それ以外} \end{cases}$	$\overline{m} = v - \mathsf{NTT}^{-1}(\hat{u} \circ \hat{s})$ $M' = \mathsf{Decode}(\overline{m})$
return	$\boldsymbol{m'} = (m'_1, \ldots, m'_\ell)$	M'

NewHope の復号関数（表 2.15 右）を説明する．健全性の証明より，$v - \mathsf{NTT}^{-1}(\hat{u} \circ \hat{s}) = v - u * s$ が $\mathsf{Encode}(M)$ と小さいノイズ和であることが示されるため，Decode 関数はノイズの除去と平文 M' の復元を同時に行う [4, Algorithm 11]．多項式 $\overline{m}$ の係数の中で，$\overline{m}_{i+256k}$ $(k = 0, \ldots, n/256 - 1)$ の中にビット M'_i の情報が埋め込まれているため，それらを多数決で決定する．具体的には，$\sum_{k=0}^{n/256-1} |\overline{m}_{i+256k} - (q-1)/2|$ が $M'_i = 0$ の場合には $(n/256) \cdot (q-1)/2 \approx (n/512)q$ に近く，$M'_i = 1$ の場合には 0 に近い値を取るため，和から $(n/1024)q$ を引いた後に符号を見ることでビット列の復元が完了する．

安全性とパラメータ: ベースとなる IND-CPA 安全な公開鍵暗号の安全性は環 $Z_q[x]/(x^n + 1)$ 上の判定版 LWE 問題に量子帰着されることが示されている．実装上の効率化のため，NewHope では鍵生成，暗号化の際に離散ガウス分布の替わりに中心を 0 とした二項分布を用いているが，その分の安全性の低下は [4, Theorem 4.1] で Rényi ダイバージェンスを用いた議論により評価されている．

Ring-LWE 問題の具体的な困難性の評価には，LWE 問題に対する Primal 攻撃, Dual 攻撃双方での，BKZ アルゴリズムを用いた必要ブロックサイズから導き出した CoreSVP 計算量による評価を用いている．

変種: 鍵共有を目的とした USENIX 版 [17]，および reconciliation によるエラー訂正プロセスを省略し簡略化した NewHope-Simple[18] が存在する．

補足情報: NIST PQC 第 2 ラウンドの選定レポート [1, p.16] によると，NewHope と CRYSTALS-Kyber の間で比較が行われた．双方ともに dual LWE 形式の構造を持つ格子上で考え，数論変換を用いた高速化を行うという方針で設計されている．Core-SVP ベースの困難性評価では双方とも同程度の強度であったが，実装時のベンチマークの結果は CRYSTALS-Kyber の方が若干良かった．また，安全性の根拠に用いている問題が Ring-LWE と Module-LWE という違いがあり，パラメータ設定の自由度において不利であったようである．

表 2.9: NewHope CPA-KEM, CCA-KEM のパラメータ [4, Table 2,3]. 2 番目のパラメータは NIST 耐量子計算機暗号 Call for proposal 基準でレベル 5 であると主張されているが，表 [4, Table 3] では 233-bit 安全性となっている．公開鍵，秘密鍵，平文，暗号文サイズの単位はそれぞれ Byte である．

(n, q, k, γ)	安全性 レベル	公開鍵 サイズ	秘密鍵サイズ (鍵カプセル化後)	平文 サイズ	暗号文サイズ (鍵カプセル化後)
$(\ 512, 12289, 8, 10968)$	レベル 1	928	869 (1,888)	32	1,088 (1,120)
$(1024, 12289, 8, \quad 7)$	レベル 5	1,824	1,792 (3,680)	32	2,176 (2,208)

2.3.3 NTRU

歴史: NTRU 暗号方式自体の歴史は長く，1996 年に国際会議 CRYPTO の Rump Session において発表され，その後 1998 年に国際会議 ANTS において発表された論文 [65] が方式の源流となる．本節では，歴史的な NTRU ではなく，NIST PQC 標準化活動の Round 3 Finalist に選定された公開鍵暗号方式 NTRU について説明する．

2017 年 11 月の NIST PQC 公募に提出された 2 件の方式，Andreas Hülsing, Joost Rijneveld, John M. Schanck, Peter Schwabe らにより提案された NTRU-HRSS-KEM[67] と，Cong Chen, Jeffrey Hoffstein, William Whyte, Zhenfei Zhang らにより提案された NTRUEncrypt[36] が Round 2 に進む際にマージされ名称が NTRU と変更，Round 3 にかけてさらに修正が加えられたものが現在の方式となる．Round 2 submission は [67] と [36] の著者 8 名に Oussama Danba を加えた合計 9 名での提案，Round 3 submission はさらに齋藤 恆和，草川 恵太，山川 高志の 3 名を加えた合計 12 名での提案となった．

現在の最新版は 2020 年 9 月に公表された仕様書 [33] である．本節の記述はこの仕様書に従う．

参照 URL: 開発者による公式ページ `https://ntru.org/` を参照した．

設計原理: NTRU は NTRU 格子上の計算困難問題に安全性の根拠を置く公開鍵暗号方式である．具体的には鍵復元攻撃の困難性が NTRU 格子上の短いベクトルを求める問題，平文復元攻撃の困難性が，ターゲットベクトルに近い NTRU 格子上の点を求める問題[*7]として捉えられる．

ベースとなる公開鍵暗号方式は ANTS バージョン [65] の NTRU と比較して，鍵 f, g のサンプリング空間の変更，メッセージ多項式のマスキング手法の変更，暗号化関数の脱乱択化[*8] などの改良が行われている．暗号化関数が決定的であるためベースの方式自体は IND-CPA 安全性を持たないが，この性質を用いることで IND-CCA2 KEM の構成において，単純な藤崎-岡本変換を用いた構成と比較してハッシュ関数の呼び出し回数を削減することが可能である．

提案ではベースとなる公開鍵暗号方式の OW-CPA 安全性を格子問題の困難性に還元した後に，齋藤ら [100] において提案された implicit rejection の導入による NTRU-HRSS-KEM の改良の構成をベースとして，デカプセル化時の再暗号化処理のスキップによる IND-CCA2 KEM の構成を行っている．最終的な方式の IND-CCA2 安全性は適当な仮定を置くことで ROM, QROM モデルにおいて元の方式の OW-CPA 安全性に帰着される．ただし標準的な仮定に

[*7] 最短・最近ベクトル問題や限界距離復号問題と似ているが，設定が少し異なるため，このように表現する．

[*8] 一般に，脱乱択化 (derandomizing) とはアルゴリズム中で乱数を用いるサブルーチンを，ほぼ同等の性能を保ったままに乱数を用いない決定的なサブルーチンに置き換えることを示す．元々の NTRU では暗号化関数内で平文をマスキングする多項式をランダムに生成していたが，提案バージョンでは多項式を関数の引数とし関数自体は決定的なものとなっている．

おいては QROM モデルでの還元はタイトではなく，いくつかの non-standard な仮定を置くことでタイトになる [5, p.39].

アルゴリズムの詳細:

以下，表 2.11，2.12，2.13 に ANTS 版の NTRU と NIST PQC Round 3 提案版の NTRU の公開鍵暗号方式を並置して解説する．

パブリックパラメータは以下で与えられる．仕様書 [33] には NTRU-HPS と NTRU-HRSS の 2 系統のパラメータセットが存在し，それぞれ $\boldsymbol{f}, \boldsymbol{g}$ のサンプリング空間，Lift 関数の構成などが異なる．

- n を定義多項式の次数，$\Phi_1 = x - 1, \Phi_n = (x^n - 1)/(x - 1)$ を円分多項式，p, q を素数の法とする．多項式の次数 n は素数で，$2, 3$ が $\mathbb{Z}_n$ の原始元となるように選ぶ．
- 素数 q と多項式 $F(x)$ に対して，記号 $\mathbb{Z}[x]/(q, F)$ で剰余環 $\mathbb{Z}_q[x]/F(x)$ を表す．特に $R/q, S/q$ はそれぞれ $\mathbb{Z}_q[x]/\Phi_1\Phi_n, \mathbb{Z}_q[x]/\Phi_n$ を定義する．n の取り方より，$S/2, S/3$ は有限体となるため，0 以外の元に常に逆元が存在することになる．
- 集合 $\mathcal{T}$, を係数が $\{0, \pm 1\}$ の多項式で次数が $n - 2$ 以下のものの全体とし，

$$\mathcal{T}' := \left\{ \boldsymbol{v} = \sum_{i=0}^{n-2} v_i x^i \in \mathcal{T} : \sum_{i=0}^{n-3} v_i v_{i+1} \geq 0 \right\}$$

  とする．
- $\mathcal{L}(d_1, d_2)$ は ANTS 版 NTRU のサンプリング空間を定義するために用いられる．次数 $n - 1$ 以下の多項式で d_1 個の係数が $+1$，d_2 個の係数が -1，残りは 0 であるものの集合とする．

暗号アルゴリズムの中で用いられるサブルーチン群は以下で与えられる．

- 多項式 $\boldsymbol{a}$ に対して，関数 $\underline{\mathsf{S3}}(\boldsymbol{a})$ を $\boldsymbol{b} \equiv \boldsymbol{a} \bmod (3, \Phi_n)$ を満たすもので，次数 $n - 2$ 以下かつ係数が $\{0, \pm 1\}$ となるものとする．これを $S/3$ の代表元とする．
- $\mathsf{Lift}(\boldsymbol{m})$ 関数は，メッセージ $\boldsymbol{m}$ のマスキングに用いられる．暗号文を $\boldsymbol{c} = \boldsymbol{r} \cdot \boldsymbol{h} + \boldsymbol{m}$ にって計算する NTRU 系の暗号の場合，$\mathcal{L}_f, \mathcal{L}_g$ の取り方によっては IND-CPA 安全性を持たない可能性がある [33, p.22]．そのため，メッセージ多項式 m を一度別の形にマスキングする必要がある．NTRUEncrypt[36] では，ランダム多項式 t を足しこむことで実現していたのだが，NTRU-HRSS[67] ではこの機能を Lift を用いて実現している．Lift 関数は $\underline{\mathsf{S3}}$ 関数を用いて実現され高速実装が可能であり，しかもマスキングのための多項式 t をサンプリングするという手間がなくなるため，より実装が単純・高速となる．そのため，Round 2 のマージにおいて NTRU-HRSS のアイディアが残った形である．

以下の表 2.10 に鍵多項式 f, g，暗号化に用いるランダム多項式 r，平文多項式 m の空間と Lift 関数の違いをまとめる．$\underline{\mathsf{S3}}(\boldsymbol{m}/\Phi_1)$ の高速計算法は文献 [66, Append. B] に掲載されている．

ANTS 版 NTRU では Lift 関数は明示されていないが，[33, Sect. 1.3.1] によると $\underline{\mathsf{S3}}(\mathsf{Lift}(\boldsymbol{m})) = \boldsymbol{m}$ を満たす単射 $\mathcal{L}_m \to \mathbb{Z}[x]$ として解釈できる．

NTRU の鍵生成関数（2.11）の詳細を記述する．最初に Sample_fg 関数により $(\boldsymbol{f}, \boldsymbol{g})$ を $\mathcal{L}_f \times \mathcal{L}_g$ から一様ランダムにサンプリングする．ANTS 版では $\mathcal{L}_f$ からランダムにサンプリングを行い，$\bmod (2, \Phi_1\Phi_n)$, $\bmod (3, \Phi_1\Phi_n)$ の双方で可逆であることを確認し，可逆でない場合にはサンプリングをやり直していた．一方で，NTRU-HPS,NTRU-HRSS では構成から可逆性が保証されているため，可逆性検査は行わない．

秘密鍵の中に $\boldsymbol{h}_q$ が含まれるのは復号関数の中で暗号化に用いた多項式 $\boldsymbol{r}$ を復元するためである．

表 2.10: 方式ごとのサンプリング空間，Lift 関数の違い

	ANTS 版 NTRU [33, Sect. 1.3.1]	NTRU-HPS [33, Sect. 1.3.2]	NTRU-HRSS [33, Sect. 1.3.3]
$\mathcal{L}_f$	$\mathcal{L}(d_f, d_f - 1)$	$\mathcal{T}$	$\mathcal{T}_+$
$\mathcal{L}_g$	$\mathcal{L}(d_f, d_f - 1)$	$\mathcal{T}(q/8 - 2)$	$\{\Phi_1 \cdot \boldsymbol{v} : \boldsymbol{v} \in \mathcal{T}_+\}$
$\mathcal{L}_r$	$\{p \cdot \phi : \phi \in \mathcal{L}(d, d)\}$	$\mathcal{T}$	$\mathcal{T}$
$\mathcal{L}_m$	係数が $[-p/2, p/2]$ に含まれる多項式の集合	$\mathcal{T}(q/8 - 2)$	$\mathcal{T}$
$\mathsf{Lift}(\boldsymbol{m})$ 関数	-	$\underline{\mathsf{S3}}(\boldsymbol{m})$	$\Phi_1 \cdot \underline{\mathsf{S3}}(\boldsymbol{m}/\Phi_1)$

表 2.11: ANTS 版 NTRU および NIST PQC 版 NTRU における鍵生成関数の比較

	ANTS NTRU [65, Sect. 1.2] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	NIST PQC NTRU [33, Figure 9] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
1:	$\boldsymbol{f} \leftarrow \mathsf{Sample\_f}()$	
	// $\Phi_1\Phi_n$ の中で可逆な元をサンプリングする	
	$\boldsymbol{g} \leftarrow \mathsf{Sample\_g}()$	$(\boldsymbol{f}, \boldsymbol{g}) \leftarrow \mathsf{Sample\_fg}()$
		$\boldsymbol{f}_q \leftarrow (1/\boldsymbol{f}) \bmod (q, \Phi_n)$
2:	$\boldsymbol{h} \leftarrow (3\boldsymbol{g}/\boldsymbol{f}) \bmod (q, \Phi_1\Phi_n)$	$\boldsymbol{h} \leftarrow (3\boldsymbol{g} \cdot \boldsymbol{f}_q) \bmod (q, \Phi_1\Phi_n)$
		$\boldsymbol{h}_q \leftarrow (1/\boldsymbol{h}) \bmod (q, \Phi_n)$
3:	$\boldsymbol{f}_p \leftarrow (1/\boldsymbol{f}) \bmod (3, \Phi_1\Phi_n)$	$\boldsymbol{f}_p \leftarrow (1/\boldsymbol{f}) \bmod (3, \Phi_n)$
return	$pk = \boldsymbol{h}, sk = (\boldsymbol{f}, \boldsymbol{f}_p)$	$pk = \boldsymbol{h}, sk = (\boldsymbol{f}, \boldsymbol{f}_p, \boldsymbol{h}_q)$

表 2.12: ANTS 版 NTRU および NIST PQC 版 NTRU における暗号化関数の比較

	ANTS NTRU [65, Sect. 1.3] $\mathsf{Enc}(pk = h, \boldsymbol{m} \in \mathcal{L}_m) \to \boldsymbol{c}$	NIST PQC NTRU [33, Figure 9] $\mathsf{Enc}(pk = h, \boldsymbol{m} \in \mathcal{L}_m; \boldsymbol{r}) \to \boldsymbol{c}$
1:	$\boldsymbol{r} \leftarrow \mathsf{Sample\_r}()$	
2:	$\boldsymbol{c} \leftarrow (\boldsymbol{r} \cdot \boldsymbol{h} + \boldsymbol{m}) \bmod (q, \Phi_1\Phi_n)$	$\boldsymbol{m}' \leftarrow \mathsf{Lift}(\boldsymbol{m})$
		$\boldsymbol{c} \leftarrow (\boldsymbol{r} \cdot \boldsymbol{h} + \boldsymbol{m}') \bmod (q, \Phi_1\Phi_n)$
return	$\boldsymbol{c}$	$\boldsymbol{c}$

表 2.12 に暗号化関数を記述する．この部分はオリジナルの ANTS 版 NTRU とほぼ同様であるが，暗号化に用いるランダム多項式 $\boldsymbol{r}$ が関数の入力として明示されている点，平文多項式の表現を Lift 関数によって変えている点が異なる．

ANTS 版 NTRU では暗号化時に乱数として生成された $\boldsymbol{r}$ を平文の一部として扱うことで，暗号化関数が決定的なものとなる．これにより，IND-CCA2 KEM を構成する際のタイトな還元を実現している [100].

表 2.13 の復号関数の説明を行う．暗号化関数が脱乱択化されたことで，復号関数は ANTS 版のものとは大きく異なるものになる．出力が平文の $(\boldsymbol{r}, \boldsymbol{m})$ の他に，復号が失敗したかどうかを示すフラグ flag を返す．このフラグがデカプセル化時の implicit rejection に用いられる．

暗号化時の $\mathcal{L}_g$ の取り方，Lift 関数の性質から正しく作られた暗号文の場合ステップ 0 の $\boldsymbol{c} \equiv 0 \ (q, \Phi_1)$ が保証され

表 2.13: ANTS 版 NTRU および NIST PQC 版 NTRU における復号関数の比較

	ANTS NTRU [65, Sect. 1.4] $\mathsf{Dec}(sk = (\boldsymbol{f}, \boldsymbol{f}_p), \boldsymbol{c}) \to \boldsymbol{m}'$	NIST PQC NTRU [33, Figure 9] $\mathsf{Dec}(sk = (\boldsymbol{f}, \boldsymbol{f}_p, \boldsymbol{h}_q), \boldsymbol{c}) \to (\boldsymbol{r}, \boldsymbol{m}, \mathsf{flag})$
0:		**if** $\boldsymbol{c} \not\equiv 0 \pmod{(q, \Phi_1)}$ **return** $(0, 0, 1)$
1:	$\boldsymbol{a} \leftarrow (\boldsymbol{c} \cdot \boldsymbol{f}) \bmod (q, \Phi_1\Phi_n)$	$\boldsymbol{a} \leftarrow (\boldsymbol{c} \cdot \boldsymbol{f}) \bmod (q, \Phi_1\Phi_n)$
2:	$\boldsymbol{m}' \leftarrow (\boldsymbol{a} \cdot \boldsymbol{f}_p) \bmod (3, \Phi_1\Phi_n)$	$\boldsymbol{m} \leftarrow (\boldsymbol{a} \cdot \boldsymbol{f}_p) \bmod (3, \Phi_n)$
		$\boldsymbol{m}' \leftarrow \mathsf{Lift}(\boldsymbol{m})$
		$\boldsymbol{r} \leftarrow ((\boldsymbol{c} - \boldsymbol{m}') \cdot \boldsymbol{h}_q) \bmod (q, \Phi_n)$
		if $(\boldsymbol{r}, \boldsymbol{m}) \in \mathcal{L}_r \times \mathcal{L}_m$ **return** $(\boldsymbol{r}, \boldsymbol{m}, 0)$
		else return $(0, 0, 1)$
return	$\boldsymbol{m}'$	$(\boldsymbol{r}, \boldsymbol{m}, \mathsf{flag})$

る．$\mathsf{flag} = 0$ の場合にこの等式が保証されることは，デカプセル化関数内での再暗号化スキップのために必要である．ステップ 2 で復元された $\boldsymbol{m}$ が $S/3$ の代表元となっていることは保証できないため，Lift 関数を用いて $\boldsymbol{m}'$ の復元を行い，それを用いて $\boldsymbol{r}$ を復元する．$(\boldsymbol{r}, \boldsymbol{m})$ が正常な平文空間 $\mathcal{L}_r \times \mathcal{L}_m$ に含まれているならば $\mathsf{flag} = 0$ をセットして復号結果を返し，そうでなければ失敗として $\boldsymbol{r} = \boldsymbol{m} = 0$ とし，失敗フラグを立てて値を返す．

安全性とパラメータ: ベースとなる公開鍵暗号方式の OW-CPA 安全性の具体的な困難性を評価するために鍵復元攻撃と平文復元攻撃が考えられている．公開鍵から秘密鍵を復元する問題は，環の定義多項式と公開鍵多項式によって定義される格子内において，秘密鍵 $(\boldsymbol{f}, \boldsymbol{g})$ に対応する短いベクトル[*9]を発見する問題として捉えることができる．また，暗号文から平文を復元する問題も $\boldsymbol{h}_q$ から定義される格子内で，$(0, \boldsymbol{c})$ に近いベクトルを探索する問題として捉えられるため，最近ベクトル問題の埋め込みによりこちらも格子内の短いベクトルを求める問題として定式化することが可能である．

以上により，暗号方式のパラメータ設定には与えられた格子内の短いベクトルを BKZ アルゴリズムを用いて発見するために必要なブロックサイズ β を求め，具体的な計算量は Core-SVP による評価を行っている．

Core-SVP の計算量評価には，篩アルゴリズムで用いる巨大なメモリ空間にアクセスするためのコストを定数と仮定した non-local model およびそうでないと仮定した local model の双方を用いた個別の評価 [33, Sect 5.3] を行っている．

秘密鍵サイズの括弧内は KEM のもので，32Bytes の乱数列 s の分大きくなる．

2.3.4 SABER

歴史: SABER は NIST PQC 公募への応募方式の一つ 2017 年 11 月に Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren の 4 名により公表され，その後同著者により 2018 年 5 月に国際会議 AFRICACRYPT において査読付き国際会議論文として発表された [42].

NIST PQC 第 2 ラウンド提出時において安全性証明に関わる微修正が行われ，第 3 ラウンドからは新たに Andrea Basso, Jose Maria Bermudo Mera, Michiel Van Beirendonck の 3 名が加わり，開発者は合計 7 名となっている．現

[*9] 多項式環や格子の構成等の違いにより秘密鍵と最短ベクトルが対応しない，最短ベクトルが複数存在するなどの状況があるため，短いベクトルという表現としている．

表 2.14: NTRU のパラメータ [33, Sect. 1.6, 3.2]. 公開鍵，秘密鍵，平文，暗号文サイズの単位はそれぞれ Byte である．

パラメータ名	(n, p, q)	安全性レベル		公開鍵	秘密鍵	平文	暗号文
		non-local	local	サイズ	サイズ	サイズ	サイズ
ntruhps2048509	$(509, 3, 2048)$	-	レベル 1	699	903(935)	204	699
ntruhps2048677	$(677, 3, 2048)$	レベル 1	レベル 3	930	1,202(1,234)	272	930
ntruhps4096821	$(821, 3, 4096)$	レベル 3	レベル 5	1,230	1,558(1,590)	328	1,230
ntruhrss701	$(701, 3, 8192)$	レベル 1	レベル 3	1,138	1,418(1,450)	280	1,138

在の最新版は Round 3 Finalist に提出された [29] であり，以下の記述はこの仕様書に従う．

参照 URL: 開発者による公式ページ `https://www.esat.kuleuven.be/cosic/pqcrypto/saber/` およびリファレンス実装 `https://github.com/KULeuven-COSIC/SABER` を参照した．

設計原理: SABER は Module-LWR 問題を安全性の根拠とする公開鍵暗号方式であり，LWE 暗号におけるノイズ付加計算をラウンディング演算に置き換えた暗号方式を構成のひな型としている．基本となる方式に対して Module 化と実装上の改良のための修正を行い IND-CPA 安全な暗号方式を構成，藤崎-岡本変換により IND-CCA2 KEM としたものである．仕様書の設計原理 [29, Sect. 4] の項には Regev[89] 暗号の “LWR version” であると記述されている一方で，Second PQC Standardization Conference の発表スライド [39, p.5] では Lindner-Peikert[73] 型（dual-LWE 型）の構成を原型としている．数式の比較から，どちらも原型と捉えることが可能であり，本報告書では仕様書に従い LWE 型に分類する．

LWR 型暗号方式の利点として，LWE 暗号の実装時に必要とされる離散ガウス分布等からのサンプリング計算の回避が挙げられる．また，処理を行う際の法 p, q を 2 のべき乗とすることでラウンディング処理がビット列の部分的なコピーのみで完了すること，同様に鍵となる行列 $A \in R_q^{\ell \times \ell}$ の生成が乱数生成ルーチンからのビット列のコピーのみで完了することから，高速処理が可能であるという特徴がある．

一方で，2 のべき乗の形で p, q をとることにより，数論変換を用いた多項式同士の乗算が単純に適用できなくなるという欠点があるが仕様書 [29, Sect.4] によると，SABER で用いられる多項式は 256 次であり，通常の乗算方法を用いても NTT によるものと処理時間に大きな差は無いと主張されている．また，[35] のように一度大きな剰余空間で乗算を計算した後に $\mathbb{Z}_p, \mathbb{Z}_q$ の世界に引き戻す実装テクニックも開発されており，多項式の乗算による効率の低下に関しての大きな問題は無いと考えられる．

IND-CPA 安全な公開鍵暗号から IND-CCA2 KEM の構成には Hofheinz ら [62] による藤崎-岡本変換の変種を用いており，ROM,QROM モデルの双方で安全であることが示されている [29, Sect. 6]．公開鍵暗号から KEM の具体的な構成は [29, Algorithm 4-6] 参照.

アルゴリズムの詳細: 表 2.15，2.16，2.17 に Regev 暗号の LWR 版と SABER（IND-CPA 安全な基本バージョン）の鍵生成，暗号化，復号アルゴリズムを並置する．

パブリックパラメータは以下で与えられる．

- n: 環を定義するための多項式 $x^n + 1$ の次数であり，全てのパラメータセットで $n = 256$ とする．
- p, q, T: ラウンディングの大きさを決定するパラメータ．全て 2 のべき乗の形で，$\epsilon_q = \log_2(q), \epsilon_p =$

$\log_2(p), \epsilon_T = \log_2(T)$ とし，$\epsilon_q > \epsilon_p > \epsilon_T$ とする．つまり $T|p|q$ の関係がある．計算を行う環は $R_q := \mathbb{Z}_q[x]/(x^n+1)$ とし，R_q の元を成分とする．

- l: モジュール格子のランクである．ベクトル，行列をそれぞれ $R_q^{l\times 1}, R_q^{l\times l}$ 等で表現する．
- 平文空間は $R_2 := \mathbb{Z}_2[x]/(x^n+1)$ であり，256bits の情報を格納できる．

表 2.15: Regev 暗号の LWR 版および SABER における鍵生成関数の比較

	Regev 暗号の LWR 版 [39, p.7] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	SABER[29, Algorithm 1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
1:	A: ランダム行列	$\mathsf{seed}_A \xleftarrow{\$} \{0,1\}^{256}$; $A \leftarrow \mathsf{gen}(\mathsf{seed}_A) \in R_q^{l\times l}$
2:	$\boldsymbol{s}$: 短いランダムベクトル	$\mathsf{r} \xleftarrow{\$} \{0,1\}^{256}$; $\boldsymbol{s} \leftarrow \beta_\mu(R_q^{l\times 1}; \mathsf{r})$
3:	$\boldsymbol{b} = \lfloor A\boldsymbol{s} \rceil_{p/q}$	$\boldsymbol{b} = ((A^T\boldsymbol{s} + \boldsymbol{h}) \bmod q) \gg (\epsilon_q - \epsilon_p) \in R_p^{l\times 1}$
return	$pk = (A, \boldsymbol{b}), sk = \boldsymbol{s}$	$pk = (\mathsf{seed}_A, \boldsymbol{b}), sk = \boldsymbol{s}$

表 2.15 の左側は Regev による LWE 暗号 [89] におけるノイズ付加関数 $A\boldsymbol{s}+\boldsymbol{e}$ をラウンディング関数 $\lfloor A\boldsymbol{s} \rceil_{p/q}$ へと置き換えたものである．ただし，実数 x に対して $\lfloor x \rceil_{p/q} := \lfloor x \cdot (p/q) \rceil$ とし，ベクトル，行列に対しては成分ごとにその操作を行うものとする．

表 2.15 の右側，SABER の鍵生成関数を説明する．ステップ 1 の gen 関数は 256bits の列を種として，R_q を成分とした擬似ランダムな $l \times l$ 行列を生成する．各成分は R_q 内の一様分布とする．

ステップ 2 の $\beta_\mu(R_q^{l\times 1}; r)$ はビット列 r を種として R_q 成分 l 次元の擬似ランダムベクトルを出力する関数である．ここで，μ は偶数であるとし，各成分の多項式は係数を独立に，パラメータ μ の二項分布の出力から平均値 $\mu/2$ を引いた値をサンプリングしたものとする．

ステップ 3 の公開鍵ベクトル $\boldsymbol{b}$ の生成は $A^T\boldsymbol{s}$ のラウンディングによるものだが，p, q がともに 2 のべき乗 $\varepsilon_p, \varepsilon_q$ であることから p/q を掛けた後のラウンディング処理が

$$\lfloor x \rceil_{p/q} := \left\lfloor \frac{p}{q} x \right\rceil = \left\lfloor \frac{p}{q} x + \frac{1}{2} \right\rfloor = \left\lfloor \frac{p}{q} \left(x + \frac{q}{2p} \right) \right\rfloor = \left\lfloor \left(x + 2^{\varepsilon_q - \varepsilon_p - 1} \right) 2^{\varepsilon_p - \varepsilon_q} \right\rfloor \tag{2.1}$$

と表現され，入力 x に定数 $h = 2^{\varepsilon_q - \varepsilon_p - 1}$ を加えた後，$2^{\varepsilon_p - \varepsilon_q}$ による乗算と切り捨て処理が $(\varepsilon_q - \varepsilon_p)$ ビットの右シフトで実現される．表中の $\boldsymbol{h}$ は，係数が全て h の多項式を成分として持つ $R_q^{l\times 1}$ のベクトルを示す．

出力される公開鍵の形式はデータ量削減のため $(A, \boldsymbol{b})$ の替わりに，行列 A を生成するためのシードを用いて $pk = (\mathsf{seed}_A, \boldsymbol{b})$ としている．

表 2.16 の右側，SABER の暗号化関数を説明する．平文空間は $R_2 = \mathbb{Z}_2[x]/(x^n+1)$ で，$n = 256$bits の情報を格納する．最初にステップ 0 として，暗号化処理に用いる行列 A を seed_A から復元する．次にステップ 1 では暗号化用のランダムベクトル $\boldsymbol{s}'$ を種となるビット列 r を用いて生成するが，与えられていなかった場合には 256bits の乱数列を擬似乱数生成器によって生成する．Enc 関数に種となるビット列が与えられているのは後にこの関数が藤崎-岡本変換を用いた IND-CCA KEM の構成に使われるためである．$\boldsymbol{s}'$ を生成する関数 β_μ，およびステップ 2 での $A\boldsymbol{s}'$ のラウンディングによる $\boldsymbol{b}'$ の生成は鍵生成と同様である．ステップ 3 では $\boldsymbol{b}, \boldsymbol{s}'$ を用いて暗号化を行うが，$(...) \gg (\epsilon_p - \epsilon_T)$ の計算は (2.1) 式における p/q の役割を T/p に置き換えたものである．$2^{\epsilon_p - 1}m$ は係数が 0 もしくは $2^{\epsilon_p - 1}$ の多項式 $(\in R_p)$ となるため，復号関数内で $v' + h_1$ のラウンディングを鍵を用いて小さいノイズに変換することで平文が復元可能となる．

表 2.16: Regev 暗号の LWR 版および SABER における暗号化関数の比較

	Regev 暗号の LWR 版 [39, p.7] $\mathsf{Enc}(pk=(A,b), m\in\{0,1\})\to \mathsf{ct}$	SABER[29, Algorithm 2] $\mathsf{Enc}(pk=(\mathsf{seed}_A,b), m\in R_2; r)\to \mathsf{ct}$
0:		$A\leftarrow \mathsf{gen}(\mathsf{seed}_A)\in R_q^{l\times l}$ //行列 A の復元
1:	$\boldsymbol{s}'$: 短いランダムベクトル	$\boldsymbol{s}'\leftarrow \beta_\mu(R_q^{l\times 1}; r)$
2:	$\boldsymbol{b}'=\lfloor A^T\boldsymbol{s}'\rceil_{p,q}$	$\boldsymbol{b}'=((A\boldsymbol{s}'+\boldsymbol{h}) \bmod q)\gg(\epsilon_q-\epsilon_p)\in R_p^{\ell\times 1}$
3:	$c_m=\left\lfloor \boldsymbol{b}^T\boldsymbol{s}'-m\cdot\left\lfloor\frac{p}{2}\right\rceil\right\rceil_{T/p}$	$v'=\boldsymbol{b}^T(\boldsymbol{s}' \bmod p)\in R_p$ $c_m=(v'+h_1-2^{\epsilon_p-1}m \bmod p)\gg(\epsilon_p-\epsilon_T)\in R_T$
return	$\mathsf{ct}=(c_m,\boldsymbol{b}')$	$\mathsf{ct}=(c_m,\boldsymbol{b}')$

表 2.17: Regev 暗号の LWR 版および SABER における復号関数の比較

	Regev 暗号の LWR 版 [39, p.7] $\mathsf{Dec}(sk=\boldsymbol{s}, \mathsf{ct}=(c_m,\boldsymbol{b}'))\to m'$	SABER[29, Algorithm 3] $\mathsf{Dec}(sk=\boldsymbol{s}, \mathsf{ct}=(c_m,\boldsymbol{b}'))\to m'$
1:	$v=(\boldsymbol{b}')^T\boldsymbol{s}$	$v=(\boldsymbol{b}')^T(\boldsymbol{s} \bmod p)\in R_p$
2:	$m'=\left\lfloor\frac{2}{q}\cdot\left(\frac{T}{p}v-c_m\right)\right\rceil$	$m'=((v-2^{\epsilon_p-\epsilon_r}c_m+h_2) \bmod p)\gg(\epsilon_p-1)\in R_2$
return	m'	m'

表 2.17 の右側，SABER の復号関数では (2.1) 式と同様の原理により $\left\lfloor\frac{2}{q}\cdot\left(\frac{T}{p}v-c_m\right)\right\rceil$ が計算される．$h_2\in R_q$ は全ての係数が $2^{\epsilon_p-2}-2^{\epsilon_p-\epsilon_T-1}+2^{\epsilon_q-\epsilon_p-1}$ である多項式とする．

安全性とパラメータ: ベースとなる IND-CPA 安全な公開鍵暗号の安全性は環 $\mathbb{Z}[x]/(x^{256}+1)$ 上の Module-LWR 問題に帰着されることが示されている．Module-LWR 問題の具体的な困難性評価は，Albrecht らによる LWE,NTRU 問題の困難性シミュレータ [12] を LWR 問題向けに修正したものを用いている．

SABER の安全性を決めるパラメータは l, n, q, p, T, μ の 6 個であり，アルゴリズムの詳細で説明したのと同様に，l が Module-LWR 問題のランク，n が多項式の次数であり，大きいほど安全性が上がるがラウンディング時のノイズの蓄積により復号エラー率が上がる．法 q を大きくとることで Module-LWR の格子体積が大きくなり安全性が下がるがラウンディング時のノイズに強くなるため復号エラー率が下がる．ラウンディングパラメータ p, T を大きくとることでラウンディング時のノイズに相当するものが大きくなり，安全性が上がると同時に復号エラー率も上がる．秘密鍵サンプリングの範囲を指定する μ は大きくとることで秘密鍵ベクトルのサンプリング空間が広がり安全性が上がるが，ラウンディング時のノイズが大きくなり復号エラー率が上がる．

秘密鍵サイズは一つ目の数字が圧縮前，括弧内の数字が圧縮後のものを示す．秘密鍵ベクトル $\boldsymbol{s}\in R_q$ の各成分が $[-\mu/2, \mu/2]$ の範囲であることから，$\lceil\log_2 q\rceil$ bits の整数として保存するのではなく，下位 $\lceil\log_2\mu\rceil$ bits のみを保存することで圧縮できる．

2.3.5 CRYSTALS-Kyber

歴史: CRYSTALS-Kyber は NICT PQC 公募への応募方式の一つとして 2017 年 11 月に Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancréde Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor

表 2.18: SABER のパラメータ [29, Table 1]. 公開鍵，秘密鍵，平文，暗号文サイズの単位はそれぞれ Byte である．

(l, n, q, p, T, μ)	安全性レベル	公開鍵暗号			鍵カプセル化後		
		公開鍵サイズ	秘密鍵サイズ	暗号文サイズ	公開鍵サイズ	秘密鍵サイズ	暗号文サイズ
$(2, 256, 2^{13}, 2^{10}, 2^3, 10)$	レベル 1	672	832(256)	736	672	1,568(992)	736
$(3, 256, 2^{13}, 2^{10}, 2^4, 8)$	レベル 3	992	1,248(288)	1,088	992	2,304(1,344)	1,088
$(4, 256, 2^{13}, 2^{10}, 2^6, 6)$	レベル 5	1,312	1,664(384)	1,472	1,312	3,040(1,760)	1,472

Seiler, Damien Stehlé の 10 名により共同で発表され [7]，その後 2018 年 4 月の国際会議 Euro S&P に Roberto Avanzi を除いた 9 名の共著により査読付き論文として発表された [24]．NIST PQC 標準化の第 3 ラウンドからは Jintai Ding が加わり 11 名での提案となった．NIST の耐量子計算機暗号標準化において唯一暗号化・鍵交換目的での Selected Algorithm として残った方式である [80].

NIST PQC 標準化のラウンドが進むごとに主に暗号化処理のパラメータに関して修正が行われ，現在の最新版は 2021 年 8 月に公開されたバージョン 3.02[9] である．以下の記述はこの仕様書に従う．

参照 URL: 開発者による公式ページ `https://pq-crystals.org/kyber/` および GitHub のリファレンスコード `https://github.com/pq-crystals/kyber` を参照した．

設計原理: CRYSTALS-Kyber は Module-LWE 問題を安全性の根拠とする暗号化方式であり，dual-LWE 暗号方式をひな型[*10]として $x^{256}+1$ を定義多項式とした環上で処理を行うことで効率化している．

ベースとして IND-CPA 安全な公開鍵暗号を構成し，それを藤崎-岡本変換のデカプセル化失敗時の戻り値を調整した Hofheinz らの変種 [62] により IND-CCA2 安全な KEM へと変換している．

アルゴリズムの詳細: 表 2.20，2.21，2.22 に Lindner-Peikert[73] による格子ベース公開鍵暗号と CRYSTALS-Kyber の鍵生成，暗号化，復号アルゴリズムを並置する．

パブリックパラメータは以下で与えられる．

- n, q: 環を定義するための多項式 x^n+1 の次数と法を示す．用いられる多項式環は $R := \mathbb{Z}[x]/(x^n+1), R_q := \mathbb{Z}_q[x]/(x^n+1)$ であり，常に $n = 256, q = 3329 = 13 \cdot 256 + 1$ と固定されている[*11].
- k: モジュール格子のランクとする．
- η_1, η_2: 鍵生成および暗号化時に生成するノイズベクトルの大きさを指定する．
- d_u, d_v: 暗号文多項式 $(\boldsymbol{u}, v)$ を表現するためのビット数を指定する．

用いられるサブルーチンのうち主なものを以下に列挙する．

- $\mathsf{NTT}(f)$ は $f = \sum_{i=0}^{255} f_i x^i \in R_q$ の NTT 表現 $\hat{f} = \sum_{i=0}^{255} \hat{f}_i x^i \in R_q$ を求める関数で，

$$\hat{f}_{2i} = \sum_{j=0}^{127} f_{2j} \zeta^{(2\mathsf{br}_7(i)+1)j} \quad \text{および} \quad \hat{f}_{2i+1} = \sum_{j=0}^{127} f_{2j+1} \zeta^{(2\mathsf{br}_7(i)+1)j}$$

[*10] 仕様書では，アルゴリズムの形が Lyubashevsky-Peikert-Rosen の Ring-LWE ベース暗号 [74] に似ているとしている．

[*11] NIST PQC 提出時には $q = 7681$ であったが，第 2 ラウンドからはこの値に変更された．

により定義される．ただし，$\mathsf{br}_7(i)$ は 7bits の整数を引数にとり，そのビット反転を出力する関数である．$\zeta = 17$ は $\mathbb{Z}_q$ における原始元である．

この表記は複数の R_q の元を並べたベクトル $\boldsymbol{s} = (s_0, s_1, \ldots, s_{k-1}) \in R_q^k$ にも有効で，$\mathsf{NTT}(\boldsymbol{s}) = (\mathsf{NTT}(s_0), \mathsf{NTT}(s_1), \ldots, \mathsf{NTT}(s_{k-1}))$ 等と解釈する．

- $\mathsf{Parse}(\mathsf{XOF}(\rho, i, j))$: XOF (extendable output function) を用いてシードの ρ, i, j から十分な長さの擬似乱数列を生成し，それを Parse 関数により R_q の元に変換する．
- $\mathsf{CBD}_\eta(\mathsf{PRF}(\sigma, i))$: 擬似乱数生成器 PRF は長さ 32Bytes の σ と 1Byte の i をシードとして 512ηbits の擬似乱数列 $\beta_0\beta_1\cdots\beta_{512\eta-1}$ へと変換する．この列を 2ηbits ごとに切り分け $i = 0, \ldots, 255$ に対して $f_i = \sum_{j=0}^{\eta-1} \beta_{i\cdot 2\eta+j} - \sum_{j=0}^{\eta-1} \beta_{i\cdot 2\eta+\eta+j}$ を計算．i 次の係数を f_i とした 255 次多項式を CBD_η 関数の出力とする．
- $\mathsf{Encode}_\ell(\hat{\boldsymbol{s}}), \mathsf{Decode}_\ell(\boldsymbol{b})$: Encode_ℓ 関数は 255 次の多項式 $\hat{\boldsymbol{s}} \in R_q$ を入力とし，各係数を ℓbits のビット列に直したものを結合した 256ℓbits のビット列を出力とする．Decode 関数はその逆を行う関数で，ビット列を多項式環の元に変換する．
- $\mathsf{Compress}_q(x, d), \mathsf{Decompress}_q(x, d)$: $x \in \mathbb{Z}_q$ を近似的に $d$bits に変換，逆変換を行う関数であり，暗号文のサイズ削減に用いられる．具体的には

$$\begin{aligned} \mathsf{Compress}_q(x, d) &:= \lceil (2^d/q) \cdot x \rfloor \bmod 2^d, \text{および} \\ \mathsf{Decompress}_q(x, d) &:= \lceil (q/2^d) \cdot x \rfloor \end{aligned}$$

で定義される．

擬似乱数生成器の実装について: アルゴリズムの仕様の中で用いられる擬似乱数生成器 XOF, PRF, G, H, KDF について，元々の SHAKE ハッシュ関数などを用いたものに加え，NIST PQC 第 2 ラウンドに合わせてアップデートされたバージョン 2.0[8] からは “90s version” として AES と SHA のみを用いたものが提案されている．これらの関数がデファクトスタンダードとして既に多くのハードウェア上で実装されている事から，高速化を狙ったものである．以下の表 2.19 に用いられる関数をまとめる．なお，本節で紹介する IND-CPA 安全な方式の中では XOF, PRF および G のみが用いられ，他の 2 つは IND-CCA2 安全な方式の構成において呼び出される．

90s version の XOF 関数では，AES-256 の CTR モードを ρ を鍵，12Bytes の nonce を $\text{nonce}[0] = i$, $\text{nonce}[1] = j$, $\text{nonce}[\ell] = 0$ for $\ell = 2, \ldots, 11$ とパディングして用いる．同様に PRF 関数では AES-256 の CTR モードを ρ を鍵，12Bytes の nonce を $\text{nonce}[0] = i$, $\text{nonce}[\ell] = 0$ for $\ell = 1, \ldots, 11$ として用いる．また，オリジナルバージョンの SHAKE-128 の呼び出し方に関してはリファレンス実装[*12] を参照した．

表 2.19: CRYSTALS-Kyber における擬似乱数生成器の実装 [9, Sect. 1.4]

	$\mathsf{XOF}(\rho, i, j)$	$\mathsf{PRF}(\sigma, i)$	$\mathrm{H}(\boldsymbol{b})$	$\mathrm{G}(\boldsymbol{b})$	$\mathsf{KDF}(\boldsymbol{b})$
オリジナル	SHAKE-128$(\rho\|\|i\|\|j)$	SHAKE-256$(\sigma\|\|i)$	SHA3-256$(\boldsymbol{b})$	SHA3-512$(\boldsymbol{b})$	SHAKE-256$(\boldsymbol{b})$
90s	AES-256	AES3-256	SHA-256$(\boldsymbol{b})$	SHA-512$(\boldsymbol{b})$	SHA-256$(\boldsymbol{b})$

CRYSTALS-Kyber の鍵生成関数（表 2.20 右）を説明する．表の中で $\mathcal{B}$ は 1Byte 分の情報を表す集合 $\{0, 1, \ldots, 255\}$ を表す．ランダムに生成した 32Bytes の d をシードとして，ハッシュ関数 G を用いて 32Bytes の擬似ランダムビットの組 (ρ, σ) を生成する．これらはそれぞれ，行列 $A \in R_q^{k\times k}$ とノイズ多項式 $\boldsymbol{s}, \boldsymbol{e} \in R_q^k$ をサンプリングするためのシー

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

表 2.20: Lindner-Peikert 格子ベース暗号および CRYSTALS-Kyber における鍵生成関数の比較

	Lindner-Peikert [73, Sect. 3.1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	CRYSTALS-Kyber [9, Algorithm 4] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
0:		$d \xleftarrow{\$} \mathcal{B}^{32}$ $(\rho, \sigma) \leftarrow \mathrm{G}(d)$
1:	A: $n_1 \times n_2$ ランダム行列	$\hat{A}[i][j] \leftarrow \mathsf{Parse}(\mathsf{XOF}(\rho, j, i))$ for $i = 0, \ldots, k-1$ and $j = 0, \ldots, k-1$
2:	S: 成分の小さい $n_2 \times \ell$ 行列	$\boldsymbol{s}[i] \leftarrow \mathsf{CBD}_{\eta_1}(\mathsf{PRF}(\sigma, i))$ for $i = 0, \ldots, k-1$ $\hat{\boldsymbol{s}} \leftarrow \mathsf{NTT}(\boldsymbol{s})$
3:	E: 成分の小さい $n_1 \times \ell$ 行列	$\boldsymbol{e}[i] \leftarrow \mathsf{CBD}_{\eta_1}(\mathsf{PRF}(\sigma, i+k))$ for $i = 0, \ldots, k-1$ $\hat{\boldsymbol{e}} \leftarrow \mathsf{NTT}(\boldsymbol{e})$
4:	$B = AS + E$	$\hat{\boldsymbol{t}} \leftarrow \hat{A} \circ \hat{\boldsymbol{s}} + \hat{\boldsymbol{e}}$
return	$pk = (A, B), sk = S$	$pk = (\mathsf{Encode}_{12}(\hat{\boldsymbol{t}} \bmod q) \| \rho), sk = \mathsf{Encode}_{12}(\hat{\boldsymbol{s}} \bmod q)$

ドとして用いられる．通常空間で R_q を一様ランダムにサンプルしたものに NTT をかけた後の分布はまた R_q 内の一様分布となるため，A は最初から NTT 空間でサンプリングされているものとみなされる．

$\boldsymbol{s}, \boldsymbol{e} \in R_q^k$ については CBD_{η_1} を用いて通常空間でのサンプリングを行い，その成分を個別に数論変換する．数論変換の性質により，最後の $\hat{\boldsymbol{t}}$ は $\mathsf{NTT}(A\boldsymbol{s} + \boldsymbol{e})$ となる．公開鍵サイズを圧縮するため，$A, \hat{\boldsymbol{t}}$ をそれぞれシード ρ，Encode 関数による圧縮形式で保存する．秘密鍵の $\hat{\boldsymbol{s}}$ に関しても同様である．

表 2.21: Lindner-Peikert 格子ベース暗号および CRYSTALS-Kyber における暗号化関数の比較

	Lindner-Peikert[73, Sect. 3.1] $\mathsf{Enc}(pk = (A, B), \boldsymbol{m} \in \{0,1\}^\ell) \to \mathsf{ct}$	CRYSTALS-Kyber [9, Algorithm 5] $\mathsf{Enc}(pk = (T \| \rho), m \in \mathcal{B}^{32}) \to \mathsf{ct}$
0:		$\hat{\boldsymbol{t}} \leftarrow \mathsf{Decode}_{12}(T)$ $\hat{A}^T[i][j] \leftarrow \mathsf{Parse}(\mathsf{XOF}(\rho, i, j))$ //行列 $\hat{\boldsymbol{A}}$ の転置の形での復元
1:	$\boldsymbol{s}', \boldsymbol{e}', \boldsymbol{e}''$: 成分の小さいベクトル	$\boldsymbol{r}[i] \leftarrow \mathsf{CBD}_{\eta_1}(\mathsf{PRF}(r, i))$ for $i = 0, \ldots, k-1$ $\boldsymbol{e}_1[i] \leftarrow \mathsf{CBD}_{\eta_2}(\mathsf{PRF}(r, i+k))$ for $i = 0, \ldots, k-1$ $e_2 \leftarrow \mathsf{CBD}_{\eta_2}(\mathsf{PRF}(r, 2k))$
2:	$\boldsymbol{u} = \boldsymbol{s}'A + \boldsymbol{e}'$	$\hat{\boldsymbol{r}} \leftarrow \mathsf{NTT}(\boldsymbol{r})$ $\boldsymbol{u} \leftarrow \mathsf{NTT}^{-1}(\hat{\boldsymbol{A}}^T \circ \hat{\boldsymbol{r}}) + \boldsymbol{e}_1$ $v \leftarrow \mathsf{NTT}^{-1}(\hat{\boldsymbol{t}}^T \circ \hat{\boldsymbol{r}}) + e_2 + \mathsf{Decompress}_q(\mathsf{Decode}_1(m), 1)$ $c_1 \leftarrow \mathsf{Encode}_{d_u}(\mathsf{Compress}_q(\boldsymbol{u}, d_u))$ $c_2 \leftarrow \mathsf{Encode}_{d_v}(\mathsf{Compress}_q(v, d_v))$
return	$\mathsf{ct} = (\boldsymbol{u}, \boldsymbol{v})$	$\mathsf{ct} = (c_1 \| c_2)$

CRYSTALS-Kyber の暗号化関数（表 2.21 右）を説明する．圧縮形で入力された公開鍵から $\hat{\boldsymbol{t}}, \hat{A}$ を復元する．このとき，処理のために行列は転置された形で復元される．

暗号化のため成分の小さい $\boldsymbol{r}, \boldsymbol{e}_1 \in R_q^k$ と $e_2 \in R_q$ をサンプリングする．通常空間と NTT 空間を使い分けて処理を効率化しているが，最終的な暗号文 $c_1 \| c_2$ は通常空間でのベクトル $\boldsymbol{u} \in R_q^k$ と多項式 $v \in R_q$ を $\mathsf{Compress}_q$ 関数で圧縮

したものとなる．ここで，2 種類のノイズ η_1, η_2 を使い分けるのは，η_1 のみによるノイズの大きさと，最後の Encode 関数によるラウンディングからの決定的ノイズと η_2 のノイズを合成したものの大きさが釣り合うように調整するためである [9, Sect. 1.5]．

表 2.22: Lindner-Peikert 格子ベース暗号および CRYSTALS-Kyber における復号関数の比較

	Lindner-Peikert[73, Sect. 3.1] $\mathsf{Dec}(sk, \mathsf{ct}) \to \boldsymbol{m}'$	CRYSTALS-Kyber [9, Algorithm 6] $\mathsf{Dec}(sk, \mathsf{ct} = (c_1 \| \| c_2)) \to m' \in \mathcal{B}^{32}$
1:	$\overline{\boldsymbol{m}} = \boldsymbol{v} - \boldsymbol{u}S$ $m'_i = \begin{cases} 0 & \lvert\overline{m}_i\rvert \le \lfloor q/4 \rfloor \\ 1 & \text{それ以外} \end{cases}$	$\boldsymbol{u} \leftarrow \mathsf{Decompress}_q(\mathsf{Decode}_{d_u}(c_1), d_u)$ $v \leftarrow \mathsf{Decompress}_q(\mathsf{Decode}_{d_v}(c_2), d_v)$ $\hat{\boldsymbol{s}} \leftarrow \mathsf{Decode}_{12}(sk)$ $m' \leftarrow \mathsf{Encode}_1(\mathsf{Compress}_q(v - \mathsf{NTT}^{-1}(\hat{\boldsymbol{s}}^T \circ \mathsf{NTT}(\boldsymbol{u})), 1))$
return	$\boldsymbol{m}' = (m'_1, \ldots, m'_\ell)$	m'

CRYSTALS-Kyber の復号関数（表 2.22 右）は圧縮されたビット列の展開, NTT 空間の利用などで表現が煩雑になっているが，Lindner-Peikert 暗号の復号処理と本質的に同様である．最後の $\mathsf{Compress}_q(\cdot, 1)$ 関数が Lindner-Peikert 暗号における $\overline{\boldsymbol{m}}$ から $\boldsymbol{m}'$ への変換に対応している．

安全性とパラメータ: ベースとなる IND-CPA 安全な公開鍵暗号の安全性は多項式環 $R_q := \mathbb{Z}_q[x]/(x^n + 1)$ 上の判定版 Module LWE 問題へと ROM，QROM モデルの下で帰着される．

パラメータの設定は Module LWE 問題を構造の無い LWE 問題とみなし Primal, Dual の双方の攻撃を BKZ アルゴリズムを用いて解いた場合の必要ブロックサイズに対応する Core SVP 計算量を通じて行われている．Module LWE 問題へと帰着する際に，二項分布によるノイズと $\mathsf{Compress}_q$ 関数の四捨五入によるノイズを総合して詳細な解析を行っている．また，パラメータ設定用のスクリプトは [46] で公開されている．

暗号の性能を決めるパラメータは $n, k, q, \eta_1, \eta_2, d_u, d_v$ の 7 個であり，大まかに以下の特徴を持つ．格子の次元は多項式の次数 n と Module LWE 問題のランク k の積であり，これらのパラメータを大きくとることで暗号の安全性が上がるが処理速度が低下し，鍵と暗号文のサイズが膨らむ．法 q を大きくとることでノイズ耐性が上がり復号エラー率が下がるが，格子が疎になり暗号の安全性が低下する．

(η_1, η_2) は鍵生成と暗号化に用いられるノイズ多項式の大きさで，大きくとることで暗号の安全性が上がるが復号エラー率が下がる．また，ノイズの中心二項分布を生成する際に必要とされるランダムビットの長さが増える．

(d_u, d_v) は暗号文 $(\boldsymbol{u}, v)$ をビット列で表現するための精度を指定する．小さくとることで暗号文サイズが削減できるが，桁落ちが発生し復号エラー率が上がる．また，これらの値を小さくとることは暗号文にノイズを与えることになり，安全性が僅かではあるが向上するが，復号エラー率への影響の方が大きい．

以下の表で δ は CCA2-KEM における復号エラー率を示す．正しい暗号文が KEM のデカプセル化 [9, Algorithm 9] で非受理となる確率である．

2.3.6 CRYSTALS-Dilithium

歴史: CRYSTALS-Dilithium は 2017 年 6 月に Cryptology ePrint Archive において Léo Ducas，Tancrède Lepoint，Vadim Lyubashevsky，Peter Schwabe，Gregor Seiler，Damien Stehlé の 6 名の連名で公表 [43] され，そ

表 2.23: CRYSTALS-Kyber のパラメータ [9, Table 1] および [5, Sect. D]. 公開鍵，秘密鍵，平文，暗号文サイズの単位はそれぞれ Byte である．

(n, k, q)	(η_1, η_2)	(d_u, d_v)	安全性 レベル	公開鍵 サイズ	秘密鍵 サイズ	平文 サイズ	暗号文 サイズ	復号 エラー率 δ
$(256, 2, 3329)$	$(3, 2)$	$(10, 4)$	レベル 1	800	1,632	32	768	2^{-139}
$(256, 3, 3329)$	$(2, 2)$	$(10, 4)$	レベル 3	1,184	2,400	32	1,088	2^{-164}
$(256, 4, 3329)$	$(2, 2)$	$(11, 5)$	レベル 5	1,568	3,168	32	1,568	2^{-174}

の後論文内での予告通りに 2017 年 11 月に NIST PQC 公募への応募方式 [40] として Eike Kiltz を加えた 7 名を開発者として提出された．査読付き論文としては国際会議 CHES 2018 において公開された版 [41] が存在する．

NIST PQC　標準化のラウンドが進むごとに微修正が行われ，現在の最新版は 2021 年 2 月に公開された仕様書 V3.1[23] である．本節の記述はこの仕様書に従う．

参照 URL: 開発者による公式ページ `https://pq-crystals.org/dilithium/` を参照した．

設計原理: CRYSTALS-Dilithium は格子ベースの署名方式であり，Lyubashevsky[77] による Fiat-Shamir with Aborts 型の構成を行っている．秘密鍵復元問題の安全性の根拠を，$x^{256} + 1$ を定義多項式とした環上における Module-LWE 問題に，署名の強偽造不可能性の根拠を SelfTargetMSIS 問題に置いている．通信コストを下げるため，公開鍵サイズと署名サイズの和の最小化を目的としてパラメータの設計を行っている．

最新の実装では，署名の検証にかかる計算時間の 80% はハッシュ関数 Keccak の処理時間であり，速度的にはこれ以上改良できない限界であるとしている [79].

アルゴリズムの詳細: 表 2.24，2.25，2.27 に Lyubashevsky による Fiat-Shamir with Aborts 型の格子ベース署名，CRYSTALS-Dilithium のテンプレートアルゴリズム [23, Fig. 1] および実装のための擬似コード [23, Fig.4] を並置して記述する．

パブリックパラメータは以下で与えられる．

- n, q: 環を定義するための多項式 $x^n + 1$ の次数と法を示す．用いられる多項式環は $R := \mathbb{Z}[x]/(x^n + 1)$, $R_q := \mathbb{Z}_q[x]/(x^n + 1)$ であり，提案方式の中では常に $n = 256, q = 2^{23} - 2^{13} + 1 = 8380417$ を用いる．
- k: モジュール格子のランクとする．
- l: ハッシュの (R_q における) 次元パラメータとする．
- d: 鍵生成時に $\boldsymbol{t}$ から分離する下位ビットの長さ
- η: 秘密鍵ベクトルのサンプリング空間の大きさ．
- τ: 署名生成時のベクトル $\boldsymbol{c}$ のサンプリング空間の大きさ．$\beta := \eta \cdot \tau$
- γ_1: 署名生成用ベクトル $\boldsymbol{y}$ のサンプリング空間の大きさ．
- γ_2: 署名生成用ベクトル $\boldsymbol{w}$ から取り出す上位ビットの長さ．

用いられるサブルーチンのうち主なものを以下に列挙する．

- $\mathsf{NTT}(\mathsf{a})$ は $\boldsymbol{a} = \sum_{i=0}^{255} a_i x^i$ の NTT 表現 $\hat{a} \in \mathbb{Z}_q^{256}$ を求める関数で，

$$\hat{a} = (a(r_0), a(-r_0), a(r_1), a(-r_1), \ldots, a(r_{127}), a(-r_{127}))$$

で計算される．ただし，$r = 1753$, $r_i = r^{\mathrm{brv}(128+i)} \bmod q$, $\mathrm{brv}(k)$ 関数は k を 8bits の 2 進数としてみたときのビット反転．[23, Sect. 2.2]

- H: ビット列の伸長のためのハッシュ関数．CRYSTALS-Dilithium の実装では SHAKE256 ハッシュ関数を用いる．
- $\mathsf{ExpandA}(\rho)$: 乱数生成のシード ρ を用いて，ランダム行列 $A \in R_q^{k \times l}$ を生成し，その NTT 表現

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,l} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,l} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k,1} & a_{k,2} & \cdots & a_{k,l} \end{bmatrix} \to \hat{A} = \begin{bmatrix} \mathsf{NTT}(a_{1,1}) & \mathsf{NTT}(a_{1,2}) & \cdots & \mathsf{NTT}(a_{1,l}) \\ \mathsf{NTT}(a_{2,1}) & \mathsf{NTT}(a_{2,2}) & \cdots & \mathsf{NTT}(a_{2,l}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{NTT}(a_{k,1}) & \mathsf{NTT}(a_{k,2}) & \cdots & \mathsf{NTT}(a_{k,l}) \end{bmatrix}$$

を計算し出力する．

- $\mathsf{ExpandS}(\rho')$: 署名に用いる多項式 $\boldsymbol{s}_1, \boldsymbol{s}_2$ を生成するための関数で，512bits のシードを入力とする．
- $\mathsf{Power2Round}_q(t,d), \mathsf{HighBits}_q(t,\alpha), \mathsf{LowBits}_q(t,\alpha)$: $\mathbb{Z}_q$ の元 t で，$0 \le t < q$ を満たすものを $t = r_1 \cdot 2^d + r_0, -q/2 < r \le q/2$ と分解したときに $\mathsf{Power2Round}_q(t,d) = (r_1, r_0)$ と定義する．$\mathbb{Z}_q$ の多項式 $t \in R_q$, R_q 成分のベクトル $\boldsymbol{t}$ に対しても成分ごとに同様の操作を行うものとして定義する．具体的には，$\boldsymbol{t} = \left(\sum t_{j,i} x^i\right)_{j=1,\ldots,k}$ と書いたときに $\mathsf{Power2Round}_q(t_{j,i}, d) \to (t_{j,i,1}, t_{j,i,0})$ とすれば，$\mathsf{Power2Round}_q(\boldsymbol{t}, d) \to (\boldsymbol{t}_1, \boldsymbol{t}_0)$ は $\boldsymbol{t}_1 = \left(\sum q_{j,i} x^i\right)_{j=1,\ldots,k}$, $\boldsymbol{t}_0 = \left(\sum r_{j,i} x^i\right)_{j=1,\ldots,k}$, ただし $t_{j,i} = q_{j,i} \cdot 2^d + r_{j,i}$ と定義したものである．
また，α を $q-1$ の約数としたとき，同様に整数 t を $t = r_1 \cdot \alpha + r_0, -q/2 < r_0 \le q/2$ の形で分解し，$\mathsf{HighBits}_q(t,\alpha), \mathsf{LowBits}_q(t,\alpha)$ をそれぞれ r_1, r_0 で定義する．
- $\mathsf{MakeHint}_q(z,r,\alpha)$, $\mathsf{UseHint}_q(h,r,\alpha)$: $\mathsf{MakeHint}_q$ 関数は $\mathsf{HighBits}_q(r,\alpha) \neq \mathsf{HighBits}_q(r+z,\alpha)$ であれば 1 を，そうでなければ 0 を返す関数である．$\mathsf{UseHint}_q$ 関数は引数から $\mathsf{HighBits}_q(r+z,\alpha)$ を復元する関数である．復元成功の十分条件は [23, Lemma 4] で与えられている．
- $\mathsf{SampleInBall}(\tilde{c})$ 関数は係数のうち τ 個が ± 1 で，それ以外が 0 である多項式の集合 B_τ から一様サンプリングを行う．τ はパブリックパラメータとして与えられており，引数の $\tilde{c}$ はサンプリングのシードとして用いられる．生成された多項式 $c \in R$ の NTT 表現 $\hat{c} = \mathsf{NTT}(c)$ が出力される．
- $\#_1 \boldsymbol{h}$ は $\boldsymbol{h} = \sum_{i=0}^{255} h_i$ の中で $h_i = 1$ となる項の数を表す．

表 2.24 の鍵生成関数について記述する．256bits のシード ζ をハッシュ関数 H により合計 1024bits に伸長し，そのうち ρ, ρ' をそれぞれ公開鍵 A のシード，秘密鍵 $\boldsymbol{s}_1, \boldsymbol{s}_2$ のシードとして用いる．鍵サイズ圧縮のため，行列 A はシード ρ の形で表現され，必要に応じて展開される．秘密鍵 $\boldsymbol{s}_1, \boldsymbol{s}_2$ は R の元をそれぞれ ℓ, k 個並べたベクトルであり，各成分は集合 $S_\eta = \{\boldsymbol{w} \in R : \|\boldsymbol{w}\|_\infty \le \eta\}$ から一様ランダムにサンプリングされる．

Fiat-Shamir 型署名における秘密鍵 $\hat{\boldsymbol{s}}$ のハッシュ関数 $a(\hat{\boldsymbol{s}})$ の計算が，ベクトル $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ と行列 A を用いた $A\boldsymbol{s}_1 + \boldsymbol{s}_2$ の計算に対応している．

計算されたベクトル $\boldsymbol{t} \in R_q^k$ に対して，$\mathsf{Power2Round}_q$ 関数により上位ビットと下位ビットに分割する．

最後に，メッセージに連結するためのランダムビット tr をハッシュ関数 H を用いて生成する．

表 2.25 の署名生成関数について記述する．ρ から行列 A の NTT 表現 $\hat{A}$ を復元する．ランダムビット tr を用いてメッセージのハッシュ値 μ を計算し，この値に署名をつける．κ は ExpandMask 関数の中で呼び出す SHAKE256 の

表 2.24: CRYSTALS-Dilithium における鍵生成関数の比較

	格子ベース署名 [77, Fig. 4] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	CRYSTALS-Dilithium テンプレート [23, Fig. 1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	CRYSTALS-Dilithium 実装のための擬似コード [23, Fig. 4] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$
1:	$\hat{\boldsymbol{s}}$: 短い多項式を 成分とするベクトル	$\boldsymbol{s}_1 \leftarrow S_\eta^l$, $\boldsymbol{s}_2 \leftarrow S_\eta^k$	$\zeta \xleftarrow{\$} \{0,1\}^{256}$ $\mathsf{H}(\zeta) \to (\rho, \rho', K) \in \{0,1\}^{256} \times \{0,1\}^{512} \times \{0,1\}^{256}$ $\mathsf{ExpandS}(\rho') \to (\boldsymbol{s}_1, \boldsymbol{s}_2) \in S_\eta^\ell \times S_\eta^k$
2:	a: ハッシュ関数	$\hat{A} \xleftarrow{\$} R_q^{k \times l}$	$\mathsf{ExpandA}(\rho) \to \hat{A} \in R_q^{k \times \ell}$
3:	$t \leftarrow a(\hat{\boldsymbol{s}})$	$\boldsymbol{t} = A\boldsymbol{s}_1 + \boldsymbol{s}_2$	$\boldsymbol{t} \leftarrow \mathsf{NTT}^{-1}(\hat{A} \cdot \mathsf{NTT}(\boldsymbol{s}_1)) = A\boldsymbol{s}_1 + \boldsymbol{s}_2$ $\mathsf{Power2Round_q}(\boldsymbol{t}, d) \quad \to (\boldsymbol{t}_1, \boldsymbol{t}_0)$ $\mathsf{H}(\rho \| \boldsymbol{t}_1) \to tr \in \{0,1\}^{256}$
return	$sk = (a, \hat{\boldsymbol{s}}), pk = (a, \boldsymbol{t})$	$sk = (A, \boldsymbol{t}, s_1, s_2)$, $pk = (A, \boldsymbol{t})$	$sk = (\rho, K, tr, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0)$, $pk = (\rho, \boldsymbol{t}_1)$

シードとなる値で，$\mathsf{H}(K||\mu) \to \rho'$ とともに用いられる．計算効率化の目的で R_q の元の乗算には NTT 表現を用いるため，予め $\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0$ を NTT 表現に変換しておく．

Fiat-Shamir 型署名の標準的な構成方法と同様に，署名の初期値 $(\boldsymbol{z}, \boldsymbol{h})$ を $\perp$ とし，**while** ループの中で生成された署名が集合 G に含まれているかどうかを検査し含まれていない場合にはループをやり直す．

ExpandMask 関数の中では，(ρ', κ) をシードとしてランダムベクトル $\boldsymbol{y} \in R_q^l$ をサンプリングする．ここで，各成分は $\tilde{S}_{\gamma_1} = \left\{ \sum_{i=0}^{255} w_i x^i : -\eta < w_i \leq \eta \right\}$ から一様ランダムにサンプリングされる．このサンプリングは表 2.25 中央の $\boldsymbol{y} \leftarrow D_{\gamma_1}^{l \times 1}$ に対応する．

署名生成のためのベクトル $c \in R_q^l$ は 256bits のシード $\tilde{c}$ により表現され，この値自体は μ と $\boldsymbol{w}_1$ を連結したハッシュ値から計算される．ここで，μ はメッセージからの要素であり，$\boldsymbol{w}_1$ は公開鍵 A と直前でサンプリングした $\boldsymbol{y}$ から来る要素である．計算効率のため，内積 $c \cdot \boldsymbol{s}_1$ は NTT 表現で計算された後に逆変換をかけ $\boldsymbol{z} = \boldsymbol{y} + c \cdot \boldsymbol{s}_1$ となる．

ステップ 5 では $z \notin G$ のチェックのため，$\boldsymbol{z}$ と $\boldsymbol{w} - c\boldsymbol{s}_2$ の下位ビットの ℓ_∞ ノルムがそれぞれ比較される．両方が閾値よりも小さい場合には次のヒント生成関数 $(*)$ が実行される．ヒント生成関数は表 2.26 により示され，$\mathsf{MakeHint}_q$ 実行後に再びノルムの大きさがチェックされ，閾値よりも大きな場合には $(\boldsymbol{z}, \boldsymbol{h}) \leftarrow \perp$ となる．つまり，2 回の **if** 文の中での 4 回の不等号検査のうち一つでも満たされない条件があれば，シード κ を増やし $\boldsymbol{y}$ の生成からやり直すことになる．ここで，$-c\boldsymbol{s}_2 + c\boldsymbol{t}_0$ の計算は前半を $\boldsymbol{r}_0$ で用いたものを使いまわし，後半を $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\boldsymbol{t}}_0)$ の形で計算する．

表 2.27 の署名検証関数について記述する．公開鍵，署名に含まれる乱数のシード $\rho, \tilde{c}$ から $\hat{A}, \hat{c}$ を復元し，メッセージに対応するハッシュ値 μ を計算する．$A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d$ は $\hat{A} \cdot \mathsf{NTT}(\boldsymbol{z}) - \mathsf{NTT}(c) \cdot \mathsf{NTT}(\boldsymbol{t}_1 \cdot 2^d)$ の形で計算する．これらの値から $\mathsf{UseHint}_q$ を用いて $\boldsymbol{w}_1'$ を復元し，$\boldsymbol{z}$ のノルム，$\boldsymbol{h}$ の 1 の数の確認を行い，正しければ accept を出力する．

安全性とパラメータ: CRYSTALS-Dilithium の安全性は，$x^n + 1$ を定義多項式とする環上のモジュール格子問題である．ROM の下で，秘密鍵復元の困難性が Module-LWE 問題に，署名の強偽造不可能性が SelfTargetMSIS 問題にそれぞれ帰着される．SelfTargetMSIS 問題は Module-SIS 問題の変種ではあるが，タイトではないものの帰着が知られているため，Module-SIS 問題を安全性の根拠と考えることもできる．

一方で，QROM においても鍵復元，署名偽造が同様に Module-LWE 問題，SelfTargetMSIS 問題それぞれ帰着されるものの Module-SIS 問題までの量子帰着が知られていない．

具体的なパラメータは，LWE 問題と SIS 問題の双方に対して BKZ アルゴリズムで解いた際の必要ブロックサイズと Core-SVP の見積から求められている．

表 2.25: CRYSTALS-Dilithium における署名生成関数の比較

	格子ベース署名 [77, Fig. 4] $\mathsf{Sign}(sk=(a,\hat{\boldsymbol{s}}),$ $\mu\in\{0,1\}^*)\to\sigma$	CRYSTALS-Dilithium テンプレート [23, Fig. 1] $\mathsf{Sign}(sk=(A,t,\boldsymbol{s}_1,\boldsymbol{s}_2),$ $\mu\in\{0,1\}^*)\to\sigma$	CRYSTALS-Dilithium 実装のための擬似コード [23, Fig. 4] $\mathsf{Sign}(sk=(\rho,K,tr,\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{t}_0),$ $M\in\{0,1\}^*\})\to\sigma$
0:			$\mathsf{ExpandA}(\rho)\to\hat{A}$ $\mathsf{H}(tr\|\|M)\to\mu\in\{0,1\}^{512}$
1:	$z\leftarrow\perp$	$z\leftarrow\perp$	$\kappa\leftarrow 0,(\boldsymbol{z},\boldsymbol{h})\leftarrow\perp$ $\mathsf{H}(K\|\|\mu)\to\rho'\in\{0,1\}^{512}$ $\hat{\boldsymbol{s}}_1\leftarrow\mathsf{NTT}(\boldsymbol{s}_1);\ \hat{\boldsymbol{s}}_2\leftarrow\mathsf{NTT}(\boldsymbol{s}_2)$ $\hat{\boldsymbol{t}}_0\leftarrow\mathsf{NTT}(\boldsymbol{t}_0)$
2:	**while** $z=\perp$ **do**	**while** $z=\perp$ **do**	**while** $(\boldsymbol{z},\boldsymbol{h})=\perp$ **do**
3:	$\hat{\boldsymbol{y}}$: 短い多項式を 成分とするベクトル	$\boldsymbol{y}\leftarrow D^{l\times 1}_{\gamma_1-1}$	$\mathsf{ExpandMask}(\rho',\kappa)\to\boldsymbol{y}\in\tilde{S}^l_{\gamma_1}$
4:	$c\leftarrow H(a(\hat{\boldsymbol{y}})\|\|\mu)$	$\boldsymbol{w}_1\leftarrow\mathsf{HighBits}(A\boldsymbol{y},2\gamma_2)$ $c=H(\mu\|\|\boldsymbol{w}_1)$	$\boldsymbol{w}\leftarrow\mathsf{NTT}^{-1}(\hat{A}\cdot\mathsf{NTT}(\boldsymbol{y}))=\boldsymbol{Ay}$ $\boldsymbol{w}_1\leftarrow\mathsf{HighBits}_q(\boldsymbol{w},2\gamma_2)$ $H(\mu\|\|\boldsymbol{w}_1)\to\tilde{c}\in\{0,1\}^{256}$ $\mathsf{SampleInBall}(\tilde{c})\to\hat{c}\in B_\tau\subset R_q$
5:	$\hat{\boldsymbol{z}}\leftarrow\hat{\boldsymbol{y}}+c\hat{\boldsymbol{s}}$ if $\hat{\boldsymbol{z}}\notin G^m$ **then** $z\leftarrow\perp$	$\boldsymbol{z}\leftarrow\boldsymbol{y}+c\boldsymbol{s}_1$ $\boldsymbol{r}_0\leftarrow\mathsf{LowBits}(A\boldsymbol{y}-c\boldsymbol{s}2,2\gamma_2)$ **if** $(\\|\boldsymbol{z}\\|_\infty\geq\gamma_1-\beta)$ OR $(\\|\boldsymbol{r}_0\\|_\infty\geq\gamma_2-\beta)$ **then** $z\leftarrow\perp$	$\boldsymbol{z}\leftarrow\boldsymbol{y}+\mathsf{NTT}^{-1}(\hat{c}\cdot\hat{\boldsymbol{s}}_1)$ $\boldsymbol{r}_0\leftarrow\mathsf{LowBits}_q(\boldsymbol{w}-\mathsf{NTT}^{-1}(\hat{c}\cdot\hat{\boldsymbol{s}}_2),2\gamma_2)$ **if** $(\\|\boldsymbol{z}\\|_\infty\geq\gamma_1-\beta)$ OR $(\\|\boldsymbol{r}_0\\|_\infty\geq\gamma_2-\beta)$ **then** $(\boldsymbol{z},\boldsymbol{h})\leftarrow\perp$ **else** $\boldsymbol{h}\leftarrow\mathsf{MakeHint}_q(\cdot)$ ... (*) $\kappa\leftarrow\kappa+l$
return	$\sigma=(\hat{\boldsymbol{z}},c)$	$\sigma=(\boldsymbol{z},c)$	$\sigma=(\boldsymbol{z},\boldsymbol{h},\tilde{c})$

表 2.26: 署名生成関数におけるヒント生成時のチェック関数

$\boldsymbol{h}\leftarrow\mathsf{MakeHint}_q(-c\boldsymbol{t}_0,\boldsymbol{w}-c\boldsymbol{s}_2+c\boldsymbol{t}_0,2\gamma_2)$ **if** $\\|c\boldsymbol{t}_0\\|_\infty\geq\gamma_2$ OR $\#_1\boldsymbol{h}>\omega$ **then** $(\boldsymbol{z},\boldsymbol{h})\leftarrow\perp$

仕様書に掲載されたパラメータセットを表 2.28 に示す．セキュリティ強度を規定するパラメータのうち，問題が定義される環とモジュールのランクに関わるものが (n,k,l,q) の 4 個，ノイズに関わるものが $(\eta,\gamma_1,\gamma_2,\beta,\tau,d)$ の 6 個である．

変種: [23, Table 3] には NIST の提唱する安全性レベル 1 よりも弱いパラメータ，安全性レベル 5 よりも強いパラメータが掲載されている．

補足情報: NIST PQC の第 3 ラウンド報告レポートにおいて署名方式 FALCON との比較が行われ，CRYSTALS-Dilithium はそのシンプルさから一般的な実装に向いているが，FALCON は署名の短さからリソースの制限されたデバイスで使われることが期待されている．[5, p.19].

ハッシュ関数 H の長さについて，V3.0 までは 384bits であったが衝突耐性を考えると 256-bit 安全性を持たず，安全性レベル 5 の要件を満たさないことが判明したため最新の V3.1 では 512 bits に修正されている [94, p.5].

表 2.27: CRYSTALS-Dilithium における署名検証関数の比較

	格子ベース署名 [77, Fig. 4]	CRYSTALS-Dilithium テンプレート [23, Fig. 1]	CRYSTALS-Dilithium 実装のための擬似コード [23, Fig. 4]
	$\mathsf{Vrfy}(pk=(a,\boldsymbol{t}),\mu\in\{0,1\}^*,$ $\sigma=(\hat{\boldsymbol{z}},c))$	$\mathsf{Vrfy}(pk=(A,\boldsymbol{t}),\mu\in\{0,1\}^*,$ $\sigma=(\boldsymbol{z},c))$	$\mathsf{Vrfy}(pk=(\rho,\boldsymbol{t}_1),M\in\{0,1\}^*,$ $\sigma=(\boldsymbol{z},\boldsymbol{h},\tilde{c}))$
0:			$\mathsf{ExpandA}(\rho)\to\hat{A}$ $\mathsf{H}(\mathsf{H}(\rho\|\|\boldsymbol{t}_1)\|\|M)\to\mu\in\{0,1\}^{512}$ $\mathsf{SampleInBall}(\tilde{c})\to\hat{c}$
1:	**if** $\hat{\boldsymbol{z}}\in G^m$ AND $c=H(a(\hat{\boldsymbol{z}})-\boldsymbol{t}c,\mu)$ **then** accept **else** $\perp$	$\boldsymbol{w}_1'=\mathsf{HighBits}(A\boldsymbol{z}-c\boldsymbol{t},2\gamma_2)$ **if** $\|\boldsymbol{z}\|_\infty<\gamma_1-\beta$ AND $c=H(M\|\|\boldsymbol{w}_1')$ **then** accept **else** $\perp$	$\boldsymbol{w}_1'\leftarrow\mathsf{UseHint}_q(\boldsymbol{h},A\boldsymbol{z}-c\boldsymbol{t}_1\cdot 2^d,2\gamma_2)$ **if** $\|\boldsymbol{z}\|_\infty<\gamma_1-\beta$ AND $\tilde{c}=H(\mu\|\|\boldsymbol{w}_1')$ AND $\#_1\boldsymbol{h}\le\omega$ **then** accept **else** $\perp$

表 2.28: CRYSTALS-Dilithium 署名方式のパラメータ [23, Table 1] , [5, Table 8]．公開鍵，秘密鍵，署名サイズの単位はそれぞれ Byte である．

(n,k,l,q)	$(\eta,\gamma_1,\gamma_2,\beta,\tau,d)$	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ
$(256,4,4,8380417)$	$(2,2^{17},\ 95232,\ 78,49,13)$	レベル 2	$1,312$	$2,528$	$2,420$
$(256,6,5,8380417)$	$(4,2^{19},261888,196,49,13)$	レベル 3	$1,952$	$4,000$	$3,293$
$(256,8,7,8380417)$	$(2,2^{19},261888,120,60,13)$	レベル 5	$2,592$	$4,864$	$4,595$

注： 秘密鍵サイズは仕様書 [23] には掲載されていないが，NIST の第 3 ラウンド報告レポート [5] を参照した．

2.3.7 FALCON

歴史: FALCON は 2017 年 11 月の NIST PQC 公募に Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang の 10 名を開発者として公表された [53]．その後修正が加えられ，現在の最新版は 2020 年 10 月に公開された v1.2[54] である．以下の記述はこの仕様書に従う．

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

設計原理: FALCON は多項式 $x^n+1, n=2^k$ により定義される NTRU 格子上の SIS 問題の困難性を安全性の根拠とした格子ベースの署名方式であり，形式的には Gentry ら [60] の Hash-and-Sign 型の格子ベース署名をひな型としている．高速フーリエサンプリングを用いるため，定義多項式の次数を 2^k の形としていることからパラメータ選択の自由度に制限があり，NIST PQC の提案方式では安全性レベル 1 および 5 のパラメータセットのみが提案されている．

アルゴリズムの詳細: 表 2.29，2.30，2.31 に，Gentry ら [60] の Hash-and-Sign 型格子ベース署名と FALCON の鍵生成，署名生成，署名検証関数を並置する．

パブリックパラメータは以下で与えられる．

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

アルゴリズム中で用いられるサブルーチンのうち，主なものを列挙する．

- $\mathsf{FFT}(f), \mathsf{invFFT}(\boldsymbol{s})$: 多項式 $f \in \mathbb{R}[x]/\phi$ に対して，そのフーリエ変換 $\mathsf{FFT}(f)$ を n 次元ベクトル $(f(\zeta_k))_{k=0,\dots,n-1}$ で定義する．ただし，$\zeta_k := \exp((2k+1)\pi i/n)$．逆演算を $\mathsf{invFFT} : \mathbb{R}^n \to \mathbb{R}[x]/\phi$ で示す．変換，逆変換ともに標準的な高速フーリエ変換の手法が利用可能である．コンピュータ上での計算には浮動小数点演算を用いるため，実行環境ごとに差が出ないように IEEE754 で規定される浮動小数点の表現と演算を用いることが指定されている．
  多項式を成分とするベクトル，行列に対しても FFT は成分ごとのフーリエ変換と定義し，invFFT も適切な切り分けにより実数成分の行列，ベクトルから多項式成分の行列，ベクトルへ変換するものとする．
  また，演算 $\mathsf{FFT}(f) \odot \mathsf{FFT}(g)$ を成分ごとの積と定義する．FFT 表現での多項式の積 $\mathsf{FFT}(fg)$ の計算に対応する．
- $\mathsf{HashToPoint}(\mathsf{str}, q, n)$: ビット列 str を多項式 $c \in \mathbb{Z}_q[x]/\phi$ に SHAKE256 ハッシュ関数を用いて写像する．
- Compress, Decompress: 多項式 $s \in \mathbb{Z}[x]$ を文字列に変換する関数とその逆関数とする．

表 2.29: Hash-and-Sign 型格子ベース署名および FALCON における鍵生成関数の比較

	Gentry らの格子ベース署名 [60, Sect. 7.1] $\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$	FALCON[54, Algorithm 4] $\mathsf{KeyGen}(\phi, q) \to (pk, sk)$
1:	$BA \equiv 0 \pmod q$ を満たす 行列の組 (A, B) を生成 B: 成分の小さい行列 A: ランダム行列	$f, g, F, G \leftarrow \mathsf{NTRUGen}(\phi, q)$ $B \leftarrow \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$ $\hat{B} \leftarrow \mathsf{FFT}(B)$ $G \leftarrow \hat{B} \times \hat{B}^*$ $T \leftarrow \mathsf{ffLDL}^*(G)$ **for** each leaf leaf of T **do** 　$\mathsf{leaf.value} \leftarrow \sigma/\sqrt{\mathsf{leaf.value}}$ $h \leftarrow gf^{-1} \bmod q$
retrun	$pk = A, sk = B$	$pk = h, sk = (\hat{B}, T)$

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

$$fG - gF = q \bmod \phi \tag{2.2}$$

と書くことができる．この関係式と公開鍵 $h = f^{-1}g$ を Hash-and-Sign フレームワーク [60] における行列 A, B と捉えると，

$$A = \begin{bmatrix} 1 \\ h \end{bmatrix},\ B = \begin{bmatrix} g & -f \\ G & -F \end{bmatrix} \tag{2.3}$$

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

また，署名の生成には $\boldsymbol{s}A \equiv H(m)$ を満たす短いベクトル $\boldsymbol{s}$ を生成する必要があり，効率化のため Ducas-Prest[44] の高速フーリエサンプリングを用いる．サンプリングアルゴリズムに必要な情報が B の FFT 表現

$$\mathsf{FFT}(B) = \begin{bmatrix} \mathsf{FFT}(g) & \mathsf{FFT}(-f) \\ \mathsf{FFT}(G) & \mathsf{FFT}(-F) \end{bmatrix} \tag{2.4}$$

およびそれを元にした LDL 木と呼ばれる木構造 T である．木の中には $\hat{B}$ のグラム行列 $G = \hat{B} \times \hat{B}^{\star}$ の[*13] LDL 分解における L の情報が格納され，それを用いて Babai の最近平面アルゴリズムの高速化および離散ガウス分布の高速なサンプリングが可能となる．サンプリングを行うための付加情報として，木の全ての葉にある値を leaf.value から $\sigma/\sqrt{\mathsf{leaf.value}}$ に書き換えることで鍵生成が完了する．

表 2.30: Hash-and-Sign 型格子ベース署名および FALCON における署名生成関数の比較

	Gentry らの格子ベース署名 [60, Sect. 7.1] $\mathsf{Sign}(sk = (\hat{B}, T), m \in \{0,1\}^*) \to \sigma$	FALCON[54, Algorithm 10] $\mathsf{Sign}(sk = (\hat{B}, T), m \in \{0,1\}^*, \lfloor \beta^2 \rfloor) \to \sigma$
1:	$\boldsymbol{c} \leftarrow H(m)$ //平文のハッシュ値をベクトル化	$r \leftarrow \{0,1\}^{320}$ $c \leftarrow \mathsf{HashToPoint}(r \| m, q, n)$ $\hat{\boldsymbol{t}} \leftarrow \left(-\frac{1}{q}\mathsf{FFT}(c) \odot \mathsf{FFT}(F), \frac{1}{q}\mathsf{FFT}(c) \odot \mathsf{FFT}(f) \right)$
2:	T を使い，$\boldsymbol{s}A \equiv \boldsymbol{c} \pmod q$ を 満たすベクトル $\boldsymbol{s}$ をサンプリング	**do** 　**do** 　　$\boldsymbol{z} \leftarrow \mathsf{ffSampling}_n(\hat{\boldsymbol{t}}, T)$ 　　$\hat{\boldsymbol{s}} \leftarrow (\hat{\boldsymbol{t}} - \hat{\boldsymbol{z}})\hat{B}$ 　**while** $\|\boldsymbol{s}\|^2 > \lfloor \beta^2 \rfloor$ 　$(s_1, s_2) \leftarrow \mathsf{invFFT}(\hat{\boldsymbol{s}})$ 　$\mathsf{s} \leftarrow \mathsf{Compress}(s_2, 8 \cdot \mathsf{sbytelen} - 328)$ **while** $(\mathsf{s} = \perp)$
return	$\sigma = \boldsymbol{s}$	$\sigma = (r, \mathsf{s})$

表 2.30 の署名生成関数の説明を記述する．平文にランダムビット r を結合した後，HashToPoint 関数で多項式 $c \in \mathbb{Z}_q/\phi$ を出力する．関係式 (2.2), (2.3) より，ベクトル $\hat{\boldsymbol{t}}$ は $(\mathsf{FFT}(c), \mathsf{FFT}(0))\hat{B}^{-1}$ と等しい事がわかる．これらの情報を用いて，署名ベクトルのサンプリングを行う．

関数 $\mathsf{ffSampling}_n$ は，離散ガウス分布のサンプリングを行い，FFT 表現で出力するサブルーチンである．具体的には，整数ベクトル $\boldsymbol{z} \in \mathbb{Z}^{2n}$ を，$\boldsymbol{t} = [c, 0]B^{-1}$ を中心として $\exp(-\|(\boldsymbol{z} - \boldsymbol{t})B\|^2/2\sigma^2)$ に比例した確率でサンプリングを行う．実装の効率化のため，実際には近似を行っている [54, Sect. 3.9.1,3.9.2]．このとき，$(\boldsymbol{t} - \boldsymbol{z})B$ は原点を中心とした集合

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

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

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

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

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

表 2.31 の署名検証関数の説明を記述する．平文，ハッシュ関数のシード値，署名文字列から各要素を復元し，$s_1 = c - s_2 h$ を計算する．署名が正しく生成されていれば $\boldsymbol{s}A = c$ の関係から，s_1 は短い元となるはずなので，$\|(s_1, s_2)\|^2 \leq \lfloor \beta^2 \rfloor$ が満たされ検証が完了する．

[*13] $B^{\star}$ は体 $\mathbb{Q}[x]/\phi$ におけるエルミート共役．詳細は [54, p.23]

表 2.31: Hash-and-Sign 型格子ベース署名および FALCON における署名検証関数の比較

	Gentry らの格子ベース署名 [60, Sect. 7.1]	FALCON[54, Algorithm 16]
	$\mathsf{Vrfy}(m \in \{0,1\}^*, \sigma = s, pk = A)$	$\mathsf{Vrfy}(m \in \{0,1\}^*, \sigma = (r, \mathsf{s}), pk = h, \lfloor \beta^2 \rfloor)$
1:	$\boldsymbol{t} \leftarrow H(m)$	$c \leftarrow \mathsf{HashToPoint}(r \| m, q, n)$
2:	**if** $\boldsymbol{t} - \boldsymbol{s}A \equiv 0 \pmod q$	$s_2 \leftarrow \mathsf{Decompress}(\mathsf{s}, 8 \cdot \mathsf{sbytelen} - 328)$
	AND $\boldsymbol{s}$ が短い **then return** accept	**if** $(s_2 = \perp)$
		retrun $\perp$
		$s_1 \leftarrow c - s_2 h \bmod q$
		if $\|(s_1, s_2)\|^2 \leq \lfloor \beta^2 \rfloor$
		retrun accept
		else
		retrun $\perp$

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

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

表 2.32: FALCON のパラメータ [54, Table 3.3], [5, Table 8] 公開鍵，秘密鍵，署名サイズの単位はそれぞれ Byte である．

$(n, q, \sigma, \lfloor \beta^2 \rfloor)$	安全性レベル	公開鍵サイズ	秘密鍵サイズ [*14]	署名サイズ
$(\ 512, 12289, 165.736617183, 34034726)$	レベル 1	897	7,553	666
$(1024, 12289, 168.388571447, 70265242)$	レベル 5	1,793	13,953	1,280

変種: 実装の複雑さによるサイドチャネル攻撃からの防御，セキュリティパラメータの多様性確保などを目的とした改良が多数提案されている [52], [38], [48]．特筆すべき点として，格子ベース署名 SOLMAE[69] が韓国の耐量子計算機暗号公募 KpqC へと提出されている．

補足情報: 格子ベースの Hash-and-Sign 署名においてエラーベクトルの圧縮表現などを用いて署名長を短くするテクニックが提案 [50] されており，標準化の際にはこれを盛り込んだ方式に修正される可能性があったが，2022 年 11 月に開催された第 4 回標準化会議におけるアップデート報告 [85] では別の削減方法を用いることが発表された．

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

2.4 格子に基づく暗号技術に関するまとめ

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

この米国 NIST PQC プロジェクトを通じて 2022 年 7 月に CRYSTALS-Kyber が標準的な暗号方式として，CRYSTALS-Dilithium および FALCON が標準的な署名方式として選定された．また，CRYSTALS-Kyber と CRYSTALS-Dilithium は 2022 年 9 月に米国国家安全保障局の Commercial National Security Algorithm Suite 2.0 (CNSA2.0) にも選定されている [81]．NIST PQC 標準化の選考プロセスから今までに漏れた方式の中でも，米国以外の公的機関において推奨暗号とされているものが存在する．一例として，FrodoKEM が 2020 年 8 月よりドイツ情報セキュリティ庁 (BSI) の推奨暗号に [51]，2022 年 1 月にはオランダ通信・安全委員会 (NLNCSA) により最も安全な暗号の例として推奨されている [57]．Google 社の Chrome ブラウザには，TLS レイヤーの性能試験目的で搭載された耐量子計算機暗号プロトコル CECPQ1[31] および CECPQ2[34] にそれぞれ NewHope の USENIX 発表バージョン [17] と NTRU が実装されていたが，2023 年 1 月現在ではともに削除されている．IBM 製テープドライブのプロトタイプとして，CRYSTALS-Kyber と CRYSTALS-Dilithium の組み合わせにより暗号化を行うものが制作されている [70]．DNS サーバの一種である PowerDNS において，耐量子機能を実現する署名として FALCON のテスト用の実装が行われている [61]．オープンソースライブラリへの導入として，WireGuard VPN protocol への SABER の実装 [63]，WolfSSL への CRYSTALS-Kyber，FALCON の実装 [103]，OpenSSH への Streamlined NTRU Prime の実装 [84] などが存在する他，Open Quantum Safe (OQS) プロジェクトによる liboqs ライブラリには暗号化・鍵交換の方式として CRYSTALS-Kyber，NTRU，SABER，FALCON，FrodoKEM，NTRU-Prime が，署名方式として CRYSTALS-Dilithium と FALCON が実装されている [83]．このように格子に基づく暗号技術の社会実装が徐々に進みつつある．

格子に基づく暗号技術の安全性の根拠となる問題としては, 先に挙げた LWE 問題, Ring-LWE 問題, NTRU 問題以外にも Compact LWE 問題, Module-LWE 問題, LWR 問題, BDD 問題, SIS 問題他, 多くのバリエーションが存在している．一般的な格子問題を解く手法としては, LLL アルゴリズム, BKZ アルゴリズムがよく知られており, LWE 問題については更に SIS 問題や BDD 問題に還元する解析手法が知られている．

格子問題の困難性をベースとした暗号方式で最初のものは, Ajtai[20] により 1996 年に行われた, SIS 問題が格子問題の最悪時と同等かそれ以上に困難であることの証明およびそれを用いた暗号学的ハッシュ関数の構成である．また, 1997 年には Ajtai と Dwork[13] により, unique SVP の最悪困難性を安全性の根拠とした公開鍵暗号が提案されている．この公開鍵暗号方式は翌年, Nguyen らによる解読実験 [82] により必要なパラメータが長大となり実用的でないことが明らかにされたものの, その後の格子に基づく暗号構成の基礎となっている．

1996 年に Hoffstein らによって提案された NTRU 暗号 [65][*15] は, 発表当初安全性証明が付けられておらず, 攻撃と修正が繰り返されていたが, 2011 年 Stehlé ら [97] により方式が修正され，イデアル格子上の問題の困難性に還元可能なことが示されている．一方で, 2016 年には subfield attack[6] のような体の構造を使って格子の次元を圧縮する攻撃も提案されており, 暗号の構成のためには次元や法の大きさだけでなく, 環・体の構造にも注意を払う必要がある．

2005 年に Regev[89] により提案された LWE 問題は, 論文発表と同時にそれを暗号の安全性根拠として保障する重要な三つの性質が示された．一つは問題の average-case to worst case reduction, つまりパラメータを固定した際, 問

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

題の (秘密ベクトル $\mathbf{s}$ に関する) 平均的な計算量が, 最悪計算量 (難しいインスタンスを生成するような $\mathbf{s}$ の集合に対する計算量) と高々多項式倍の違いしか無いことであり, 残りの二つは判定 LWE と探索 LWE の等価性, および量子アルゴリズムによる困難な格子問題への還元である. これらの定理を組み合わせることにより, Regev 自身により提案された公開鍵暗号を解読することが平均的に難しいことが示され, その後の様々な LWE ベース暗号の構成の基礎なった. LWE 格子問題への還元に関して, 2013 年には古典計算機による還元も示されている [27].

LWE 問題の欠点である鍵サイズの大きさを改善するため, 2010 年には Lyubashevsky ら [74, 75] により Ring-LWE 問題が, 2015 年には Langlois ら [76] により Module-LWE 問題が暗号化方式と同時に提案され, LWE 問題における関係と類似の, 解読の平均的な困難さが証明されている. 一方で, これらの変種とオリジナルの LWE 問題との関係性は自明ではなく, 同程度の難しさを持つかどうかは未解決問題である. 一般的に Ring(Module)-LWE 問題のインスタンスは LWE 問題のインスタンスとして書きなおすことができるため, LWE 問題は Ring(Module)-LWE 問題よりも困難であるという関係は自明であるが, 逆の関係は知られていない. 法 q が大きい場合には, Ring-LWE は Module-LWE よりも困難であることが知られている [14].

実装時の問題として, 離散 Gauss 分布を正確に生成することは難しいことが挙げられる. ノイズをある整数区間から一様分布として取った場合でも, 格子問題へと量子帰着が可能であることが 2013 年に Döttling ら [45] により示された. この方向性の研究として, Bai ら [28] により提案された, Rényi エントロピーを用いた, 理想的な Gauss 分布を用いた暗号方式とそれを近似的な分布に置き換えた方式の間での安全性の低下を議論するものがある.

また, 格子問題の計算機による具体的な求解に関して, 2016 年より暗号解読コンテスト LWE Challenge[101] が開催されている. 第 2.1 節に, 2022 年 8 月現在の状況について記載した. 特に 2.3 節で示された各暗号方式のパラメータから見ると, 解が得られている値からは, 大きな隔たりがみられる. 格子に基づく暗号技術は, 各方式毎にパラメータ設定手法に対する制約が異なっていることから, 解読コンテストのサイズに基づく解読到達レベルを, 具体的な暗号方式の安全性の根拠とすることは, 難しいところではあるものの, 古典計算機での解読困難性を測る上での検討の一つに値すると考えられる.

格子に基づく暗号技術の安全性の根拠となる問題は, 古典計算機・量子計算機のいずれにおいても現時点で効率的な解読手法は見つかっていないが, 格子に基づく暗号技術は未だ研究途上にあり, 今後も研究の進捗を注視する必要がある.

第 2 章の参照文献

[1] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, J. Kelsey, Y.-K. Liu, C. Miller et al. Status report on the second round of the NIST post-quantum cryptography standardization process. https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf, 2020.

[2] E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. de la Piedra, T. Pöppelmann, P. Schwabe, D. Stebila. NewHope algorithm specifications and supporting documentation Version 1.0. https://newhopecrypto.org/data/NewHope_2017_12_21.pdf, 2017-11-30. (2023-04-07 閲覧)

[3] E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. de la Piedra, T. Pöppelmann, P. Schwabe, D. Stebila et al. NewHope algorithm specifications and supporting documentation Version 1.02 (Updated March 15, 2019). https://newhopecrypto.org/data/NewHope_2019_04_10.pdf. (2023-04-07 閲覧)

[4] E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. de la Piedra, T. Pöppelmann, P. Schwabe, D. Stebila et al. *NewHope algorithm specifications and supporting documentation Version 1.1 (Updated April 10, 2020)*. https://newhopecrypto.org/data/NewHope_2020_04_10.pdf. (2023-04-07 閲覧)

[5] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, Y.-K. Liu et al. Status report on the third round of the NIST post-quantum cryptography standardization process, NIST IR 8413-upd1, https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413-upd1.pdf, 2022-07. (2023-04-11 閲覧)

[6] M. Albrecht, S. Bai, L. Ducas. A subfield lattice attack on overstretched NTRU assumptions. *CRYPTO 2020*, Part I, volume 9814 of LNCS, pp. 153–178, Springer, 2016.

[7] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe et al. CRYSTALS-Kyber – Algorithm specifications and supporting documentation. https://pq-crystals.org/kyber/data/kyber-specification.pdf, 2017-11-30. (2023-04-07 閲覧)

[8] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe et al. CRYSTALS-Kyber – Algorithm specifications and supporting documentation (version 2.0). https://pq-crystals.org/kyber/data/kyber-specification-round2.pdf, 2019-04-01. (2023-04-07 閲覧)

[9] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe et al. CRYSTALS-Kyber – Algorithm specifications and supporting documentation (version 3.02). https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf, 2021-08-04. (2023-04-07 閲覧)

[10] FrodoKEM – learning with errors key encapsulation. Algorithm specifications and supporting documentation (November 30, 2017). https://frodokem.org/files/FrodoKEM-specification-20171130.pdf. (2023-02-21 閲覧)

[11] E. Alkim, J. W. Bos, L. Ducas, P. Longa, I. Mironov, M. Naehrig, V. Nikolaenko, C. Peikert et al. FrodoKEM – learning with errors key encapsulation. Algorithm specifications and supporting documentation (June 4,

2021). https://frodokem.org/files/FrodoKEM-specification-20210604.pdf. (2023-02-21 閲覧)

[12] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, T. Wunderer. Estimate all the {LWE, NTRU} schemes!(2018). *SCN 2018*, volume 11035 of LNCS, pp. 351–367, Springer, 2018. https://estimate-all-the-lwe-ntru-schemes.github.io/docs/.

[13] M. Ajtai, C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. *STOC '97*, pp. 284–293, ACM, 1997.

[14] M. Albrecht, A. Deo. Large modulus Ring-LWE ≥ Module-LWE. *ASIACRYPT 2017*, Part I, volume 10624 of LNCS, pp. 267–296, Springer, 2017.

[15] M. R. Albrecht, L. Ducas. Lattice attacks on NTRU and LWE: A history of refinements. *Computational Cryptography: Algorithmic Aspects of Cryptology*, Chanter 2, London Mathematical Society Lecture Notes Series 469, Cambridge University Press, 2021.

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

[17] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe. Post-quantum key exchange: A new hope. *SEC'16: 25th USENIX Conference on Security Symposium*, pp. 327-343, USENIX Association, 2016.

[18] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe. NewHope without reconciliation. *IACR Cryptology ePrint Archive*, 2016/1157.

[19] M. R. Albrecht, F. Göpfert, F. Virdia, T. Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. *EUROCRYPT 2020*, volume 10624 of LNCS, pp. 297–322, Springer, 2020.

[20] M. Ajtai. Generating hard instances of lattice problems. *STOC'96*, pp. 99–108, ACM, 1996.

[21] M. Ajtai. Generating hard instances of the short basis problem. *ICALP 1999*, volume 1644 of LNCS, pp. 1–9, Springer, 1999.

[22] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, D. Stebila. Frodo: Take off the Ring! Practical, quantum-secure key exchange from LWE. *ACM CCS 2016*, pp. 1006–1018, ACM, 2016.

[23] S. Bai, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Dilithium algorithm specifications and supporting documentation (Version 3.1). https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf, 2021-02-08. (2023-04-07 閲覧)

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

[25] S. Bai, S. D. Galbraith. Lattice decoding attacks on binary LWE. *ACISP 2014*, volume 8544 of LNCS, pp. 322–337, Springer, 2014.

[26] L. G. Bruinderink, A. Hülsing, T. Lange, Y. Yarom. Flush, Gauss, and reload – A cache attack on the BLISS lattice-based signature scheme. *CHES 2014*, volume 9813 of LNCS, pp. 323–345, Springer, 2016.

[27] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, D. Stehlé. Classical hardness of learning with errors. *STOC'13*, pp. 575–584, ACM, 2013.

[28] S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, R. Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. *Journal of Cryptology*,

volume 31, issue 2, pp. 610–640, 2018.

[29] A. Basso, J. M. B. Mera, J.-P. D'Anvers, A. Karmakar, S. S. Roy, M. V. Beirendonck, F. Vercauteren. SABER: Mod-LWR based KEM (Round 3 Submission). https://www.esat.kuleuven.be/cosic/pqcrypto/saber/files/saberspecround3.pdf.（2023-04-07 閲覧）

[30] A. Banerjee, C. Peikert, A. Rosen. Pseudorandom functions and lattices. *EUROCRYPT 2012*, volume 7237 of LNCS, pp. 719–737, Springer, 2012.

[31] M. Braithwaite. Experimenting with post-quantum cryptography. https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html, 2016-07-07.（2023-04-07 閲覧）

[32] Z. Brakerski, V. Vaikuntanathan. Fully homomorphic encryption from Ring-LWE and security for key dependent messages. *CRYPTO 2011*, volume 6841 of LNCS, pp. 505–524, Springer, 2011.

[33] C. Chen, O. Danba, J. Hoffstein, A. Hülsing, J. Rijneveld, J. M. Schanck, T. Saito, P. Schwabe et al. NTRU Algorithm Specifications And Supporting Documentation (September 30,2020). https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/NTRU-Round3.zip.（2023-02-21 閲覧）

[34] The Chromium Projects. CECPQ2. https://www.chromium.org/cecpq2/.（2023-02-21 閲覧）

[35] C.-M. M. Chung, V. Hwang, M. J. Kannwischer, G. Seiler, C.-J. Shih, B.-Y. Yang. NTT multiplication for NTT-unfriendly rings. *IACR Cryptology ePrint Archive*, 2020/1397.

[36] C. Chen, J. Hoffstein, W. Whyte, Z. Zhang. NIST PQ Submission: NTRUEncrypt A lattice based encryption algorithm. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/NTRUEncrypt.zip.（2023-02-21 閲覧）

[37] Y. Chen, P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. *ASIACRYPT 2011*, volume 7073 of LNCS, pp. 1–20, Springer, 2011.

[38] C. Chuengsatiansup, T. Prest, D. Stehlé, A. Wallet, K. Xagawa. ModFalcon: compact signatures based on Module-NTRU lattices. *ACM ASIACCS'20*, pp. 853–866, ACM, 2020.

[39] J.-P. D'Anvers. SABER: Module-LWR based KEM. *Second PQC Standardization Conference*, https://csrc.nist.gov/Presentations/2019/saber-round-2-presentation, 2019-08-23.（2023-04-07 閲覧）

[40] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Dilithium algorithm specifications and supporting documentation. https://pq-crystals.org/dilithium/data/dilithium-specification.pdf, 2017-11-30.（2023-04-07 閲覧）

[41] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018, issue 1, pp. 238–268, Ruhr-Universität Bochum.

[42] J.-P. D'Anvers, A. Karmakar, S. S. Roy, F. Vercauteren. Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. *AFRICACRYPT 2018*, volume 10831 of LNCS, pp. 282–305, Springer, 2018.

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

[44] L. Ducas, T. Prest. Fast Fourier orthogonalization. *ISSAC'16*, pp. 191–198, ACM, 2016.

[45] N. Döttling, J. Müller-Quade. Lossy codes and a new variant of the learning-with-errors problem. *EUROCRYPT 2013*, volume 7881 of LNCS, pp. 13–34, Springer, 2013.

[46] L. Ducas, J. Schanck. Security estimation scripts for Kyber and Dilithium. https://github.com/pq-

crystals/security-estimates.（2023-04-12 閲覧）

[47] L. Ducas, M. Stevens, W. van Woerden. Advanced lattice sieving on GPUs, with tensor cores. *EUROCRYPT 2021*, volume 12697 of LNCS, pp. 249–279, Springer, 2021.

[48] T. Espitau, P.-A. Fouque, F. Gérard, M. Rossi, A. Takahashi, M. Tibouchi, A. Wallet, Y. Yu. Mitaka: A simpler, parallelizable, maskable variant of Falcon. *EUROCRYPT 2022*, volume 13277 of LNCS, pp. 222–253, Springer, 2022.

[49] ETSI TR 103 616 V1.1.1 (2021-09) CYBER; Quantum-safe signatures. https://www.etsi.org/deliver/etsi_tr/103600_103699/103616/01.01.01_60/tr_103616v010101p.pdf.（2023-04-07 閲覧）

[50] T. Espitau, M. Tibouchi, A. Wallet, Y. Yu. Shorter hash-and-sign lattice-based signatures. *CRYPTO 2022*, volume 13508 of LNCS, pp. 245–275, Springer, 2022.

[51] Federal Office for Information Security. BSI – Technical guideline (Cryptographic mechanisms: Recommendations and key lengths). Version 2023-01, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile&v=10, 2023-01-09.（2023-04-07 閲覧）

[52] P.-A. Fouque, F. Gérard, M. Rossi, Y. Yu. Zalcon: an alternative FPA-free NTRU sampler for Falcon. *Third PQC Standardization Conference*, https://csrc.nist.gov/Presentations/2021/zalcon-an-alternative-fpa-free-ntru-sampler-for-fa, 2021-06-09.（2023-04-07 閲覧）

[53] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler et al. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Specification v1.0. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/Falcon.zip.（2023-04-07 閲覧）

[54] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler et al. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Specification v1.2 – 01/10/2020. https://falcon-sign.info/falcon.pdf.（2023-04-07 閲覧）

[55] M. Fahr, H. Kippen, A. Kwong, T. Dang, J. Lichtinger, D. Dachman-Soled, D. Genkin, A. Nelson et al. When Frodo flips: End-to-end key recovery on FrodoKEM via rowhammer. *IACR Cryptology ePrint Archive*, 2022/952.

[56] A. Fiat, A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. *CRYPTO 1986*, volume 263 of LNCS, pp. 186–194, Springer, 1987.

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

[58] O. Goldreich, S. Goldwasser, S. Halevi. Public-key cryptosystems from lattice reduction problems. *CRYPTO 1997*, volume 1294 of LNCS, pp. 112–131, Springer, 1997.

[59] Q. Guo, T. Johansson, A. Nilsson. A key-recovery timing attack on post-quantum primitives using the Fujisaki-Okamoto transformation and its application on FrodoKEM. *CRYPTO 2020*, volume 12171 of LNCS, pp. 359–386, Springer, 2020.

[60] C. Gentry, C. Peikert, V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. *STOC '08*, pp. 197–206, ACM, 2008.

[61] M. Grillere, P. Thomassen, N. Wisiol, FALCON-512 in PowerDNS. https://blog.powerdns.com/2022/

04/07/falcon-512-in-powerdns/, 2022-04-07.（2023-04-07 閲覧）

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

[63] A. Hülsing, K.-C. Ning, P. Schwabe, F. Weber, P. R. Zimmermann. Post-quantum WireGuard. *IEEE S&P 2021*, pp. 304–321, IEEE, 2021.

[64] J. Howe, T. Pöppelmann, M. O'Neill, E. O'Sullivan, T. Güneysu. Practical lattice-based digital signature scheme. *ACM Transactions on Embedded Computing Systems*, volume 14, issue 3, article 41, pp. 1–24, ACM, 2015.

[65] J. Hoffstein, J. Pipher, J. H. Silverman. NTRU: A ring-based public key cryptosystem. *ANTS-III*, volume 1423 of LNCS, pp. 267–288, Springer, 1998.

[66] A. Hülsing, J. Rijneveld, J. Schanck, P. Schwabe. High-speed key encapsulation from NTRU. *CHES 2017*, volume 10529 of LNCS, pp. 232–252, Springer, 2017.

[67] A. Hülsing, J. Rijneveld, J. M. Schanck, P. Schwabe. NTRU-HRSS-KEM Algorithm Specifications And Supporting Documentation (November 30, 2017). https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/NTRU_HRSS_KEM.zip.（2023-02-21 閲覧）

[68] R. Kannan. Improved algorithms for integer programming and related lattice problems. *STOC '83*, pp. 193–206, ACM, 1983.

[69] K. Kim, M. Tibouchi, A. Wallet, T. Espitau, A. Takahashi, Y. Yu, S. Guilley. SOLMAE – Algorithm specifications. *KpqC Competition Round 1*, https://kpqc.or.kr/images/pdf/SOLMAE.pdf.（2023-04-07 閲覧）

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

[71] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, volume 261, pp. 515–534, Springer, 1982.

[72] K. Lauter, M. Naehrig, V. Vaikuntanathan. Can homomorphic encryption be practical? *ACM CCSW 2011*, pp. 113–124, ACM, 2011.

[73] R. Lindner, C. Peikert. Better Key sizes (and attacks) for LWE-Based encryption. *CT-RSA 2011*, volume 6558 of LNCS, pp. 319–339, Springer, 2011.

[74] V. Lyubashevsky, C. Peikert, O. Regev. On ideal lattices and learning with errors over rings. *EUROCRYPT 2010*, volume 6110 of LNCS, pp. 1–23, Springer, 2010.

[75] V. Lyubashevsky, C. Peikert, O. Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM*, volume 60, issue 6, pp. 1–35, ACM, 2013.

[76] A. Langlois, D. Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, volume 75, issue 3, pp. 565–599, Springer, 2015.

[77] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. *ASIACRYPT 2009*, volume 5912 of LNCS, pp. 319–339, Springer, 2009.

[78] V. Lyubashevsky. Lattice signatures without trapdoors. *EUROCRYPT 2012*, volume 7237 of LNCS, pp. 738–755, Springer, 2012.

[79] V. Lyubashevsky. CRYSTALS-Dilithium Round 3 presentation. *Third PQC Standardization Conference*, https://csrc.nist.gov/Presentations/2021/crystals-dilithium-round-3-presentation, 2021-06-

07.（2023-02-21 閲覧）

[80] NIST. Selected Algorithms 2022. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.（2023-02-21 閲覧）

[81] National Security Agency. Announcing the commercial national security algorithm suite 2.0. https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF, 2022-09-07.（2023-04-07 閲覧）

[82] P. Q. Nguyen, J. Stern. Cryptanalysis of the Ajtai-Dwork cryptosystem. *CRYPTO 1998*, volume 1462 of LNCS, pp. 223–242, Springer, 1998.

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

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

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

[86] T. Plantard, W. Susilo, K. T. Win. A digital signature scheme based on CVP_∞. *PKC 2008*, volume 4939 of LNCS, pp. 288–307, Springer, 2008.

[87] E. W. Postlethwaite, F. Virdia. On the success probability of solving unique SVP via BKZ. *PKC 2021*, volume 12710 of LNCS, pp. 68–98, Springer, 2021.

[88] C. Peikert, V. Vaikuntanathan, B. Waters. A framework for efficient and composable oblivious transfer. *CRYPTO 2008*, volume 5157 of LNCS, pp. 554–571, Springer, 2008.

[89] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *STOC '05*, pp. 84–93, ACM, 2005.

[90] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, volume 56, issue 6, pp. 1–40, 2009.

[91] O. Regev. The learning with errors problem (invited survey). *CCC 2010*, pp. 191–204, IEEE, 2010.

[92] M. Roşca, A. Sakzad, D. Stehlé, R. Steinfeld. Middle-product learning with errors. *CRYPTO 2017*, volume 10403 of LNCS, pp. 283–297, Springer, 2017.

[93] C. Saliba. Error correction and reconciliation techniques for lattice-based key generation protocols. https://tel.archives-ouvertes.fr/tel-03718212/file/SALIBA_2022.pdf, tel-03718212, v1, 2022.

[94] P. Schwabe. 6 years of NIST PQC, looking back and ahead. *PQCrypto 2022*, Invited talk, 2022.

[95] C. P. Schnorr, M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical programming*, volume 66, pp. 181-199, Springer, 1994.

[96] C. Saliba, L. Luzzi, C. Ling. Error correction for FrodoKEM using the Gosset lattice. *arXiv*: 2110.01740, 2021.

[97] D. Stehlé, R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. *EUROCRYPT 2011*, volume 6632 of LNCS, pp. 27–47, Springer, 2011.

[98] D. Stehlé, R. Steinfeld. Making NTRUEncrypt and NTRUSign as secure as standard worst-case problems over ideal lattices. *IACR Cryptology ePrint Archive*, 2013/004.

[99] D. Stehlé, R. Steinfeld, K. Tanaka, K. Xagawa. Efficient public key encryption based on ideal lattices.

ASIACRYPT 2009, volume 5912 of LNCS, pp. 617–635, Springer, 2009.

[100] T. Saito, K. Xagawa, T. Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. *ASIACRYPT 2018*, volume 10822 of LNCS, pp. 520–551, Springer, 2018.

[101] TU Darmstadt, UC San Diego. LWE Challenge. `https://www.latticechallenge.org/lwe_challenge/challenge.php`.（2023-04-12 閲覧）

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

[103] wolfSSL. wolfSSL support for Apache httpd and curl (Post-Quantum Edition). `https://github.com/wolfSSL/osp/blob/master/apache-httpd/README_post_quantum.md`.（2023-02-21 閲覧）

[104] Y. Yu, L. Ducas. Second order statistical behavior of LLL and BKZ. *SAC 2017*, volume 10719 of LNCS, pp. 3–22, Springer, 2017.

第 3 章

符号に基づく暗号技術

本章では符号に基づく暗号技術についてまとめる．符号に基づく暗号技術の安全性は LPN 問題やシンドローム復号問題を解く計算の困難性に依存している．

■準備: 本章で使用する記号・用語を以下にまとめる．以下では, q を素数 p のべきとする．すなわち, ある正整数 k が存在して $q = p^k$ である．以下では log の底が省略されている場合は底が 2 であることを示す．自然対数を用いる場合は ln と書く．

有限体: $\mathbb{F}_q$ で位数が q の有限体を表す.
ハミング重みとハミング距離: V_n を有限体 $\mathbb{F}_q$ 上の n 次元ベクトル空間とする.

- ベクトル $\boldsymbol{v} = (v_1, v_2, \ldots, v_n) \in V_n$ のハミング重みとは, 非ゼロの係数の数である．すなわち, $\mathsf{HW}(\boldsymbol{v}) = \#\{i \mid v_i \neq 0\}$ である.
- ハミング距離を $d_H(\boldsymbol{x}, \boldsymbol{y}) = \mathsf{HW}(\boldsymbol{x} - \boldsymbol{y})$ で定義する.
- $\mathcal{S}_H(n, w)$ でハミング重みが w の n 次元ベクトル全体の集合を表す.
- $\mathcal{S}_H^{\leq}(n, w)$ でハミング重みが w 以下の n 次元ベクトル全体の集合を表す.

■線形符号: 自然数 n および 素数べき q について, $\mathbb{F}_q$ 上の n 次元ベクトル空間の部分空間を $\mathbb{F}_q$ 上の線形符号と呼び, $\mathcal{C}$ で表す．n を符号長と呼ぶ．$\mathbb{F}_q$ 上の線形符号 $\mathcal{C}$ の符号語 $\boldsymbol{c}$ は, 一次独立な k 個の符号語 $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$ の $\mathbb{F}_q$ 係数の一次結合で表すことができる．このとき $[\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k]$ を基底と呼ぶ．k を線形符号の次元と呼ぶ．$\mathbb{F}_q$ 上の線形符号の符号長が n, 次元が k であるとき, $[n, k]_q$ -線形符号とよぶ．$[n, k]_q$ -線形符号 $\mathcal{C}$ の生成行列とは, 符号 $\mathcal{C}$ の基底ベクトルを行とする行列 $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$ である．$[n, k]_q$ -線形符号 $\mathcal{C}$ のパリティ検査行列とは, 行列 $\boldsymbol{H} \in \mathbb{F}_q^{r \times n}$ で, $\boldsymbol{c} \in \mathbb{F}_q^n$ に対して, $\boldsymbol{c} \in \mathcal{C}$ ならばかつその時に限り $\boldsymbol{c} \cdot \boldsymbol{H}^\top = \boldsymbol{0}$ となるものである．$\boldsymbol{H}$ の行が線形独立であれば, $r = n - k$ である．

$[n, k]_q$ -線形符号 $\mathcal{C}$ の最小距離 d とは, 符号 $\mathcal{C}$ の非ゼロ符号語の中で最小のハミング重みをもつ符号語のハミング重みのことを言う．すなわち, $d = \min_{\boldsymbol{c} \in \mathcal{C} \setminus \{\boldsymbol{0}\}} \mathsf{HW}(\boldsymbol{c})$ である．

$[n, k]_q$ -線形符号 $\mathcal{C}$ の生成行列 $\boldsymbol{G}$ が決まっている場合, メッセージ $\boldsymbol{s} \in \mathbb{F}_q^k$ を 符号語 $\boldsymbol{sG}$ と一対一対応させることができる．符号は, 生成行列やパリティ検査行列をうまく設計することで, 符号語に加えられた誤りを訂正することができる．送信する符号語を $\boldsymbol{c}$ とし, 通信路上で乗った誤りを $\boldsymbol{e}$ とする．受信者側は受信語として, $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ を得る．受信者は符号の復号アルゴリズムを用いて復号を行い, $\boldsymbol{y}$ から $\boldsymbol{c}$ を得る．受信者がハミング重み t までの誤り $\boldsymbol{e}$ を一意に訂正できるとき, 符号の訂正能力が t であるという．一般に, $t \leq \lfloor (d-1)/2 \rfloor$ であることが分かる．復号アルゴリズムはパリティ検査行列を用いることもある．その際に, $\boldsymbol{t} = \boldsymbol{y}\boldsymbol{H}^\top$ を計算し, これをシンドロームと呼ぶ.

本稿では，具体的な線形符号 (リード・ソロモン符号, リード・マラー符号, Goppa 符号) の詳細については扱わな

い. 符号理論の教科書や, 電子情報通信学会 知識の森 「1 群 2 編 符号理論」 [29] などを参照されたい.

3.1 符号に基づく暗号技術の安全性の根拠となる問題

本節では Learning Parity with Noise（LPN）問題や符号に関連する問題の困難性について報告する.

3.1.1 LPN 問題とは

LPN 問題とは誤差付きの線形方程式を解けるかどうかという問題である. 1993 年に, Blum, Furst, Kearns, Lipton [8] が困難と思われる問題として挙げ, 定式化を行った. 第 2 章において, この問題を一般化した LWE 問題を既に扱っている.

Ber_τ でパラメータ τ のベルヌーイ分布を表すことにする. (確率 τ で 1, 確率 $1-\tau$ で 0 となる $\mathbb{F}_2$ 上の分布である.) また, 自然数 $k \geq 1$ について, Ber_τ^k で, Ber_τ から独立に k 個サンプルを取ったときの $\mathbb{F}_2^k$ 上の分布を表す.

■ **LPN 問題**: $\mathbb{F}_2$ 上の分布 χ および $\boldsymbol{s} \in \mathbb{F}_2^k$ について, オラクル $\mathcal{O}_{\boldsymbol{s},\chi}$ を以下で定義する. (1) $\boldsymbol{a}$ を $\mathbb{F}_2^k$ から一様ランダムに選び, (2) e を分布 χ に従い選び, (3) $b = \boldsymbol{s} \cdot \boldsymbol{a}^\top + e$ と計算し, (4) $(\boldsymbol{a}, b)$ を出力する.

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

定義 3.1 (探索版 LPN 問題) 探索版 LPN 問題とは, オラクル $\mathcal{O}_{\boldsymbol{s},\chi}$ へのアクセスが可能なときに, $\boldsymbol{s}$ を出力する問題である.

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

定義 3.2 (探索版 LPN 仮定) $\mathbb{F}_2$ 上の確率分布 χ について, 攻撃者 $\mathcal{A}$ の優位性を

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

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

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

定義 3.3 (判定版 LPN 問題) 判定版 LPN 問題とは, オラクル $\mathcal{O}_{\boldsymbol{s},\chi}$ またはオラクル $\mathcal{U}$ へのアクセスが与えられたときに, どちらのオラクルにアクセスしているかを判定する問題である.

定義 3.4 (判定版 LPN 仮定) $\mathbb{F}_2$ 上の確率分布 χ について, 攻撃者 $\mathcal{A}$ の優位性を

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

で定義する. 任意の多項式時間の攻撃者 $\mathcal{A}$ について, その優位性が無視できるほど小さいとき, 判定版 LPN 仮定が成立するという.

探索版 LPN 問題にはランダム自己帰着が存在する [8]. すなわち, 一様ランダムに選ばれた $\boldsymbol{s} \in \mathbb{F}_2^k$ について探索版 LPN 問題を解けるならば, 任意の $\boldsymbol{s} \in \mathbb{F}_2^k$ について探索版 LPN 問題を解くことが出来る.

Katz, Shin, Smith [32] によれば, [8, 45] と同様に判定版 LPN 仮定を探索版 LPN 仮定に帰着することが出来る.

定理 3.5 ([32]) 判定版 $\mathsf{LPN}_{k,\tau}$ 仮定を破る t ステップ, n 回のクエリ, 優位性 δ の攻撃者が存在すると仮定する. このとき, 探索版 $\mathsf{LPN}_{k,\tau}$ 仮定を破る t' ステップ, n' 回のクエリ, 優位性 δ' の攻撃者が存在する. ここで,

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

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

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

3.1.2 LPN 問題の拡張

3.1.2.1 復号問題

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

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

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

3.1.2.2 シンドローム復号問題

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

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

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

$\boldsymbol{A} \in \mathbb{F}_2^{k\times n}$ で生成される符号のパリティ検査行列を $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ とし, $\boldsymbol{b}\cdot\boldsymbol{H}^\top (= \boldsymbol{e}\cdot\boldsymbol{H}^\top)$ を $\boldsymbol{u}$ とすれば, $\mathsf{LPN}_{k,n,\tau}$ 問題や復号問題をシンドローム復号問題 $\mathsf{SD}_{k,n,O(\tau n)}$ に変換可能である.

3.1.2.3 Exact-LPN 問題

誤差分布として, $\boldsymbol{e} \leftarrow \mathsf{Ber}_\tau^n$ ではなく, ハミング重みが丁度 w のものだけを考える (すなわち $\boldsymbol{e} \leftarrow \mathcal{S}_H(n,w)$). このように誤差分布を変えた問題を Exact-LPN 問題と呼ぶ.

3.1.2.4 Sparse-LPN 問題

一部の暗号方式では, $\boldsymbol{s}$ のハミング重みが小さい, すなわち, 疎 (スパース, sparse) であることを要求する. Applebaum ら [4] は $\boldsymbol{s}$ を誤差分布である χ^k から選んだ場合の LPN 問題と $\boldsymbol{s}$ を $\mathbb{F}^k$ からランダムに選んだ場合の問題とが等価であることを示している. このように $\boldsymbol{s}$ の分布を変えた問題を Sparse-LPN 問題と呼ぶ.

3.1.2.5 Ring-LPN 問題

Heyse, Kiltz, Lyubashevsky, Paar, Pietrzak [28] は, Ring-LPN 問題を定義した. この問題は Ring-LWE 問題 (2.1.1.1 節) と同様に定義される.

定義 3.6 (探索版 Ring-LPN 問題) 適当な k 次の $\mathbb{F}_q$ 係数多項式 $f(x)$ を考え, 環 $R_q = \mathbb{F}_q[x]/(f(x))$ を固定する. R_q 上の確率分布 χ を固定する.

R_q 上の誤差分布 χ および $s \in R_q$ について, オラクル $\mathcal{O}_{\boldsymbol{s},\chi}$ を以下で定義する. (1) a を R_q から一様ランダムに選び, (2) e を分布 χ に従い選び, (3) $b = sa + e$ と計算し, (4) $(a, b) \in R_q^2$ を出力する.

探索版 Ring-LPN 問題とは, オラクル $\mathcal{O}_{s,\chi}$ へのアクセスが可能なときに, $s \in R_q$ を出力する問題である.

3.1.2.6 Module-LPN 問題

Module-LWE 問題 (2.1.1.1 節) と同様に, Module-LPN 問題を定義することができる.

定義 3.7 (探索版 Module-LPN 問題) 適当な k_0 次の $\mathbb{F}_q$ 係数多項式 $f(x)$ を考え, 環 $R_q = \mathbb{F}_q[x]/(f(x))$ を固定する. R_q 上の確率分布 χ を固定する.

R_q 上の誤差分布 χ および $\boldsymbol{s} \in R_q^k$ について, オラクル $\mathcal{O}_{\boldsymbol{s},\chi}$ を以下で定義する. (1) $\boldsymbol{a}$ を R_q^k から一様ランダムに選び, (2) e を分布 χ に従い選び, (3) $b = \boldsymbol{s} \cdot \boldsymbol{a}^\top + e$ と計算し, (4) $(\boldsymbol{a}, b) \in R_q^{(k+1)}$ を出力する.

探索版 Module-LPN 問題とは, オラクル $\mathcal{O}_{\boldsymbol{s},\chi}$ へのアクセスが可能なときに, $\boldsymbol{s} \in R_q^k$ を出力する問題である.

3.1.3 LPN 問題に対する評価

サンプル数を固定した場合, $\boldsymbol{A}$ および $\boldsymbol{b}$ の最悪時を考えると NP 困難になることが Berlekamp, McEliece, van Tilborg [15] によって示されている. [*1] また, Håstad [27] により近似版 LPN 問題[*2] の NP 困難性も示されている.

しかし平均時の困難性についてはよく分かっていない. そのため LPN 問題を解くための提案されたアルゴリズムについて調査を行った.

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

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

1. Blum, Kalai, Wasserman [10] の BKW アルゴリズム
2. Arora, Ge [5] の「再線形化」アルゴリズム
3. シンドローム復号問題として解くアルゴリズム

[*1] $\boldsymbol{A}$ および $\boldsymbol{b}$ を与えられたときに, 線形方程式 $\boldsymbol{s}\boldsymbol{a}_i^\top = b_i$ を満たす数を最大化する $\boldsymbol{s}$ を探索する問題を考える.

[*2] $\boldsymbol{A}$ および $\boldsymbol{b}$ を与えられたときに, 線形方程式 $\boldsymbol{s}\boldsymbol{a}_i^\top = b_i$ を近似度×最大値以上満たす $\boldsymbol{s}$ を探索する問題.

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

3.1.3.1 BKW アルゴリズムおよびその改良

Blum, Kalai, Wasserman [10] は BKW アルゴリズムと呼ばれるアルゴリズムを提案した.

基本アイデアは以下である. オラクルからのサンプル $(\boldsymbol{a}, b)$ が常に $\boldsymbol{a} = (1, 0, \ldots, 0)$ という形であれば, $b = s_1 + e$ となる. このようなサンプルを大量に集めれば, s_1 を多数決法で求めることが出来る. 一般に $\boldsymbol{u}_j$ を j 番目の単位ベクトルとして, $(\boldsymbol{u}_j, b)$ という形のサンプルを集めれば s_j を多数決法で求められる. そこでオラクル $\mathcal{O}_{\boldsymbol{s},\tau}$ からのサンプルを用いて, このようなサンプルを生成することを目指す.

■BKW アルゴリズムの概要: $(t-1)\ell < k \leq t\ell$ を満たす適当な自然数 t, ℓ を固定する. 以下では,

$$A_{\boldsymbol{s},\delta,i} = \{\boldsymbol{a} \leftarrow \mathbb{F}_2^{k-i\ell} \times \{0\}^{i\ell}, e \leftarrow \mathsf{Ber}_{(1+\delta)/2} : (\boldsymbol{a}, \boldsymbol{s} \cdot \boldsymbol{a}^\top + e)\}$$

というオラクルを考える. $A_{\boldsymbol{s},\delta,i}$ から得たサンプル $(\boldsymbol{a}, b)$ は $\boldsymbol{a}$ の末尾から $i\ell$ 個の要素が必ず 0 である. $i = 0$, $\delta = 1 - 2\tau$ とすれば, $A_{\boldsymbol{s},\delta,i} = \mathcal{O}_{\boldsymbol{s},\tau}$ となる.

基本アルゴリズムは以下である.

1. $A_{\boldsymbol{s},\delta_0,0} = \mathcal{O}_{\boldsymbol{s},\tau}$ からのサンプルを L_0 個用意する.
2. $i = 0, 1, \ldots, t-2$ について, サイズ L_i の $A_{\boldsymbol{s},\delta_i,i}$ からのサンプルを用いて, $O(L_i)$ 時間でサイズ $L_{i+1} = L_i - 2^k$ の $A_{\boldsymbol{s},\delta_i^2,i+1}$ からのサンプルを構成する.
   - サンプル $(\boldsymbol{a}, b) \in L_i$ について, $\boldsymbol{a} = (a_1, a_2, \ldots, a_{k-i\ell}, 0, \ldots, 0) \in \mathbb{F}_2^k$ の $(a_{k-(i+1)\ell+1}, a_{k-(i+1)\ell+2}, , \ldots, a_{k-i\ell}) \in \mathbb{F}_2^\ell$ に従って分類を行う.
   - 各組で代表を一つとり, それを $(\boldsymbol{a}^*, b^*)$ とする.
   - 各組の代表以外の要素 $(\boldsymbol{a}, b)$ を $(\boldsymbol{a} - \boldsymbol{a}^*, b - b^*)$ で置き換える.
   - 全組をまとめてサイズ $L_i - 2^\ell$ の $A_{\boldsymbol{s},\delta_i^2,i+1}$ からのサンプルとする.

   最終的に, サイズ $L_{t-1} = L_0 - (t-1)2^\ell$ の $A_{\boldsymbol{s},\delta_0^{2^{t-1}},t-1}$ からのサンプルが得られる.
3. 得られた L_{t-1} 個の $A_{\boldsymbol{s},\delta_0^{2^{t-1}},t-1}$ からのサンプルを用いて, s_j を投票で決める.
   - $j = 1, 2, \ldots, k - (t-1)\ell$ について, $\boldsymbol{u}_j$ を $\mathbb{F}_2^k$ の標準基底 j 番目の単位ベクトルとする. サンプル $\{(\boldsymbol{a}_i, b_i)\}_{i=1,2,\ldots,m}$ から z 個のベクトルを $\boldsymbol{a}_{i_1} + \boldsymbol{a}_{i_2} + \cdots + \boldsymbol{a}_{i_z} = \boldsymbol{u}_j$ となるように選ぶ. このとき, $b_{i_1} + b_{i_2} + \cdots + b_{i_z} = s_j + e_{i_1} + \cdots + e_{i_z}$ となり, 誤差が 0 になる確率は $\Pr[e_{i_1} + e_{i_2} + \cdots + e_{i_z} = 0] > 1/2 + (1 - 2\delta_0^{2^{t-1}})^z/2$ で与えられる. 適当な回数この試行を行い, s_j を多数決投票で決めれば良い.

Blum らの見積もりでは, サンプル数および計算ステップ数は $\delta_0 = 1 - 2\tau$ として, $\mathrm{poly}\left(\delta_0^{-2^t}, 2^\ell\right)$ であった. $\tau < 1/2$ を定数とし, $t = \dfrac{1}{2}\log k$, $\ell = 2k/\log k$ とすれば, 時間計算量・空間計算量ともに $2^{O(k/\log k)}$ を得る.

■LF アルゴリズム: Levieil と Fouque [36] は BKW アルゴリズムの一部を改良し LF1 アルゴリズムを提案した.

簡単のために $k = t\ell$ を仮定する. BKW アルゴリズムでは基本アルゴリズムのステップ 3 において $\boldsymbol{s}$ の各要素を 1 つずつ決定している. ステップ 3 において得られたサンプルは, $A_{\boldsymbol{s},\delta_0^{2^{t-1}},t-1}$ からのサンプルであるため, $((a_1, a_2, \ldots, a_k, 0, \ldots, 0), b)$ という形をしている. このとき, $b = \displaystyle\sum_{i=1}^{\ell} a_i s_i + e$ となり, サンプルに影響を与えるのは, $\boldsymbol{s}$ の ℓ ビット分である. LF アルゴリズムでは, $s_1, s_2, \ldots, s_\ell$ を総当りで計算する.

Levieil と Fouque は BKW アルゴリズムおよび LF1 アルゴリズムが必要とするサンプル数および計算ステップ数

を, 以下のように詳細に解析した. [*4]

定理 3.8 $k = t\ell$ とし, $\delta = 1 - 2\tau$ とする.

- *BKW* アルゴリズムはクエリ数 $n = 20\ln(4k)2^{\ell}\delta^{-2^t}$, ステップ数 $T = O(ktn)$, メモリ量 $M = kn$, 成功確率 $\theta = 1/2$ で $\mathsf{LPN}_{k,n,\tau}$ 問題を解く.
- *LF1* アルゴリズムはクエリ数 $n = (8\ell{+}200)\delta^{-2^t} + (t{-}1)2^{\ell}$, ステップ数 $T = O(ktn)$, メモリ量 $M = kn{+}\ell 2^{\ell}$. 成功確率 $\theta = 1/2$ で $\mathsf{LPN}_{k,n,\tau}$ 問題を解く.

Levieil と Fouque は, LF1 アルゴリズムに一部のヒューリスティクを組み合わせた LF2 アルゴリズムも提案している. 報告によれば, $k = 99$, $\tau = 1/4$, $n = 10000$ の LPN 問題を CPU: Pentium 4 (3GHz), RAM: 1GB のマシンで解くことが可能である. Devadas, Ren, Xiao [18] は LF2 アルゴリズムについて詳細な解析を与え, BKW アルゴリズムとの比較を行っている. Devadas らの報告によれば, メモリを $O(\delta^{-2^t})$ 倍多く使うが, 時間計算量が $O(\delta^{-2^t})$ 倍改善されるとのことである.

■Kirchner の指摘: Kirchner [30] は一様ランダムに選ばれた $\boldsymbol{s}$ よりは Ber_{τ} に従って選ばれる誤りベクトル $\boldsymbol{e}$ の方が, ハミング重みが小さく, 取りうる値が少ないことに着目した. そこで, LPN 問題を Sparse-LPN 問題に置き換えた上で問題を解くことを提案している.

Kirchner の手法は以下のようにまとめられる.

1. Applebaum ら [4] と同様の手法を用いて, $\mathcal{O}_{\boldsymbol{s},\chi}$ というオラクルを $\boldsymbol{e}' \leftarrow \mathsf{Ber}_{\tau}^{k}$ とランダムに選んだ場合の $\mathcal{O}_{\boldsymbol{e}',\chi}$ というオラクルに変換する.
2. BKW アルゴリズムや LF アルゴリズムと同様に基本アルゴリズムのステップ 1, 2 を行い, $A_{\boldsymbol{e}',\delta^{2^{t-1}},t-1}$ からのサンプルを得る.
3. ステップ 3 で, ℓ ビットを決定する際に, $\boldsymbol{e}'$ の該当部分のハミング重みが少ないことを考慮して総当りを行う.
4. ステップ 1 の逆を行い, $\boldsymbol{e}'$ を $\boldsymbol{s}$ に戻す.

一般の $\boldsymbol{s}$ であれば, 総当りに必要な回数は 2^{ℓ} となる. 一方, $\boldsymbol{e}'$ は疎であることが期待される. $d \geq 1$ を固定し ℓ が十分に大きいとする. このとき, Chernoff の補題により, 圧倒的な確率の下で, ハミング重みは $(1 + 1/d)\tau\ell$ 以下である. よって, $\boldsymbol{e}'$ の候補数は $\binom{\ell}{(1+1/d)\tau\ell}$ 以下となり, 総当りに必要な回数が削減される.

■Ring-LPN 問題への応用: Bernstein と Lange [11] は Levieil と Fouque の高速化手法および Kirchner のアイデアを用いることにより, Ring-LPN 問題の解法が高速化できることを示している.

■その後の進展: Guo, Johansson, Löndahl [25] は, covering codes と呼ばれる符号を用いて Kirchner の手法の高速化を提案している. Kirchner の手法ではステップ 3 で, $A_{\boldsymbol{e}',\delta^{2^{t-1}},t-1}$ からのサンプル $\{(\boldsymbol{a}_i, b_i)\}$ が得られる. この $\boldsymbol{a}_i$ を covering code の受信語とみなすことで探索空間の圧縮を行い, 高速化を行っている. [*5]

Zhang, Jiao, Wang [47] は別の符号を用いて GJL アルゴリズムを改良している.

Bogos と Vaudenay [16] は GJL アルゴリズムの解析が一部欠けていることを分析し, 最適化を行いつつ詳細な計算量評価を与えた. Bogos らは同時に Gauss アルゴリズムと呼ばれるアルゴリズムについても解析を与えている.

[*4] 後に, Zhang, Jiao, Wang [47] らにより, この解析にはヒューリスティックが必要との指摘があった.

[*5] ただし, 国際会議でのプレゼンテーションではサンプル数が不足していたとの報告があり, 計算量・メモリ・サンプル数の評価は見直されている. 詳しくは, [47] および [16] を参照のこと

Esser, Kübler, May [21] では BKW および Gauss アルゴリズムにより詳細な解析を行った.

Esser, Heuer, Kübler, May, Sohler [20] は BKW アルゴリズムに対して時間・メモリのトレードオフを提案している.

■サンプル数が少ない場合: これまでに挙げてきた BKW アルゴリズムおよびその改良では, サンプルが $O(2^{k/\log k})$ 個必要であった. Lyubashevsky [37] はサンプル数が $k^{1+\epsilon}$ 個と少ない場合であっても, BKW アルゴリズムを適用できるような指数個のサンプルの構成法を示している. Kirchner [30] も同様の構成法を示している. Esser, Kubler, May [21] はサンプル数が少ない場合の BKW および Gauss アルゴリズムについて, より詳細な解析を行った.

3.1.3.2 Arora-Ge アルゴリズム

Arora と Ge [5] は多変数多項式問題で古くから用いられている再線形化と呼ばれる手法を用いて, LPN 問題を解くことを考えた. このアルゴリズムを $\mathsf{LPN}_{k,n,\tau}$ に用いた場合, $w = \tau n$ として, $\mathrm{poly}(k^w)$ 時間で解くことができる. $\mathrm{poly}(k^w) = 2^{O(\tau n \log k)}$ であるから, $\tau = o(k/(n\log^2 k))$ のようにエラーが疎であれば, BKW アルゴリズムよりも効率が良い. 実際の符号暗号のパラメータ設定では, エラーをこのように疎に設定することはないため, 暗号の攻撃アルゴリズムとして用いるには重要度が低い. そのため, 本稿では詳細を省く.

3.1.3.3 SD 問題を経由するアルゴリズム

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

対応する線形符号の最小距離を d と置く. (2 進符号の場合, Gilvert-Varshamov 限界により, $k/n \approx 1 - H(d/n)$ である[*6].) $w \approx d$ の場合を Full Distance Decoding の場合と呼び, $w \approx d/2$ の場合を Half Distance Decoding と呼ぶ.

この問題を総当りで解くには, ハミング重みが w の n 次元ベクトル $\boldsymbol{e}$ を列挙すればよい. そのため, 時間計算量は $O\left(\binom{n}{w}\right)$ となる.

より効率的な手法として, Prange は “Information set decoding” と呼ばれる手法 [44] を提案した. 基本アイデアは以下である:

1. 一様ランダムに $\boldsymbol{H}$ の列ベクトルを入れ替え, $\tilde{\boldsymbol{H}} = \boldsymbol{H}\cdot\boldsymbol{P}$ とする. ($\boldsymbol{P}$ は置換行列.)
2. $\tilde{\boldsymbol{H}}$ を組織化し, $[\boldsymbol{I}_{n-k} \mid \boldsymbol{Z}] = \boldsymbol{S}\cdot\tilde{\boldsymbol{H}}$ とする.
3. $\boldsymbol{u}' = \boldsymbol{u}\boldsymbol{S}^\top$ を計算する.
4. $\boldsymbol{u}'$ のハミング重みが w 以下であれば, この置換 $\boldsymbol{P}$ を採用し $\boldsymbol{e} = (\boldsymbol{u}', \boldsymbol{0}_k)\cdot\boldsymbol{P}^\top$ を出力する.

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

Stern [46] 以降, 空間計算量を犠牲にすることで時間計算量を引き下げるアルゴリズムが多数提案されている. 以下では, Both と May [14] による時間計算量の表を, 表 3.1 および 3.2 に示す. この表は, 時間計算量を最小化した場合の $R = k/n$ の最悪時 (1/2 の少し下) についてまとめられている. したがって, 問題のパラメータによっては, 表の数値よ

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

表 3.1: 確率 1/2 以上で SD 問題を解く場合のパラメータ例 (Full Distance Decoding の場合)

	$\log(\text{Time})/n$	$\log(\text{Space})/n$	備考
Pra62 (Lee-Brickel)	0.121	–	[44], [34]
Stern89	0.117	0.0135	[46]
MMT11	0.112	0.0216	[39]
BJMM12	0.102	0.0286	[9]
MO15	0.0967	0.0???	[40]; 空間計算量が明記されていないため, 0.0???とした
BM17	0.0953	0.0910	[13]; MO15 を最適化したもの
BM18	0.0885	0.0736	[14]

表 3.2: 確率 1/2 以上で SD 問題を解く場合のパラメータ例 (Half Distance Decoding の場合)

	$\log(\text{Time})/n$	$\log(\text{Space})/n$	備考
Pra62 (Lee-Brickel)	0.0576	–	[34]
Stern89	0.0557	0.0135	[46]
BLP	0.0555	0.0148	[12]
MMT11	0.0537	0.0216	[39]
BJMM12	0.0494	0.0286	[9]
MO15	0.0473	0.0???	[40]; 空間計算量が明記されていないため, 0.0???とした
BM18	0.0465	0.0294	[14]

りも速く解くことが可能となる.

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

3.1.3.4 量子アルゴリズムへの耐性

現在のところ多項式時間で LPN 問題を解く量子アルゴリズムは提案されていない. しかし量子アルゴリズムを利用した攻撃の高速化方法を Kachigar と Tillich [33] が提案している.[*7] Esser, Kübler, May [21] は, BKW や Gauss アルゴリズムの変種を量子アルゴリズムで高速化できる点を指摘している.

3.1.3.5 現状の進展

格子の場合と同様に “Decoding Challenge” (`https://decodingchallenge.org/`) というウェブサイトが作成された.

1. $\mathbb{F}_2$ 係数の一様ランダムな線形符号に対するシンドローム復号問題
2. $\mathbb{F}_2$ 係数の一様ランダムな線形符号に対するハミング重みが小さい符号語を探索する問題

[*7] Kirshanova [31] が Kachigar と Tillich の結果 [33] の改良を提案していたが, 誤りがあったことが報告されている. そのため, 2018 年時点でのベストな量子アルゴリズムは Kachigar と Tillich [33] であると考えられる.

3. $\mathbb{F}_3$ 係数の一様ランダム線形符号に対するシンドローム復号問題
4. Goppa 符号を用いた Niederreiter 暗号の場合のシンドローム復号問題 (Classic McEliece の一方向性に対応 3.3.1)
5. QC-MDPC 符号に基づくシンドローム復号問題のチャレンジ (BIKE や HQC の一方向性に対応 3.3.2 3.3.3)

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

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

での解が得られている．1 の結果については, Narisada, Fukushima, Kiyomoto [41] を, 4 の結果については, Esser, May, Zweydinger [22] を参照されたい.

3.2 符号に基づく代表的な暗号方式

本節では, 符号に基づく代表的な暗号方式と署名方式の説明を行う．以下では, $\mathrm{GL}_k(\mathbb{F}_q)$ で k 次の $\mathbb{F}_q$ 要素正則行列全体がなす群を表す．また, S_n で n 次対称群を表す．S_n の要素である置換を $\mathrm{GL}_n(\mathbb{F}_q)$ 中の置換行列と同一視することとする.

3.2.1 McEliece 暗号

McEliece [38] が提案した古典的な暗号方式である．以下では $q = 2$ とする.

- k : 安全性パラメータ
- n : サンプルの個数
- τ : 誤差パラメータ (例: $\tau n = O(k)$)
- t : 誤り訂正符号の誤り訂正能力 ($t = \Omega(\tau n)$)

鍵生成: 誤り訂正能力が t である $[n, k]_2$-線形符号の生成行列 $\boldsymbol{G}$ を生成する．$\boldsymbol{S} \leftarrow \mathrm{GL}_k(\mathbb{F}_2)$ を一様ランダムに選ぶ．
$\boldsymbol{P} \leftarrow S_n$ を一様ランダムに選ぶ．$\tilde{\boldsymbol{G}} = \boldsymbol{SGP}$ とする．
公開鍵を $\tilde{\boldsymbol{G}}$ とし, 秘密鍵を $(\boldsymbol{S}, \boldsymbol{G}, \boldsymbol{P})$ とする.
暗号化: 平文を $\boldsymbol{m} \in \mathbb{F}_2^k$ とする．乱数 $\boldsymbol{e} \leftarrow \mathsf{Ber}_\tau^n$ を選び, 暗号文 $\boldsymbol{c} = \boldsymbol{m}\tilde{\boldsymbol{G}} + \boldsymbol{e}$ を計算する.
復号: $\hat{\boldsymbol{v}} = \boldsymbol{cP}^{-1}$ を計算する．$\hat{\boldsymbol{v}}$ を誤り訂正符号で訂正し $\boldsymbol{m}' \in \mathbb{F}_2^k$ を得る．$\boldsymbol{m} = \boldsymbol{m}'\boldsymbol{S}^{-1}$ を出力する.

復号の正当性は以下で確認される．$\boldsymbol{c} = \boldsymbol{m}\tilde{\boldsymbol{G}} + \boldsymbol{e}$ として, $\hat{\boldsymbol{v}} = \boldsymbol{cP}^{-1}$ を計算すると,

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

を得る．$\boldsymbol{mSG}$ は符号語であり, $\boldsymbol{e}\boldsymbol{P}^{-1}$ は誤りである．$\boldsymbol{e}\boldsymbol{P}^{-1}$ のハミング重みが t 以下であれば, 誤り訂正符号の復号により, $\boldsymbol{m}' = \boldsymbol{mS}$ を得る．よって, 高い確率で復号に成功する.

平文 $\boldsymbol{m}$ および $\tilde{\boldsymbol{G}}$ が一様ランダムであれば, 暗号文 $\boldsymbol{c}$ は LPN 仮定の下で擬似ランダムである．$\tilde{\boldsymbol{G}}$ が擬似ランダムであることを言うためには, McEliece 仮定と呼ばれる仮定が必要となる.

定義 3.9 (McEliece 仮定) $[n,k]_q$-符号のクラス $\mathcal{C}$ を固定する．攻撃者 $\mathcal{A}$ の優位性を

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

で定義する．任意の確率的多項式時間の攻撃者 $\mathcal{A}$ について，その優位性が無視できるほど小さいとき，McEliece 仮定が成立するという．

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

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

3.2.2 Niederreiter 暗号

Niederreiter [42] が 1986 年に提案した．のちに McEliece 暗号と「等価」であることが示された．詳しくは [35] を参照のこと．以下では $q = 2$ とする．

- k：安全性パラメータ
- n：サンプルの個数
- t：誤り訂正符号の誤り訂正能力

鍵生成: 誤り訂正能力が t である $[n,k]$ -線形符号のパリティ検査行列 $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ を生成する．$\boldsymbol{T} \leftarrow \mathrm{GL}_{n-k}(\mathbb{F}_2)$ を一様ランダムに選ぶ．$\boldsymbol{Q} \leftarrow S_n$ を一様ランダムに選ぶ．$\tilde{\boldsymbol{H}} = \boldsymbol{THQ}$ とする．
公開鍵を $\tilde{\boldsymbol{H}}$ とし，秘密鍵を $(\boldsymbol{T}, \boldsymbol{H}, \boldsymbol{Q})$ とする．

暗号化: 平文を $\boldsymbol{e} \in \mathcal{S}_H(n,t)$ とする．暗号文 $\boldsymbol{d} = \boldsymbol{e} \cdot \tilde{\boldsymbol{H}}^\top \in \mathbb{F}_2^{n-k}$ を計算する．

復号: $\hat{\boldsymbol{w}} = \boldsymbol{d} \cdot \boldsymbol{T}^{-\top}$ を計算する．$\hat{\boldsymbol{w}}$ を誤り訂正符号で訂正し復号し，誤りとして $\boldsymbol{e}'$ を得る．$\boldsymbol{e} = \boldsymbol{e}'\boldsymbol{Q}^{-\top}$ を出力する．

復号の正当性は以下で確認される．$\boldsymbol{d} = \boldsymbol{e} \cdot \tilde{\boldsymbol{H}}^\top$ として，$\hat{\boldsymbol{w}} = \boldsymbol{dT}^{-\top}$ を計算すると，

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

を得る．$\boldsymbol{eQ}^\top$ のハミング重みが t 以下であれば，誤り訂正符号の復号により，$\boldsymbol{e}' = \boldsymbol{eQ}^\top$ を得る．よって，高い確率で復号に成功する．

平文 $\boldsymbol{e}$ および $\tilde{\boldsymbol{H}}$ が一様ランダムであれば，暗号文 $\boldsymbol{d}$ は 判定版 LPN 仮定の下で擬似ランダムである．$\tilde{\boldsymbol{H}}$ が擬似ランダムであることを言うためには，McEliece 暗号で McEliece 仮定を考えたように，Niederreiter 仮定を考えればよい．

定義 3.10 (Niederreiter 仮定) $[n,k]_q$-符号のクラス $\mathcal{C}$ を固定する．攻撃者 $\mathcal{A}$ の優位性を

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

で定義する．任意の確率的多項式時間の攻撃者 $\mathcal{A}$ について，その優位性が無視できるほど小さいとき，Niederreiter 仮定が成立するという．

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

3.2.3 符号版 Lyubashevsky-Peikert-Regev (LPR)/Lindner-Peikert (LP) 暗号

鍵共有方式として考えられているが, 本稿では暗号方式として書く. McEliece 暗号では公開鍵の擬似ランダム性そのものを McEliece 仮定として導入していた. 一方, 符号版の Lyubashevsky-Peikert-Regev (LPR)/Lindner-Peikert (LP) 暗号では公開鍵の擬似ランダム性を判定版 LPN 仮定から示すことができる.

以下では $q = 2$ とする.

- k : 安全性パラメータ
- $n = n_1 + n_2$: サンプルの個数
- ℓ : 平文長
- τ : 誤差パラメータ (例: $\tau n = O(\sqrt{k})$)
- t : 誤り訂正符号の誤り訂正能力 ($t = \Omega\big((\tau n)^2\big)$)

鍵生成: 誤り訂正能力が t である $[n_2, \ell]$ -線形符号の生成行列 $\boldsymbol{G}_c$ を生成する. $\boldsymbol{A} \leftarrow \mathbb{F}_2^{k \times n_1}$ とする. $\boldsymbol{X} \leftarrow \mathsf{Ber}_\tau^{n_1 \times n_2}$, $\boldsymbol{Y} \leftarrow \mathsf{Ber}_\tau^{k \times n_2}$ とし, $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{Y} \in \mathbb{F}_2^{k \times n_2}$ とする.
公開鍵を $\tilde{\boldsymbol{G}} = [\boldsymbol{A} \mid \boldsymbol{B}] \in \mathbb{F}_2^{k \times n}$ とし, 秘密鍵を $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{X})$ とする.

暗号化: 平文を $\boldsymbol{m} \in \mathbb{F}_2^\ell$ とする. 乱数 $\boldsymbol{s} \leftarrow \mathsf{Ber}_\tau^k$ と 乱数 $\boldsymbol{e} \leftarrow \mathsf{Ber}_\tau^n$ を選び, 暗号文 $\boldsymbol{c} = \boldsymbol{s}\tilde{\boldsymbol{G}} + \boldsymbol{e} + (\boldsymbol{0}_{n_1}, \boldsymbol{m}\boldsymbol{G}_c) \in \mathbb{F}_2^n$ を計算する.

復号: $\boldsymbol{d} = \boldsymbol{c} \cdot \binom{-\boldsymbol{X}}{\boldsymbol{I}_{n_2}}$ を計算する. $\boldsymbol{d}$ を誤り訂正符号で訂正し復号すると $\boldsymbol{m}$ を得る.

復号の正当性は以下で確認される. $\boldsymbol{c} = \boldsymbol{s}\tilde{\boldsymbol{G}} + \boldsymbol{e} + (\boldsymbol{0}_{n_1}, \boldsymbol{m}\boldsymbol{G}_c)$ なので, 前半部を $\boldsymbol{u} = \boldsymbol{s}\boldsymbol{A} + \boldsymbol{e}_1$, 後半部を $\boldsymbol{v} = \boldsymbol{s}\boldsymbol{B} + \boldsymbol{e}_2 + \boldsymbol{m}\boldsymbol{G}_c$ と書く.

$\boldsymbol{d} = \boldsymbol{c} \cdot \binom{-\boldsymbol{X}}{\boldsymbol{I}_{n_2}}$ を計算すると,

$$\boldsymbol{d} = \boldsymbol{v} - \boldsymbol{u}\boldsymbol{X} = \boldsymbol{s}\boldsymbol{B} + \boldsymbol{e}_2 + \boldsymbol{m}\boldsymbol{G}_c - \boldsymbol{s}\boldsymbol{A}\boldsymbol{X} - \boldsymbol{e}_1\boldsymbol{X} = \boldsymbol{m}\boldsymbol{G}_c + (\boldsymbol{e}_2 - \boldsymbol{e}_1\boldsymbol{X} + \boldsymbol{s}\boldsymbol{Y})$$

を得る. $\boldsymbol{m}\boldsymbol{G}_c$ は符号語であり, $\boldsymbol{e}_2 - \boldsymbol{e}_1\boldsymbol{X} + \boldsymbol{s}\boldsymbol{Y}$ は誤りベクトルである. よって, $\boldsymbol{e}_2 - \boldsymbol{e}_1\boldsymbol{X} + \boldsymbol{s}\boldsymbol{Y}$ のハミング重みが t 以下であれば, 誤り訂正符号の復号により, $\boldsymbol{m}$ を得る. 高い確率で $\boldsymbol{e}_2 - \boldsymbol{e}_1\boldsymbol{X} + \boldsymbol{s}\boldsymbol{Y}$ のハミング重みが t 以下になるように τ を設定しているため, 高い確率で復号に成功する.

$\tilde{\boldsymbol{G}}$ が一様ランダムであれば, 暗号文 $\boldsymbol{c}$ は判定版 LPN 仮定の下で擬似ランダムである. $\tilde{\boldsymbol{G}}$ が擬似ランダムであることを言うためには, $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{Y}$ が擬似ランダムであればよい. これは, パラメータを変更した 判定版 LPN 仮定の下, 成立する.

3.2.4 CFS 署名

CFS 署名は Courtois, Finiasz, Sendrier が 2001 年に提案した署名である [17]. のちに, 安全性証明に用いられた仮定 (Niederreiter 仮定) が提案パラメータセットでは成り立たないことが示された [23, 24]. しかし後の方式に大きな影響を与えたため, ここに記す. Niederreiter 暗号を思い出すと, 秘密鍵を持っている場合, ハミング重みが t 以下の誤りは訂正できる. 一方, 訂正可能なシンドロームの集合 $\{\boldsymbol{e}\tilde{\boldsymbol{H}} \in \mathbb{F}_2^{n-k} \mid \boldsymbol{e} \in \mathcal{S}_H^{\leq}(n, t)\}$ のサイズは $\mathbb{F}_2^{n-k}$ のサイズに比べれば圧倒的に少ない.

2.2.5 節のように Hash-and-Sign に基づいた構成を考える. メッセージ M のハッシュ値を シンドローム $\boldsymbol{u} \in \mathbb{F}_2^{n-k}$ と捉えた場合, 正しく復号できないシンドロームになることが多い. そこで CFS 署名では, ハッシュ値を $\boldsymbol{u} = \mathsf{Hash}(M, i)$

と i をインクリメントしながら計算し, ハッシュ値が $\{\boldsymbol{e}\tilde{\boldsymbol{H}} \in \mathbb{F}_2^{n-k} \mid \boldsymbol{e} \in \mathcal{S}_H^{\leq}(n,t)\}$ に入るものを採用する.

署名鍵と検証鍵: パリティ検査行列 $\tilde{\boldsymbol{H}} \in \mathbb{F}_2^{(n-k)\times n}$ を検証鍵とする. また署名鍵を用いると, ハミング重み t 以下の符号語を訂正できることとする.

署名: 文書 M について,

1. $i = 0$ とする
2. $\boldsymbol{u} = \mathsf{Hash}(M, i)$ を計算する
3. ハミング重み t 以下の $\boldsymbol{e}$ で, $\boldsymbol{e} \cdot \tilde{\boldsymbol{H}}^\top = \boldsymbol{u}$ となるものを計算する. なければ $i \leftarrow i+1$ としてステップ 2 に戻る
4. $\sigma = (\boldsymbol{e}, i)$ を出力する.

検証: 文書 M と $\sigma = (\boldsymbol{e}, i)$ について, $\mathsf{HW}(\boldsymbol{e}) \leq t$ と $\boldsymbol{e} \cdot \tilde{\boldsymbol{H}}^\top = \mathsf{Hash}(M, i)$ ならば, 受理する. そうでないならば, 不受理とする.

安全性の根拠として, 以下の 2 つの仮定を必要とする.

- Niederreiter 仮定: トラップドアが入っている $\tilde{\boldsymbol{H}}$ は一様ランダムな符号のパリティ検査行列と区別が付かない
- 探索版 SD 仮定: 探索版 SD 問題が困難

3.3 符号に基づく主要な暗号方式

本稿では以下の暗号方式を取り上げる. いずれも NIST PQC 標準化の中で第 4 ラウンドに進んだものである.

1. Classic McEliece: Niederreiter 暗号を採用し, 符号の構成が非常に保守的という観点からこれを取り上げる.
2. BIKE: Niederreiter 暗号的な構成を採用している. McEliece 暗号を採用, QC-MDPC 符号を用いて鍵を圧縮している, という観点からこれを取り上げる.
3. HQC: 符号版の LPR/LP 暗号を採用, Quasi-Cyclic 符号を用いて鍵を圧縮している, という特徴からこれを取り上げる.

表 3.3: 符号に基づく暗号の分類

文献	暗号化	鍵交換	署名
Classic McEliece [3]	○	○	–
BIKE [2]	○	○	–
HQC [1]	○	○	–

3.3.1 Classic McEliece

- 提案者: Albrecht, Bernstein, Chou, Cid, Gilcher, Lange, Maram, von Maurich, Misoczki, Niederhagen, Paterson, Persichetti, Peters, Schwabe, Sendrier, Szefer, Tjhai, Tomlinson, Wang.
- 基本方式の説明: Niederreiter 暗号方式に基づいている. 基本符号方式として $\mathbb{F}_2$ 上の Goppa 符号を利用している. (具体的な Goppa 符号の生成方法や符号化および復号の方法については提案方式の仕様書を参照のこと.)

表 3.4: Classic McEliece のパラメータ．公開鍵長，秘密鍵長，暗号文長の単位はそれぞれ Byte とする．

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

$q = 2^m$ とし, $n \leq q$ を用いる．2 以上の t を $mt < n$ となるように取り, $k = n - mt$ とする．

鍵生成: t 誤りを訂正できる Goppa 符号のパリティ検査行列 $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$ をランダムに生成する．組織符号化し, $\tilde{\boldsymbol{H}} = [\boldsymbol{I}_{n-k} \mid \boldsymbol{T}]$ とする．公開鍵を $\boldsymbol{T} \in \mathbb{F}_2^{(n-k)\times k}$ とする．復号鍵を符号生成に使ったパラメータ Γ (t 次のモニックな $\mathbb{F}_q$ 係数既約多項式および相異なる $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_q$) とする．

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

復号 $D(sk, \boldsymbol{c})$： ハミング重み t のベクトル $\boldsymbol{e}$ を復号する．

1. $\boldsymbol{c}$ に k 個ゼロを加え, $\boldsymbol{v} = (\boldsymbol{c}, \boldsymbol{0}_k) \in \mathbb{F}_2^n$ を考える
2. Goppa 符号の復号アルゴリズムを用いて, $\boldsymbol{v}$ と距離 t 以下にある符号語 $\boldsymbol{d}$ を計算する．(なければ $\perp$ を出力する)
3. $\boldsymbol{e} = \boldsymbol{v} + \boldsymbol{d}$ とする．
4. $\mathsf{HW}(\boldsymbol{e}) = t$ かつ $\boldsymbol{c} = \tilde{\boldsymbol{H}}\boldsymbol{e}$ ならば $\boldsymbol{e}$ を出力する．(そうでなければ $\perp$ を出力する.)

- 鍵カプセル化方式の説明: 基本方式を決定性の公開鍵暗号とみなし, $\mathsf{U}_m^{\not\perp}$ 変換をかけたものとみなせる．(第 3 ラウンドまでは $\mathsf{HU}_m^{\not\perp}$ を用いていた．以下ではハッシュ関数 $\mathsf{H}\colon \{0,1\}^* \to \{0,1\}^{256}$ を用いる．

鍵生成: ℓ ビットのシード δ から乱数を生成し, 鍵生成を行う．（乱数の生成方法は省略する.）公開鍵は同じく T である．復号鍵は Γ に加えて, n ビットの一様ランダムな文字列 s を用いる．

鍵カプセル化: 1. $e \leftarrow \mathcal{S}_H(n,t)$ をあるアルゴリズムに従ってランダム生成する．

2. $C = E(pk, e)$ を計算する．
3. $K = \mathsf{H}(1, e, C)$ とする．
4. 暗号文は C , セッション鍵は K となる．

デカプセル: 1. $C \in \mathbb{F}_2^{n-k}$ を入力とする．

2. $b = 1$ とする．
3. $e \leftarrow D(sk, C)$ とする．$e = \perp$ であれば, $b = 0$, $e = s$ と上書きする
4. $K = \mathsf{H}(b, e, C)$ を計算する．
5. K を出力する．

パラメータセットとして mceliece348864, mceliece348864f, mceliece460896, mceliece460896f, mceliece6688128, mceliece6688128f, mceliece6960119, mceliece6960119f, mceliece8192128, mceliece8192128f が提案されている．今回末尾に f が付くものは扱っていない（鍵長・暗号文長は f 無しのものと変わらない）. 表 3.4 に鍵カプセル化方式の鍵長および暗号文長, 想定セキュリティレベルをまとめた.

3.3.2 BIKE

- 提案者: Aragon, Barreto, Bettaieb, Bidoux, Blazy, Deneuville, Gaborit, Gueron, Güneysu, Aguilar Melchor, Misoczki, Persichetti, Sendrier, Tillich, Zémor, Vasseur, Ghosh, Richter-Brokmann.
- 基本方式の説明: Niederreiter 暗号方式に基づいている. 基本となる符号に QC-MDPC 符号を採用し, 公開鍵サイズを圧縮している. そのため, 鍵や暗号化は格子暗号の一種の NTRU 暗号と非常に近い形をしている点が特徴である. 以下では, $\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1)$ とする.

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

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

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

  1. ch_0 を計算する.
  2. QC-MDPC 符号の復号アルゴリズムを用いて, ch_0 をシンドロームとするベクトル (e_0, e_1) を計算する.
- 鍵カプセル化方式の説明: 基本方式を決定性公開鍵暗号方式とみなす. 基本方式とハッシュ関数 $\mathsf{L}\colon \{0,1\}^* \to \{0,1\}^{256}$ を用いて, 平文 $m \in \{0,1\}^{256}$ と乱数 (e_0, e_1) に対して $(c_0 = E(pk, (e_0, e_1)), c_1 = m \oplus \mathsf{L}(e_0, e_1))$ と暗号化を行う IND-CPA 安全な乱択公開鍵暗号を構成する. 鍵カプセル化方式は, この乱択公開鍵暗号に $\mathsf{FO}_m^{\not\perp}$ 変換を適用したものとみなせる. 以下ではハッシュ関数 $\mathsf{H}, \mathsf{L}\colon \{0,1\}^* \to \{0,1\}^{256}$ と $\mathsf{G}\colon \{0,1\}^* \to \mathcal{S}_H(2n, t)$ を用いる.

  鍵生成: 適切な長さのシード δ から乱数を生成し, 鍵生成を行う. (乱数の生成方法は省略する.) 公開鍵は同じく h である. 復号鍵は (h_0, h_1) に加えて, ℓ ビットの一様ランダムな文字列 s を用いる.

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

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

表 3.5 に鍵カプセル化方式の鍵長および暗号文長をまとめた. 3 つのパラメータセットがそれぞれレベル 1, 3, 5 相当として提案された.

3.3.3 HQC

- 提案者: Aguilar Melchor, Aragon, Bettaieb, Bidoux, Blazy, Deneuville, Gaborit, Persichetti, Zémor, Bos, Dion, Lacan, Robert, Veron.
- 基本方式の説明: 符号版の LPR/LP 暗号に基づき, 公開鍵暗号を構成している. $\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1)$ とする.

表 3.5: BIKE のパラメータ．公開鍵長，秘密鍵長，暗号文長の単位はそれぞれ Byte とする．

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

$n' = n_1 n_2$ とし, $[n', k]$ 線形符号 $\mathcal{C}$ を採用する．線形符号 $\mathcal{C}$ の符号化・復号アルゴリズムを $\mathsf{encode}, \mathsf{decode}$ とする．$n \geq n'$ を仮定する．以下では, 暗号文の第二要素 v を $\mathcal{R}$ 要素 (n ビットベクトル) として扱っているが, 実際には n' ビットに縮めて用いる．

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

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

復号 $D(sk, c)$: $c = (u, v)$ に対して, $\mathsf{decode}(v - ux)$ を出力する．

- 鍵カプセル化方式: 基本方式を乱択な公開鍵暗号とみなし, $\mathsf{HFO}^{\perp}$ 変換を適用したものとみなせる．以下ではハッシュ関数 $\mathsf{H}, \mathsf{H}' \colon \{0,1\}^* \to \{0,1\}^{256}$ を用いる．また, XOF [*8] として $\mathsf{H}_G \colon \{0,1\}^* \to \{0,1\}^*$ も用いる．(第 4 ラウンドで G への入力に seed と salt が追加された.)

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

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

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

3 つのパラメータセットがそれぞれレベル 1, 3, 5 相当として提案された．表 3.6 に鍵カプセル化方式の鍵長および暗号文長をまとめた．表中では, 秘密鍵はシードだけ記憶していることにされており, 40 バイトしかない．また公開鍵の a の部分もシードから再生成されることと定義されている点に注意されたい．

3.4 符号に基づく暗号技術に関するまとめ

基本となる McEliece 暗号方式は, McEliece により 40 年以上前に提案されており, パラメータは改訂されているものの, いまだに破られていない．Classic McEliece などのように, 公開鍵や秘密鍵は長いものの, 暗号文は短い方式が多

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

表 3.6: HQC のパラメータ．公開鍵長，秘密鍵長，暗号文長の単位はそれぞれ Byte とする．

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

い．LPN 問題は学習理論や符号理論から派生した問題であり, 誤り確率 τ が十分大きい場合の LPN 問題を確率的多項式時間で効率的に解くことは困難であると予想されている．

共通鍵や公開鍵の分野で多くの方式が LPN 問題に基づいて提案されている．LWE 問題と比較した場合, 利点としては,

- $\mathbb{F}_2$ およびその拡大体を基に構成するため, ハードウェア構成との相性が良い点
- 誤差分布としてベルヌーイ分布やその一般化した分布を用いるため, 誤差のサンプリングが容易である点

が挙げられる．一方, 欠点として,

- 鍵や暗号文のサイズが大きくなりやすい点
- 符号の復号アルゴリズムが複雑になりがちな点
- ID ベース暗号や完全準同型暗号といった発展的な応用が少ない点

が挙げられる．

暗号方式のパラメータ設定の際には, 3.1 節で挙げたさまざまなアルゴリズムを考慮する必要がある．アルゴリズムの高速化について盛んに研究されており, 動向を注視する必要がある．また, 攻撃に用いられるアルゴリズムの研究は理論的なものが多く, 攻撃実験報告は小さいパラメータに対して行ったものが多い．そのため, 攻撃実験に関する研究もこれから非常に重要である．

公開鍵や秘密鍵を圧縮しようと特殊な符号を採用したり, 距離の定義を変える提案も多くある．これらは解読攻撃を受けることも多く, 評価が確定していない暗号・署名方式については注視が必要である．

第 3 章の参照文献

[1] C. Aguilar Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J. Bos, J.-C. Deneuville, A. Dion et al. Hamming Quasi-Cyclic (HQC) – Fourth round version (Updated version 01/10/2022). `https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/HQC-Round4.zip`. (2023-04-07 閲覧)

[2] N. Aragon, P. S. L. M. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Ghosh et al. BIKE: Bit flipping key encapsulation (Round 4 submission). `https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/BIKE-Round4.zip`, 2022-10-10. (2023-04-07 閲覧)

[3] M. R. Albrecht, D. J. Bernstein, T. Chou, C. Cid, J. Gilcher, T. Lange, V. Maram, I. von Maurich. Classic McEliece: conservative code-based cryptography. `https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/mceliece-Round4.tar.gz`, 2022-10-23. (2023-04-07 閲覧)

[4] B. Applebaum, D. Cash, C. Peikert, A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. *CRYPTO 2009*, volume 5677 of LNCS, pp. 595–618, Springer, 2009.

[5] S. Arora, R. Ge. New algorithms for learning in presence of errors. *ICALP 2011*, Part I, volume 6755 of LNCS, pp. 403–415, Springer, 2011.

[6] M. Alekhnovich. More on average case vs approximation complexity. *Computational Complexity*, volume 20, pp. 755–786, Springer, 2011.

[7] É. Barelli, A. Couvreur. An efficient structural attack on NIST submission DAGS. *ASIACRYPT 2018*, volume 11272 of LNCS, pp. 93–118, Springer, 2018.

[8] A. Blum, M. L. Furst, M. J. Kearns, R. J. Lipton. Cryptographic primitives based on hard learning problems. *CRYPTO 1993*, volume 773 of LNCS, pp. 278–291, Springer, 1993.

[9] A. Becker, A. Joux, A. May, A. Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. *EUROCRYPT 2012*, volume 7237 of LNCS, pp. 520–536, Springer, 2012.

[10] A. Blum, A. Kalai, H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, volume 50, issue 4, pp. 506–519, ACM, 2003.

[11] D. J. Bernstein, T. Lange. Never trust a bunny. *RFIDSec 2012*, volume 7739 of LNCS, pp. 137–148, Springer, 2012.

[12] D. J. Bernstein, T. Lange, C. Peters. Smaller decoding exponents: Ball-collision decoding. *CRYPTO 2011*, volume 6841 of LNCS, pp. 743–760, Springer, 2011.

[13] L. Both, A. May. Optimizing BJMM with nearest neighbors: Full decoding in $2^{2n/21}$ and McEliece secu-

rity. *Workshop on Coding and Cryptography 2017*. https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/paper/bjmm+.pdf, 2017.

[14] L. Both, A. May. Decoding linear codes with high error rate and its impact for LPN security. *PQCrypto 2018*, volume 10786 of LNCS, pp. 25–46, Springer, 2018.

[15] E. R. Berlekamp, R. J. McEliece, H. C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transaction on Information Theory*, volume 24, issue 3, pp. 384–386, 1978.

[16] S. Bogos, S. Vaudenay. Optimization of LPN solving algorithms. *ASIACRYPT 2016*, Part I, volume 10031 of LNCS, pp. 703–728, Springer, 2016.

[17] N. Courtois, M. Finiasz, N. Sendrier. How to achieve a McEliece-based digital signature scheme. *ASIACRYPT 2001*, volume 2248 of LNCS, pp. 157–174, Springer, 2001.

[18] S. Devadas, L. Ren, H. Xiao. On iterative collision search for LPN and subset sum. *TCC 2017*, Part II, volume 10678 of LNCS, pp. 729–746, Springer, 2017.

[19] T. Debris-Alazard, J.-P. Tillich. Two attacks on rank metric code-based schemes: RankSign and an IBE scheme. *ASIACRYPT 2018*, volume 11272 of LNCS, pp. 62–92, Springer, 2018.

[20] A. Esser, F. Heuer, R. Kübler, A. May, C. Sohler. Dissection-BKW. *CRYPTO 2018*, Part II, volume 10992 of LNCS, pp. 638–666, Springer, 2018.

[21] A. Esser, R. Kübler, A. May. LPN decoded. *CRYPTO 2017*, Part II, volume 10402 of LNCS, pp. 486–514, Springer, 2017.

[22] A. Esser, A. May, F. Zweydinger. McEliece needs a break – Solving McEliece-1284 and quasi-cyclic-2918 with modern ISD. *EUROCRYPT 2022*, Part III, volume 13277 of LNCS, pp. 433–457, Springer, 2022.

[23] J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, J.-P. Tillich. A distinguisher for high rate McEliece cryptosystems. *IEEE Information Theory Workshop 2011*, pp. 282–286, IEEE, 2011.

[24] J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, J.-P. Tillich. A distinguisher for high-rate McEliece cryptosystems. *IEEE Transaction on Information Theory*, volume 59, issue 10, pp. 6830–6844, 2013.

[25] Q. Guo, T. Johansson, C. Löndahl. Solving LPN using covering codes. *ASIACRYPT 2014*, Part I, volume 8873 of LNCS, pp. 1–20, Springer, 2014.

[26] H. Gilbert, M. J. B. Robshaw, Y. Seurin. $\mathrm{HB}^{\#}$: Increasing the security and efficiency of HB^{+}. *EUROCRYPT 2008*, volume 4965 of LNCS, pp. 361–378, Springer, 2008.

[27] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, volume 48, issue 4, pp. 798-859, 2001.

[28] S. Heyse, E. Kiltz, V. Lyubashevsky, C. Paar, K. Pietrzak. Lapin: An efficient authentication protocol based on Ring-LPN. *FSE 2012*, volume 7549 of LNCS, pp. 346–365, Springer, 2012.

[29] 電子情報通信学会. **知識ベース 知識の森**, 1 群 (信号・システム) 2 編 (符号理論), https://ieice-hbkb.org/portal/doc_608.html. (2023-04-11 閲覧)

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

[31] E. Kirshanova. Improved quantum information set decoding. *PQCrypto 2018*, volume 10786 of LNCS, pp. 507–527, Springer, 2018.

[32] J. Katz, J. S. Shin, A. D. Smith. Parallel and concurrent security of the HB and HB^{+} protocols. *Journal of Cryptology*, volule 23, issue 3, pp. 402–421, Springer, 2010.

[33] G. Kachigar, J.-P. Tillich. Quantum information set decoding algorithms. *PQCrypto 2017*, volume 10346 of

LNCS, pp. 69–89, Springer, 2017.

[34] P. J. Lee, E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. *EUROCRYPT 1988*, volume 330 of LNCS, pp. 275–280, Springer, 1988.

[35] Y. Li, R. H. Deng, X. Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Transaction on Information Theory*, volume 40, issue 1, pp. 271–273, IEEE, 1994.

[36] É. Levieil, P.-A. Fouque. An improved LPN algorithm. *SCN 2006*, volume 4116 of LNCS, pp. 348–359, Springer, 2006.

[37] V. Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. *APPROX 2005, RANDOM 2005*, volume 3624 of LNCS, pp. 378–389, Springer, 2005.

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

[39] A. May, A. Meurer, E. Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054n})$. *ASIACRYPT 2011*, volume 7073 of LNCS, pp. 107–124, Springer, 2011.

[40] A. May, I. Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. *EUROCRYPT 2015*, Part I, volume 9056 of LNCS, pp. 203–228, Springer, 2015.

[41] S. Narisada, K. Fukushima, S. Kiyomoto. Multiparallel MMT : Faster ISD algorithm solving high-dimensional syndrome decoding problem. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences.*, volume E106-A, number 3, pp. 241–242, IEICE, 2023.

[42] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problemy Upravleniia i Teorii Informatsii (Problems of Control and Information Theory)*, volume 15, pp. 19–34, Akadémiai Kiadó, 1986.

[43] R. Nojima, H. Imai, K. Kobara, K. Morozov. Semantic security for the McEliece cryptosystem without random oracles. *Designs, Codes and Cryptography*, volume 49, issue 1-3, pp. 289–305, Springer, 2008.

[44] E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, volume 8, issue 5, pp. 5–9, IEEE, 1962.

[45] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, volume 56, issue 6, pp. 1–40, ACM, 2009.

[46] J. Stern. A method for finding codewords of small weight. *Coding Theory 1988: Coding Theory and Applications*, volume 388 of LNCS, pp. 106–113, Springer, 1989.

[47] B. Zhang, L. Jiao, M. Wang. Faster algorithms for solving LPN. *EUROCRYPT 2016*, Part I, volume 9665 of LNCS, pp. 168–195, Springer, 2016.

第 4 章

多変数多項式に基づく暗号技術

多変数公開鍵暗号[*1]（Multivariate Public Key Cryptosystems）における暗号方式の特徴として，有限体上の多変数多項式を用いた連立方程式

$$\begin{cases} p_1(x_1, x_2, \ldots, x_n) = 0, \\ p_2(x_1, x_2, \ldots, x_n) = 0, \\ \qquad\qquad \vdots \\ p_m(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

の求解問題（MP 問題）を解く計算の困難性が安全性の根拠として必要なことが挙げられる．連立線形方程式は多項式時間で求解可能であるから，多変数公開鍵暗号に現れる MP 問題における多項式の最大次数は 2 次以上に限定できる．本報告書では，多変数公開鍵暗号の多くの方式で採用されている双極型システムを中心に解説する．

4.1 多変数多項式に基づく暗号技術の安全性の根拠となる問題

$\mathbb{F}_q$ で位数 q の有限体を表し，$\mathbf{x} = (x_1, x_2, \ldots, x_n)$ で（代数的に独立な）変数の集合を表すものとする．$\mathbf{x}$ に関する $\mathbb{F}_q$ 上の多変数多項式の組，すなわち，多変数多項式 $p_i(\mathbf{x})$ $(i = 1, \ldots, m)$ により，$P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ と表されるものを（$\mathbb{F}_q$ 上の）多変数多項式系と呼ぶことにする．この多変数多項式系 $P(\mathbf{x})$ は代入評価により，$\mathbb{F}_q^n$ から $\mathbb{F}_q^m$ への写像を構成する．この（多変数多項式）写像を $P : \mathbb{F}_q^n \to \mathbb{F}_q^m$ と表すことにする．

4.1.1 MP 問題（MQ 問題）

MP 問題は次のように述べられる．

MP 問題 多変数多項式系 $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ と $\mathbf{d} = (d_1, d_2, \ldots, d_m) \in \mathbb{F}_q^m$ に対して，変数 $\mathbf{x}$ に関する連立方程式

$$\begin{cases} p_1(x_1, x_2, \ldots, x_n) = d_1, \\ p_2(x_1, x_2, \ldots, x_n) = d_2, \\ \qquad\qquad \vdots \\ p_m(x_1, x_2, \ldots, x_n) = d_m \end{cases} \tag{4.1}$$

の解（が存在するなら）少なくとも 1 つ求めよ．

[*1] かつては「多次多変数公開鍵暗号」と呼ばれていた．

連立方程式 (4.1) の右辺の各 d_i を左辺に移行して $p_i(\mathbf{x})$ に吸収させることができるので，右辺を 0 として MP 問題を表現する場合もある．MP 問題において，$P(\mathbf{x})$ の全ての成分 $p_i(\mathbf{x})$ が 1 次以下となる場合，MP 問題は単に線形方程式を解く問題となり，ガウスの消去法などで m, n に関し多項式時間で求解することが可能である．よって，MP 問題を考える場合は通常，各 $p_i(\mathbf{x})$ の次数は 2 以上であると仮定する．特に，$p_i(\mathbf{x})$ の次数が全て 2 となるとき，MP 問題は MQ 問題と呼ばれる．$\mathbb{F}_q = \mathbb{F}_2$ の場合，MQ 問題は NP 完全であることが知られている [20].

MQ 問題を解くコンテストとして Fukuoka MQ challenge が知られている．扱われている MQ 問題は，有限体は $q = 2, 31, 256$ の 3 種類と m, n に関しては $m = 2n,\ n \approx 1.5m$ の 2 種類の計 6 種類である．投稿され解かれた問題の (m, n) の値の最大は表 4.1 のようになっている．

表 4.1: Fukuoka MQ challenge で解かれた MQ 問題のパラメータの最大値（2022/9/30 時点）

タイプ	I	II	III	IV	V	VI
$\mathbb{F}_q$	$\mathbb{F}_2$	$\mathbb{F}_{31}$	$\mathbb{F}_{256}$	$\mathbb{F}_2$	$\mathbb{F}_{31}$	$\mathbb{F}_{256}$
(m, n)	$m = 2n$	$m = 2n$	$m = 2n$	$n \approx 1.5m$	$n \approx 1.5m$	$n \approx 1.5m$
(m, n) の最大	$(148, 74)$	$(74, 37)$	$(76, 38)$	$(69, 103)$	$(19, 28)$	$(20, 30)$

4.1.2 MP 問題を解く計算の計算量

MP 問題に対する一般的な解き方として，総当たり法や XL [32], Gröbner 基底攻撃 [11] が知られている．Gröbner 基底攻撃とは，イデアルの Gröbner 基底計算 [15, 16] を利用する MP 問題の解き方である．解きたい MP 問題の右辺の $\mathbf{d}$ を左辺に移行し，$P(\mathbf{x})$ の中に吸収させてしまうことにより，MP 問題は，

$$P(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})) = \mathbf{0}_m$$

の求解問題と表現できる．これは，イデアル $I = \langle p_1(\mathbf{x}), \ldots, p_m(\mathbf{x}) \rangle$ で定義される代数多様体の $\mathbb{F}_q$-有理点を求める問題と同値になる．さらに，イデアル I は $I' = \langle p_1(\mathbf{x}), \ldots, p_m(\mathbf{x}), x_1^q - x_1, \ldots, x_n^q - x_n \rangle$ に変更することができる．$p_{m+1}(\mathbf{x}) = x_1^q - x_1, \ldots, p_{m+n}(\mathbf{x}) = x_n^q - x_n$ と置く．これにより，$I' = \langle p_1(\mathbf{x}), \ldots, p_{m+n}(\mathbf{x}) \rangle$ と表すことができる．各 $p_i(\mathbf{x})$ $(i = 1, \ldots, m+n)$ に対し，その最高次斉次部分を $p_i^h(\mathbf{x})$ （d_i 次斉次多項式） と表し，$\mathbb{F}_q[\mathbf{x}]$ の斉次イデアルを J を

$$J = \langle p_1^h(\mathbf{x}), \ldots, p_{m+n}^h(\mathbf{x}) \rangle$$

で定める．$d \geq 0$ に対し， $\mathbb{F}_q[\mathbf{x}]_d$ で d-次斉次多項式のなす $\mathbb{F}_q[\mathbf{x}]$ の部分ベクトル空間を表し，$J_d := J \cap \mathbb{F}_q[\mathbf{x}]_d$ とする．次数環 $\mathbb{F}_q[\mathbf{x}]/J = \bigoplus_{d=0}^{\infty} \mathbb{F}_q[\mathbf{x}]_d/J_d$ の Hilbert 級数は

$$\mathrm{HS}_{\mathbb{F}_q[\mathbf{x}]/J}(t) = \sum_{d=0}^{\infty} \dim_{\mathbb{F}_q}(\mathbb{F}_q[\mathbf{x}]_d/J_d)\, t^d \in \mathbb{Z}[[t]] \quad \text{(形式的べき級数)}$$

で定義される．J の Krull-次元が 0，すなわち，J が $\mathbb{F}_q[\mathbf{x}]$ の極大イデアルとなるとき，$\mathrm{HS}_{\mathbb{F}_q[\mathbf{x}]/J}(t)$ は多項式となる．このとき，$d_{\mathrm{reg}} = \deg(\mathrm{HS}_{\mathbb{F}_q[\mathbf{x}]/J}(t)) + 1$ とおき，これを正則性の次数 (degree of regularity) と呼ぶ．これ以外にも，Gröbner 基底計算と関係のある不変量として，solving degree d_{sol} や first fall degree d_{ff} などが存在する [8, 10].

これらの不変量 d_{reg}, d_{sol}, d_{ff} は Gröbner 基底計算中に現れる多項式の次数の最大を評価する数である．一般に，これらの不変量の計算は Gröbner 基底計算と同程度困難であろうと考えられている．d をこれら不変量のうちの 1 つとしたとき，Gröbner 基底攻撃の計算量は以下のようになる [4]：

$$\mathcal{O}\left(\left(\begin{array}{c} n+d \\ d \end{array}\right)^{\omega}\right). \tag{4.2}$$

ここで，$2 \leq \omega \leq 3$ は線形方程式を解くために利用するアルゴリズムにより定まる定数である．

任意の $S(t) \in \mathbb{Z}[[t]]$ に対し，$[S(t)]_+ \in \mathbb{Z}_{>0}[[t]]$ で，$S(t)$ の最初に現れる非正係数の次数以降（この項も含む）を切り捨てた多項式を表すことにする．もし，

$$\mathrm{HS}_{\mathbb{F}_q[\mathbf{x}]/J}(t) = \left[\frac{\prod_{i=1}^{m+n}(1-t^{d_i})}{(1-t)^n}\right]_+ = \left[\left(\prod_{i=1}^{n}(1-t^{d_i})\right)\left(\frac{1-t^q}{1-t}\right)^n\right]_+$$

を満たすならば，$p_1(\mathbf{x}), \ldots, p_{m+n}(\mathbf{x})$ は半正則であるという．任意の m, n に対して, $p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})$ の係数をランダムに選ぶと，多くの場合に $p_1(\mathbf{x}), \ldots, p_{m+n}(\mathbf{x})$ は半正則となることが実験的に知られている．半正則であれば，正則性の次数 d_{reg} は容易に計算可能である．

XL は，単項式をすべて独立な変数と見て，MP 問題を線形方程式に変形して解く手法である．現れる線形方程式の行列部分は疎になる．実際，$p_i(\mathbf{x})$ $(i = 1, \ldots, m)$ が含む単項式の個数の最大を L とすると，行列の各行の非零成分の個数も L 個以下となる．

$$d_{\mathrm{XL}} = \deg\left(\left[\frac{\prod_{i=1}^{m}(1-t^{d_i})}{(1-t)^{n+1}}\right]_+\right) + 1$$

とおくと，q がある程度大きい場合，XL の計算量は以下のようになる．

$$\mathcal{O}\left(\left(\begin{array}{c} n+d_{\mathrm{XL}} \\ d_{\mathrm{XL}} \end{array}\right)^2 L\right).$$

4.1.3 MinRank 問題

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

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

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

MinRank 問題は HFEv- や Rainbow など様々な方式の安全性に関わっている．また，MinRank 問題を解く計算の困難性をベースとした署名方式などがいくつか提案されている [9, 3, 29, 1]．MinRank 問題は MP 問題に帰着できることが知られている [23, 17, 2].

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

$$SC = \sum_{i=1}^{k} x_i M_i \tag{4.3}$$

となるものが存在する．$\mathbf{r}_j$ を $\sum_{i=1}^{k} x_i M_i$ の第 j 行とすると，(4.3) より $\mathbf{r}_j$ は C の行ベクトルが張る空間に属する．よって，行列 $C'_j \in \mathbb{F}_q^{(r+1)\times n}$ を

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

で定めると，各 $j = 1, \ldots, m$ に対して，$\operatorname{Rank} C'_j \leq r$ を満たす．従って，C'_j の任意の $(r+1)\times(r+1)$ 小行列の行列式 $= 0$ という関係式が得られるが，このような関係式は j と小行列を動かすことにより $m\begin{pmatrix} n \\ r+1 \end{pmatrix}$ 個存在する．$\sharp T = r$ となる $T \subset \{1, 2, \ldots, n\}$ に対して，T に属する列番号からなる C の $r \times r$ 小行列を C_T と表し，さらにその行列式を c_T と表すことにすると，C'_j の任意の $(r+1)\times(r+1)$ 小行列の行列式は変数 $x_1, \ldots, x_k$ と c_T $(T \subset \{1, \ldots, m\},\ \sharp T = r)$ を用いて多項式で表すことができる．これらの変数の個数は $k + \begin{pmatrix} n \\ r \end{pmatrix}$ である．つまり，MinRank 問題は $k + \begin{pmatrix} n \\ r \end{pmatrix}$ 個の変数の $m\begin{pmatrix} n \\ r+1 \end{pmatrix}$ 個の方程式からなる MP 問題に帰着される．

4.1.4 IP 問題，EIP 問題

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

IP 問題 S, T をそれぞれ, $\mathbb{F}_q^n$, $\mathbb{F}_q^m$ 上のアフィン同型写像とする．多変数多項式系 $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ に対し，多変数多項式系 $\tilde{P}(\mathbf{x})$ を合成により，$\tilde{P}(\mathbf{x}) = T \circ P(\mathbf{x}) \circ S$ で定める．このとき，$P(\mathbf{x})$, $\tilde{P}(\mathbf{x})$ の情報から S, T を求めよ．

IP 問題において，S や T の行列成分やベクトル成分をすべて独立な変数と見た場合，等式 $\tilde{P}(\mathbf{x}) = T \circ P(\mathbf{x}) \circ S$ は連立多項式方程式と見ることができる．すなわち，IP 問題は MP 問題に変換される．

多変数多項式系のクラス $\mathcal{C}$ を 1 つ固定する．ここで多変数多項式系のクラスとは多変数多項式系の集合 $\mathbb{F}_q[\mathbf{x}]^m$ の部分集合のことである．このとき，（クラス $\mathcal{C}$ に関する）EIP（Extended Isomorphism of Polynomials）問題は以下のように述べられる．

EIP 問題 多変数多項式系 $\tilde{P}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ は，$\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T とクラス $\mathcal{C}$ に属する多変数多項式系 $P(\mathbf{x})$ により，$\tilde{P}(\mathbf{x}) = T \circ P(\mathbf{x}) \circ S$ で表されるとする．このとき，分解 $\tilde{P}(\mathbf{x}) = T' \circ P'(\mathbf{x}) \circ S'$ で，S', T' は $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像，$P'(\mathbf{x}) \in \mathcal{C}$ となるものを見つけよ．

$\mathcal{C} = \{P(\mathbf{x})\}$ に関する EIP 問題が通常の IP 問題であるから，EIP 問題は IP 問題の拡張である．4.2 節で見るように，EIP 問題は双極型システムで構成される暗号方式，署名方式の鍵復元攻撃に対する安全性に関わる．EIP 問題を解く方法はクラス $\mathcal{C}$ の取り方（あるいは方式）に依存する．

4.2 多変数多項式に基づく代表的な暗号方式

4.2.1 双極型システム

IP 問題ベース [25] や MinRank 問題ベース [9, 1, 3, 29] の方式も存在するが，多変数公開鍵暗号の多くの方式が MP 問題をベースとして構成されている．中でも双極型システム [11] と呼ばれる構成方法が多く利用されているため，こ

の構成方法について説明する．(1 次多項式で構成されてなくても）多変数多項式系 $P(\mathbf{x})$ によっては，多くの $\mathbf{d} \in \mathbb{F}_q^m$ に対して MP 問題が効率的に計算できる場合がある．例えば，$n = m$ とし，$P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ が三角型多変数多項式系である，すなわち，

$$
\begin{aligned}
&p_1(\mathbf{x}) = x_1,\\
&p_2(\mathbf{x}) = x_2 + g_2(x_1) \quad (g_2(x_1) \in \mathbb{F}_q[x_1]),\\
&p_3(\mathbf{x}) = x_3 + g_3(x_1, x_2) \quad (g_2(x_1, x_2) \in \mathbb{F}_q[x_1, x_2]),\\
&\qquad \vdots\\
&p_m(\mathbf{x}) = x_m + g_m(x_1, \ldots, x_{m-1}) \quad (g_m(x_1, \ldots, x_{m-1}) \in \mathbb{F}_q[x_1, \ldots, x_{m-1}])
\end{aligned}
$$

の形で表されるとすると，任意の $\mathbf{d} \in \mathbb{F}_q^m$ に対して $P(\mathbf{x}) = \mathbf{d}$ の（唯 1 つの）解が，x_1 から逐次的に求められることが分かる．このことはすなわち，多変数多項式系のクラス $\mathcal{C}$ を三角型多変数多項式系の全体で定めると，任意の $P \in \mathcal{C}$ に対して，$P(\mathbf{x}) = \mathbf{d}$ $(\mathbf{d} \in \mathbb{F}_q^m)$ の解が効率的に計算可能ということである．

双極型システムでは，まず，上の例のような MP 問題が効率的に計算できる多変数多項式系のクラス $\mathcal{C}_{\mathrm{cent}}$ を見つけ固定する．$G(\mathbf{x}) \in \mathcal{C}_{\mathrm{cent}}$ と $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T をそれぞれ任意にとり，これらを合成した多変数多項式系 $F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ をトラップドア付き一方向関数として利用するのが，双極型システムのアイデアである．ただし，$F(\mathbf{x})$ が実際にトラップドア付き一方向関数となるかどうかは $\mathcal{C}_{\mathrm{cent}}$ のとり方に依存する．

双極型システムの鍵生成は次のように行う．

鍵生成

1. $G(\mathbf{x}) \in \mathcal{C}_{\mathrm{cent}}$ をランダムに選ぶ.
2. $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T をランダムに選ぶ.
3. $F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ とする.

このとき，公開鍵は $F(\mathbf{x})$，秘密鍵は $G(\mathbf{x}), S, T$ となる．$F(\mathbf{x})$ はその係数集合が公開鍵として保管される．また，$G(\mathbf{x})$ を（この方式の）**中心写像**とよぶ．中心写像のクラス $\mathcal{C}_{\mathrm{cent}}$ は公開鍵長（や秘密鍵長）を出来るだけ小さくするために 2 次の多変数多項式系の部分集合で選ぶことが多い．双極型システムは暗号方式，署名方式両方の構成に用いることができる．

暗号方式の暗号化・復号は次のように行う．

暗号化　平文 $M \in \mathbb{F}_q^n$ に対し，$C = F(M)$ を計算する．C が暗号文となる．

復号　暗号文 $C \in \mathbb{F}_q^m$ に対し，(1) $B_1 = T^{-1}(C)$, (2) $G(B_2) = B_1$ となる B_2 を計算, (3) $M' = S^{-1}(B_2)$ の順に計算する．M' が平文と一致する．

復号が成功するためには，$G(\mathbf{x})$（あるいは $F(\mathbf{x})$）が単射である必要がある．単射の条件を少し緩めて，「$G(\mathbf{x})$（あるいは $F(\mathbf{x})$）の逆像の個数が十分少ない」とすることもできる．この場合，M' が複数得られることになるので，ハッシュ値などを用いて平文 M と一致する M' を特定する．

双極型システムの署名方式の署名生成・検証は次のように行う．

署名生成　メッセージ $M \in \mathbb{F}_q^m$ に対し，(1) $B_1 = T^{-1}(M)$, (2) $G(B_2) = B_1$ となる B_2 を計算, (3) $\sigma = S^{-1}(B_2)$ の順に計算する．σ が署名となる．

検証　署名 $\sigma \in \mathbb{F}_q^n$ に対し，$M' = F(\sigma)$ を計算する．$M = M'$ ならば署名を受理，それ以外は棄却する．

署名生成がいつでも実行できるためには，どのようなメッセージ $M \in \mathbb{F}_q^m$ に対しても，$B_2 = G^{-1}(B_1)$ の計算ができ

る，すなわち，$G(\mathbf{x})$（あるいは $F(\mathbf{x})$）が全射である必要がある．

双極型システムでは，中心写像のクラス $\mathcal{C}_{\mathrm{cent}}$ の取り方を変えることで幅広い方式の構成が可能である．例えば，$\mathcal{C}_{\mathrm{cent}} = \{$ 三角型多変数多項式系 $\}$ とすると暗号方式が得られる．双極型システムにおいて，$\mathcal{C}_{\mathrm{cent}}$ に関する EIP 問題がその安全性に大きく関わってくる．実際，EIP 問題を解けた場合，$F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ の代わりに分解 $F(\mathbf{x}) = T' \circ G'(\mathbf{x}) \circ S'$ を用いても，暗号方式における復号および，署名方式における署名生成（偽造）が実行可能となる．EIP 問題はクラス $\mathcal{C}$ の選び方に依存するので，$\mathcal{C}$ の選び方に応じて個々に解析される必要がある．例えば，$\mathcal{C}_{\mathrm{cent}} = \{$ 三角型多変数多項式系 $\}$ としたときの EIP 問題は効率的に解けることが知られている [19]．

双極型システムの代表的な構成法として，simple field 法と big field 法がある．Simple field 方式は中心写像の構成に $\mathbb{F}_q$ 以外の有限体を利用しない．Big field 方式は中心写像の構成に $\mathbb{F}_q$ の n 次拡大体 $\mathbb{F}_{q^n}$ を利用する．Big field 方式は中心写像を構成しやすいが，Gröbner 基底攻撃が効果的となる場合が多いという性質を持つ．以下では，big field 方式の代表として署名方式 HFE および HFEv-，simple field 方式の代表として署名方式 Rainbow について説明する．

4.2.2 双極型システムの modifier

Modifier [31, 11] は双極型システムの方式からその変種の方式を構成する．様々な Modifier があり，それぞれに安全性を強化したり，効率性を向上させたりといった効果がある．以下では代表的な modifier を 4 つ紹介する．4.2.3 節で説明する HFEv-方式では，2 つの modifier（マイナス手法と External Perturbation）が利用されている．

4.2.2.1 マイナス手法 "-"

マイナス手法は，公開鍵 $F(\mathbf{x})$ のいくつかの成分を削除する方法である．すなわち，$F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ と表されるとき，r 個の成分を削除し，$\tilde{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_{m-r}(\mathbf{x}))$ を新たな公開鍵とする方式である．$F(\mathbf{x})$ に対して有効な秘密鍵復元攻撃がある場合でも，$\tilde{F}(\mathbf{x})$ は $F(\mathbf{x})$ よりも情報が欠落しているため，$\tilde{F}(\mathbf{x})$ に対しては同じ秘密鍵復元攻撃が適用できなくなる可能性があり，安全性強化につながる．暗号方式では公開鍵の単射性が失われたり，復号が困難になるといった理由により，マイナス手法はあまり用いられない．署名方式では，$F(\mathbf{x})$ に対する署名生成を利用して，$\tilde{F}(\mathbf{x})$ に対する署名生成ができる．

4.2.2.2 プラス手法 "+"

プラス手法は，中心写像 $G(\mathbf{x})$ にランダムな多項式を成分として加える方法である．すなわち，中心写像 $G(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_m(\mathbf{x}))$ に対し，r 個のランダムな多項式 $g_{m+1}(\mathbf{x}), \ldots, g_{m+r}(\mathbf{x})$ を用意し，$\tilde{G}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_{m+r}(\mathbf{x}))$ を新たな中心写像とする方式である．中心写像に特定の構造を持たない多項式が混ざることにより，秘密鍵復元攻撃の成功率を下げることができる．署名方式では公開鍵の全射性が失われたり，署名生成が困難になるといった理由により，プラス手法はあまり用いられない．暗号方式では，復号において $\tilde{G}(\mathbf{x})$ の逆写像を計算しなければならないが，その計算に $G(\mathbf{x})$ の逆写像計算が利用できる．

4.2.2.3 External Perturbation "v"

この modifier は，元々の変数 $\mathbf{x} = (x_1, \ldots, x_n)$ に新たな変数 $\mathbf{v} = (x_{n+1}, \ldots, x_{n+v})$（vinegar 変数）を加える方法である．この modifier は主に署名方式で利用される．署名方式を定める中心写像のクラスを $\mathcal{C}$ とする．新たな中心写像のクラス $\mathcal{C}'$ を多項式写像 $G(\mathbf{x}, \mathbf{v}) : \mathbb{F}_q^{n+v} \to \mathbb{F}_q^m$ で，任意の $\mathbf{v}_0 \in \mathbb{F}_q^v$ に対し，$G(\mathbf{x}, \mathbf{v}_0) \in \mathcal{C}$ となるもの全体として定める．すると，$G(\mathbf{x}, \mathbf{v}) \in \mathcal{C}'$ に対し，$G(\mathbf{x}, \mathbf{v}) = \mathbf{d}$ $(\mathbf{d} \in \mathbb{F}_q^m)$ の解が次のように得られる．

1. $\mathbf{v}_0 \in \mathbb{F}_q^v$ をランダムに選ぶ．

2. $G(\mathbf{x}, \mathbf{v}_0) = \mathbf{d}$ を $\mathbf{x}$ に関して解く．（解を $\mathbf{x}_0$ とする．）
3. $(\mathbf{x}, \mathbf{v}) = (\mathbf{x}_0, \mathbf{v}_0)$ を出力．

この計算を利用して，新たな署名方式が構成できる．Vinegar 変数は $\mathcal{C}$ とは無関係な変数なので追加することで安全性強化が期待できる．

4.2.2.4 Internal Perturbation "I"

この modifier は，中心写像 $G(\mathbf{x})$ にノイズを加えて安全性を強化する方法である．変数 $\mathbf{z} = (z_1, \ldots, z_w)$ と多項式写像 $H(\mathbf{z}) : \mathbb{F}_q^w \to \mathbb{F}_q^m$，および，アフィン写像 $S : \mathbb{F}_q^n \to \mathbb{F}_q^w$ を用意する．また，$H(\mathbf{z})$ の像 $W \subset \mathbb{F}_q^m$ が分かっており，W に属する元の個数は十分少ないと仮定する．新たな中心写像 $\tilde{G}(\mathbf{x}) : \mathbb{F}_q^n \to \mathbb{F}_q^m$ を $G(\mathbf{x}) + H(S(\mathbf{x}))$ で定める．このとき，$\tilde{G}(\mathbf{x}) = \mathbf{d}$ $(\mathbf{d} \in \mathbb{F}_q^m)$ の解が次のように得られる．

1. $\mathbf{w}_0 \in W$ をランダムに選ぶ．
2. 次の方程式の解 $\mathbf{x}_0$ を求める．

$$\begin{cases} G(\mathbf{x}) = \mathbf{d} - \mathbf{w}_0, \\ H(S(\mathbf{x})) = \mathbf{w}_0. \end{cases}$$

   もし，解が得られなかったら 1. に戻る．
3. $\mathbf{x}_0$ を出力．

よって，この $\tilde{G}(\mathbf{x})$ を中心写像として方式が構成できる．

4.2.3 HFE 方式，HFEv-方式

4.2.3.1 暗号方式 HFE

$K = \mathbb{F}_{q^n}$ を $\mathbb{F}_q$ の n 次拡大体とし，$\mathbb{F}_q$-線形同型写像 $\phi : \mathbb{F}_q^n \xrightarrow{\sim} K$ を 1 つ固定する．D を正の整数として，K 上の 1 変数多項式 $\mathcal{G}(X)$ を次のようにとる．

$$\mathcal{G}(X) = \sum_{0 \le i \le j}^{q^i + q^j \le D} \alpha_{i,j} X^{q^i + q^j} + \sum_{0 \le i}^{q^i \le D} \beta_i X^{q^i} + \gamma \qquad (\alpha_{i,j},\ \beta_i,\ \gamma \in K)$$

$\mathcal{G}(X)$ の形の 1 変数多項式は HFE 多項式と呼ばれる．このとき，多変数多項式写像 $G : \mathbb{F}_q^n \to \mathbb{F}_q^n$ を $G = \phi^{-1} \circ \mathcal{G} \circ \phi$ と定めると，対応する多変数多項式系 $G(\mathbf{x})$ の成分は全て 2 次多項式となる．$\mathbf{d} \in \mathbb{F}_q^n$ に対して，$G(\mathbf{x}) = \mathbf{d}$ が解を持つならば，この解は全て効率的に計算することができる．実際，次の手順で計算できる．

1. $B = \phi(\mathbf{d}) \in K$ を計算する．
2. $A = \mathcal{G}^{-1}(B)$ を Cantor-Zassenhaus アルゴリズムなどの因数分解アルゴリズムを用いて計算する．
3. $\phi^{-1}(A)$ を計算する．

但し，**2** の計算が効率的に実行できるためには D をある程度小さくとる必要がある．上述のことを踏まえて，$\alpha_{i,j},\ \beta_i,\ \gamma \in K$ を動かしてできる $G(\mathbf{x})$ のなすクラス $\mathcal{C}_{\mathrm{HFE}}$ に対し，$\mathcal{C}_{\mathrm{cent}} = \mathcal{C}_{\mathrm{HFE}}$ の双極型システムは暗号方式を構成する．この暗号方式を HFE [25] と呼ぶ．HFE 自体は 1999 年，Kipnis と Shamir により効果的な攻撃が発見されている [23]．その後，HFE から派生した変種方式がいくつか提案されており，以下の HFEv- もその 1 つである．

4.2.3.2 署名方式 HFEv-

HFEv- [25] は，暗号方式 HFE を署名方式に応用したものである．HFE と同様に $\mathbb{F}_q$ の n 次拡大体 $K = \mathbb{F}_{q^n}$ をとり，$\mathbb{F}_q$-線形同型写像 $\phi : \mathbb{F}_q^n \to K$ を固定する．正の整数 a $(a < n)$ と v を固定する．まず，$\mathcal{G}(X)$ は次のように変更される．

$$\mathcal{G}(X) = \sum_{0 \le i \le j}^{q^i+q^j \le D} \alpha_{i,j} X^{q^i+q^j} + \sum_{0 \le i}^{q^i \le D} \beta_i(x_{n+1}, \ldots, x_{n+v}) X^{q^i} + \gamma(x_{n+1}, \ldots, x_{n+v}) \quad (\alpha_{i,j} \in K). \tag{4.4}$$

ここで，$\beta_i(x_{n+1}, \ldots, x_{n+v}), \gamma(x_{n+1}, \ldots, x_{n+v})$ は共に $\mathbb{F}_q^v$ から K への多項式写像であり，$\beta_i(x_{n+1}, \ldots, x_{n+v})$ は線形関数，$\gamma(x_{n+1}, \ldots, x_{n+v})$ は 2 次関数である．多変数多項式系 $G(\mathbf{x})$ は，多変数多項式写像 $G = \phi^{-1} \circ \mathcal{G} \circ (\phi \times id_v) : \mathbb{F}_q^{n+v} \to \mathbb{F}_q^n$ により定める．$\alpha_{i,j} \in K$ と $\beta_i(x_{n+1}, \ldots, x_{n+v})$, $\gamma(x_{n+1}, \ldots, x_{n+v})$ を動かしてできる $G(\mathbf{x})$ のなすクラスを $\mathcal{C}_{\text{HFEv-}}$ と定める．基本的には，これを $\mathcal{C}_{\text{cent}} = \mathcal{C}_{\text{HFEv-}}$ として構成される双極型システムを考えるのであるが，双極型システムを若干変更する．S は $\mathbb{F}_q^{n+v}$ 上のアフィン同型写像のままでよいが，T は $\mathbb{F}_q^n$ から $\mathbb{F}_q^{n-a}$ への最大ランクのアフィン写像と変更する．公開鍵は通常の双極型システムと同じように，$F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ と定める．よって，F は $\mathbb{F}_q^{n+v}$ から $\mathbb{F}_q^{n-a}$ への多変数多項式写像となる．メッセージ $M \in \mathbb{F}_q^{n-a}$ に対する署名 $\sigma = F^{-1}(M)$ は以下のよう計算される．

1. $\mathbf{c} = T^{-1}(M) \in \mathbb{F}_q^n$ （の 1 つ）を計算する．
2. $B = \phi(\mathbf{c}) \in K$ を計算する．
3. $B' \in \mathbb{F}_q^v$ をランダムに選び， $A = \mathcal{G}^{-1}(B \| B')$ を Cantor-Zassenhaus アルゴリズムなどを用いて計算する．$\mathcal{G}^{-1}(B \| B')$ が存在しない場合は，B' の選択からやり直す．
4. $\mathbf{e} = \phi^{-1}(A)$ を計算する．
5. $\sigma = S^{-1}(\mathbf{e})$ を計算する．

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

4.2.4 署名方式 Rainbow

署名方式 Rainbow [13] は，双極型システムを用いており，署名方式 UOV [21] を多層化した構造を持っている．UOV の詳細については，4.3.1 節で説明する．

正の整数 $t, v_1, o_1, \ldots, o_t$ に対し，$v_{i+1} = v_i + o_i$ により，$v_2, \ldots, v_{t+1}$ を順次定める．また，$i = 1, \ldots, t$ に対し，$S_i = \{1, \ldots, v_i\}$, $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ とおく．S_i の個数は v_i で，O_i の個数は o_i である．変数の個数を $n = v_{t+1}$，式数を $m = n - v_1$ とする多変数多項式系 $G(\mathbf{x}) = (g_{v_1+1}(\mathbf{x}), \ldots, g_n(\mathbf{x}))$ を次の形で与える：

$$g_k(x_1, \ldots, x_n) = \sum_{i \in O_h, j \in S_h} \alpha_{i,j}^{(k)} x_i x_j + \sum_{i,j \in S_h,\, i \le j} \beta_{i,j}^{(k)} x_i x_j + \sum_{i \in S_{h+1}} \gamma_i^{(k)} x_i + \eta^{(k)} \quad (k = v_1+1, \ldots, n).$$

但し，h は k が属する層番号，すなわち，“$k \in O_h$” で定まる整数 $1 \le h \le t$ である．$\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in \mathbb{F}_q$ を動かしてできる $G(\mathbf{x})$ のなすクラスを $\mathcal{C}_{\text{Rainbow}}$ と定め，これを Rainbow の中心写像のクラスとする．

署名生成

1. メッセージ $\mathbf{M} \in \mathbb{F}_q^m$ に対し，$\mathbf{c} = (c_{v_1+1}, \ldots, c_n) \leftarrow T^{-1}(\mathbf{M})$ とする．

2. $b_1, \ldots, b_{v_1} \in \mathbb{F}_q$ をランダムにとる．
3. $h = 1, 2, \ldots, t$ に対し，以下を実行：

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

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

   の解を計算し，それを $b_{v_h+1}, \ldots, b_{v_{h+1}}$ と置く．もし解がなければ 2 に戻る．
4. $\mathbf{b} \leftarrow (b_1, \ldots, b_n)$.
5. $\sigma \leftarrow S^{-1}(\mathbf{b})$. （$\sigma$ が署名となる．）

Rainbow は NIST PQC 標準化の第 3 ラウンドの候補に選ばれたが，Rainbow の EIP 問題を解くことにより，小さなサイズの UOV への攻撃に帰着する攻撃が提案され [5, 7]，その結果，安全性レベル 1, 3, 5 として提案されていたパラメータがその安全性レベルに到達しないことになり（143-bit 安全性が 69-bit 安全性に，207-bit 安全性が 157-bit 安全性に，272-bit 安全性が 206-bit 安全性に下がった），第 4 ラウンドに進むことはできなかった．

4.3 多変数多項式に基づく主要な暗号方式

MPKC で標準化が期待されるのは署名方式の UOV である．4.2.4 節で述べたように Rainbow は UOV を多層化したものであるが，その多層化が結果として暗号の強度を弱めることになった [5, 7]．しかし，多層化していない UOV は高い安全性を維持している．また，NIST の PQC の新たな署名方式の公募 [24] では，短い署名と効率的な検証を持つ方式を望んでおり，UOV はその性質を持つため標準化が期待される．

表 4.2: 多変数多項式に基づく暗号の分類

文献	暗号化	鍵交換	署名
UOV [21]			○

4.3.1 署名方式 UOV

4.3.1.1 UOV の概要

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

v, o を正の整数とし，$m = o$, $n = v + o$ とする．2 次多項式からなる多変数多項式系 $G(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_m(\mathbf{x}))$ を次の形で与える．

$$g_k(\mathbf{x}) = \sum_{\substack{1 \le i \le v \\ v+1 \le j \le n}} \alpha_{i,j}^{(k)} x_i x_j + \sum_{1 \le i \le j \le v} \beta_{i,j}^{(k)} x_i x_j + \sum_{1 \le i \le n} \gamma_i^{(k)} x_i + \eta^{(k)} \quad (k = 1, \ldots, m).$$

ここで，$\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in \mathbb{F}_q$ である．$G(\mathbf{x})$ は逆写像を効率的に計算することができる．具体的には，任意の $\mathbf{c} = c_i) \in \mathbb{F}_q^m$ に対し，$\mathbf{b} = G^{-1}(\mathbf{c})$（の一つ）が以下のように計算できる．

1. $b_1, \ldots, b_v \in \mathbf{F}_q$ をランダムにとる．
2. $g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})$ に $x_1 = b_1, \ldots, x_v = b_v$ を代入し，$x_{v+1}, \ldots, x_n$ に関する 1 次の多項式系 $\bar{g}_1(x_{v+1}, \ldots, x_n)$, $\ldots, \bar{g}_m(x_{v+1}, \ldots, x_n)$ を得る．1 次方程式

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

   の解を計算し，それを $b_{v+1}, \ldots, b_n$ と置く．もし解がなければ 1 に戻る．
3. $\mathbf{b} = (b_1, \ldots, b_n)$.

$\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in \mathbb{F}_q$ を動かしてできる $G(\mathbf{x})$ の集合を $\mathcal{C}_{\mathrm{UOV}}$ としたとき，$\mathcal{C}_{\mathrm{cent}} = \mathcal{C}_{\mathrm{UOV}}$ として構成される双極型システムの署名方式を UOV と呼ぶ．但し，アフィン同型写像 T は UOV の安全性には貢献しないので，通常は T は恒等写像で選ぶ．$\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$ を暗号学的ハッシュ関数とする．

鍵生成

1. $G(\mathbf{x}) \in \mathcal{C}_{\mathrm{UOV}}$ をランダムに選ぶ.
2. $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S をランダムに選ぶ.
3. S^{-1} を計算する．
4. $F(\mathbf{x}) = G(\mathbf{x}) \circ S$.

公開鍵は $F(\mathbf{x})$，秘密鍵は $G(\mathbf{x}), S^{-1}$ である．次に，署名生成である．メッセージを $M \in \{0,1\}^*$ とする．

署名生成

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

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

   の解を計算し，それを $b_{v+1}, \ldots, b_n$ と置く．もし解がなければ **3** に戻る．
4. $\mathbf{b} \leftarrow (b_1, \ldots, b_n)$.
5. $\sigma \leftarrow S^{-1}(\mathbf{b})$.

σ が署名となる．最後に検証である．

検証

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

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

4.3.1.2 UOV のパラメータ選択

UOV の設計に必要なパラメータは，有限体の位数 q，方程式数 m, 変数の個数 n である．PQC Forum [28] では，表 4.3 のように UOV のパラメータ見積もりが公開されている．

表 4.3: UOV のパラメータと安全性レベルの見積もり

(q, m, n)	安全性レベル	公開鍵サイズ	秘密鍵サイズ	署名サイズ
$(16, 64, 160)$	レベル 1	$3,297,280$ Bytes	$412,160$ Bytes	640 bits
$(256, 44, 112)$	レベル 1	$2,227,456$ Bytes	$278,432$ Bytes	896 bits
$(256, 72, 184)$	レベル 3	$9,803,520$ Bytes	$1,225,440$ Bytes	$1,472$ bits
$(256, 96, 244)$	レベル 5	$22,955,520$ Bytes	$2,869,440$ Bytes	$1,952$ bits

4.4 多変数多項式に基づく暗号技術に関するまとめ

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

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

第 4 章の参照文献

[1] G. Adj, L. Rivera-Zamarripa, J. A. Verbel. MinRank in the head: Short signatures from zero-knowledge Proofs. *IACR Cryptology ePrint Archive*, 2022/1501.

[2] M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. A. Perlner, D. Smith-Tone, J.-P. Tillich, J. A. Verbel. Improvements of algebraic attacks for solving the rank decoding and MinRank problems. *ASIACRYPT 2020*, Part I, volume 12491 of LNCS, pp. 507–536, Springer, 2020.

[3] E. Bellini, A. Esser, C. Sanna, J. Verbel. MR-DSS – Smaller MinRank-based (Ring-)signatures. *IACR Cryptology ePrint Archive*, 2022/973.

[4] M. Bardet, J.-C. Faugère, B. Salvy, B.-Y. Yang. Asymptotic expansion of the degree of regularity for semi-regular systems of equations. *Effective Methods in Algebraic Geometry (MEGA) 2005*.

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

[6] W. Beullens. MAYO: Practical post-quantum signatures from Oil-and-Vinegar Maps. *IACR Cryptology ePrint Archive*, 2021/1144.

[7] W. Beullens. Breaking Rainbow takes a weekend on a laptop. *CRYPTO 2022*, Part II, volume 13508 of LNCS, pp. 464–479, Springer, 2022.

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

[9] N. T. Courtois. Efficient zero-knowledge authentication based on a linear algebra problem MinRank. *ASIACRYPT 2001*, volume 2248 of LNCS, pp. 402–421, Springer, 2001.

[10] V. Dubois, N. Gama. The degree of regularity of HFE systems. *ASIACRYPT 2010*, volume 6477 of LNCS, pp. 557–576, Springer, 2010.

[11] J. Ding, J. E. Gower, D. S. Schmidt. Multivariate public key cryptosystems. *Advances in Information Security 25*, Springer, 2006.

[12] J. Ding, T. J. Hodges. Inverting HFE systems is quasi-polynomial for all fields. *CRYPTO 2011*, volume 6841 of LNCS, pp. 724–742, Springer, 2011.

[13] J. Ding, D. Schmidt. Rainbow, a new multivariable polynomial signature scheme. *ACNS 2005*, volume 3531 of LNCS, pp. 164–175, Springer, 2005.

[14] J. Ding, B.-Y. Yang, C.-H. O. Chen, M.-S. Chen, C. M. Cheng. New differential-algebraic attacks and reparametrization of Rainbow. *ACNS 2008*, volume 5037 of LNCS, pp. 242–157, Springer, 2008.

[15] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, volume 139, pp. 61–88, North-Holland, 1999.

[16] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5).

ISSAC'02, pp. 75–83, ACM, 2002.

[17] J.-C. Faugère, M. S. El Din, P.-J. Spaenlehauer. Computing loci of rank defects of linear matrices using Gröbner bases and applications to cryptology. *ISSAC'10*, pp. 257–264, ACM, 2010.

[18] H. Furue, Y. Ikematsu, Y. Kiyomura, T. Takagi. A new variant of unbalanced Oil and Vinegar using quotient ring: QR-UOV. *ASIACRYPT 2021*, Part IV, volume 13093 of LNCS, pp. 187–217, Springer, 2021.

[19] L. Goubin, N. T. Courtois. Cryptanalysis of the TTM cryptsystem. *ASIACRYPT 2000*, volume 1976 of LNCS, pp. 44–57, Springer, 2000.

[20] M. R. Garay, D. S. Johnson. Computers and intractability; A guide to the theory of NP-completeness. W.H.Freeman & Co., 1990.

[21] A. Kipnis, L. Patarin, L. Goubin. Unbalanced Oil and Vinegar schemes. *EUROCRYPT 1999*, volume 1592 of LNCS, pp. 206–222, Springer, 1999.

[22] A. Kipnis, A. Shamir. Cryptanalysis of the Oil and Vinegar signature scheme. *CRYPTO 1998*, volume 1462 of LNCS, pp. 257–266, Springer, 1998.

[23] A. Kipnis, A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. *CRYPTO 1999*, volume 1666 of LNCS, pp. 19–33, Springer, 1999.

[24] NIST, Standardization of additional digital signature schemes, call for proposals. `https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf`

[25] J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. *EUROCRYPT 1996*, volume 1070 of LNCS, pp. 33–48, Springer, 1996.

[26] J. Patarin. The Oil and Vineger signature scheme. Presented at *Dagstuhl Workshop on Cryptography*, 1997-09.

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

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

[29] B. Santoso, Y. Ikematsu, S. Nakamura, T. Yasuda. Three-pass identification scheme based on MinRank problem with half cheating probability. *arXiv*: 2205.03255, 2022.

[30] C. Tao, A. Petzoldt, J. Ding. Efficient key recovery for all HFE signature variants. *CRYPTO 2021*, Part I, volume 12825 of LNCS, pp. 70–93, Springer, 2021.

[31] C. Wolf. Taxonomy of public key schemes based on the problem of multivariate quadratic equations. *IACR Cryptology ePrint Archive*, 2005/077.

[32] B.-Y. Yang, J.-M. Chen. All in the XL family: Theory and practice. *ICISC 2004*, volume 3506 of LNCS, pp. 67–86, Springer, 2005.

第 5 章

同種写像に基づく暗号技術

本章では同種写像に基づく暗号技術についてまとめる．同種写像に基づく暗号技術の安全性は, 同種写像問題を解く計算の困難性及び（それと同値な）自己準同型環計算問題の困難性に依存している．そして，本報告書の 2018 年度版と比べて，特に，自己準同型環計算問題に基づく署名構成に進展が見られている（5.3.1 節参照）．

5.1 節では, 安全性の根拠となる問題として, 同種写像問題の一般形を述べた後, 最近発見された SIDH (Supersingular Isogeny Diffie–Hellman) 同種写像問題 [31] に対する解法について述べる．そして，その攻撃法を回避する計算問題として，同種写像に基づく一方向性群作用に関する計算問題，自己準同型環計算問題及び SQISign（Short Quaternion and Isogeny Signature）署名方式 [39] の安全性に関する計算問題の順に, その概要を記述していく．5.2 節では, 代表的な暗号方式として, 一方向性群作用に基づく CSIDH (Commutative Supersingular Isogeny Diffie–Hellman) 鍵共有 [20] とその変種, SIDH 型の鍵共有方式，CSIDH ベースの SeaSign 署名 [37]，CSI-FiSh (Commutative Supersingular Isogeny Fiat–Shamir) 署名 [6]，そして自己準同型環計算問題に基づく GPS (Galbraith–Petit–Silva) 署名 [49] を取り上げる．5.3 節では, 主要な暗号方式として，GPS 署名を改良した SQISign 署名方式を解説する．

本章では, 超特異楕円曲線を用いた暗号技術しか扱わない．しかし，通常楕円曲線に基づく CRS (Couveignes–Rostovtsev–Stolbunov) 鍵共有法 [26, 79] を改良した De Feo ら [40] の方式は, それ自体は実用的な性能にはまだ遠いが, 5.2.1.1 節で説明する CSIDH 鍵共有の原型を与えているという点で重要である．

同種写像の数学的詳細については, De Feo の概説記事 [36] や Washington の楕円曲線の教科書 [87] を，四元数環については Voight の教科書 [86] を参照のこと．また, Galbraith–Vercauteren による同種写像関連問題のサーベイ [50] も参照する．

■記法　$x \leftarrow_R X$ は, x を集合 X から一様ランダムにサンプリングすることを表す．以下では, 有限体上に定義された楕円曲線のみを扱い, 同種写像暗号では, 多くの場合, モンゴメリ型の楕円曲線定義式 $E_{a,b} : by^2 = x^3 + ax^2 + x$ が用いられる．標数 p の有限体 $\mathbb{F}$ 上定義された楕円曲線 E に対し, O_E は E の無限遠点であり, $\mathbb{F}$ の拡大体 $\mathbb{K}$ に対して, $\mathbb{K}$-有理点群は $E(\mathbb{K}) := \{(x, y) \in \mathbb{K}^2 \,|\, (x, y) \text{ は } E \text{ の定義式を満たす}\} \cup \{O_E\}$ で与えられる．また, E の r-ねじれ部分群は $E[r] := \{P \in E(\overline{\mathbb{F}}_p) \,|\, rP = O_E\}$ で与えられる．

5.1　同種写像に基づく暗号技術の安全性の根拠となる問題

同種写像問題の一般形を述べた後，最近発見された SIDH 同種写像問題に対する解法について述べ，そして，その攻撃法を回避する計算問題として，同種写像に基づく一方向性群作用に関する計算問題，自己準同型環計算問題と SQISign 署名方式の安全性に関する計算問題の順に, その概要及びそれら問題に対する解析状況について記述していく．

5.1.1 同種写像問題の一般形

同種写像とは, 2 つの楕円曲線 E, E' の間の写像 φ であり, E の座標 (x, y) の有理式で与えられると共に, 楕円曲線の加法構造に関する準同型性, 即ち $\varphi(P+Q) = \varphi(P) + \varphi(Q)$, を有する非零写像である. (その正確な定義は, 前掲の各文献を参照のこと.) また, E, E' の間に, 同種写像 φ が存在する時に, E と E' は同種であるという.

同種写像 φ は, その核 $C = \ker(\varphi)$ によって決まるので, φ の定義域曲線（始点曲線）E に対して φ の値域となる楕円曲線を E/C と書き表す, すなわち, $\varphi : E \to E/C$. 核 $C = \ker(\varphi)$ の位数がセキュリティパラメータ λ の多項式サイズであれば, $C = \ker(\varphi)$ となる φ を効率的に計算するアルゴリズムが Vélu によって与えられている [85]. (モンゴメリ型楕円曲線に対する Vélu の公式に関しては, [77] を参照のこと.) 特に核の位数 $\#C$ が小素数になる同種写像を同種写像基本演算として，それらの合成が同種写像暗号での基本的な暗号演算を与えることになる．そして, その合成における基本演算の組み合わせ方法が, 秘密鍵情報を与える.

つまり, 同種な楕円曲線の間の同種写像を計算することを要求する次の同種写像問題が, 具体的な暗号方式の安全性を根拠づける次節以降の諸問題の基本形となる．（超特異同種写像問題と自己準同型環計算問題との計算量的同値性に関しては 5.1.4 節で触れる．）

定義 5.1 (一般形同種写像問題 [50]) *2* つの同種な楕円曲線 E, E' に対して, 同種写像 φ を計算せよ. (φ のコンパクトな表現を与えよ.)

ここで, 「φ のコンパクトな表現」とは, 様々な表現方法が考えられる．例えば, $\deg(\varphi)$ が小素数 ℓ_i によって $\prod_i \ell_i^{e_i}$ となっている場合には, この分解に沿って φ を分解した各 ℓ_i 次同種写像の像に現れる値域楕円曲線 (又は j 不変量) の列挙で与えることができる．また，5.1.2 節にて後述する SIDH 同種写像問題（定義 5.2）の設定では, 核の生成点が, 同種写像のコンパクトな表現を与える．そして，5.2.1.1 節で与えられる CSIDH 鍵共有では虚 2 次整環 (オーダー) のイデアル類によって同種写像が表現される．SIDH 同種写像問題，5.1.3 節にて後述する CSIDH-(R)EGA-DL 問題（定義 5.3, 5.5）は，定義 5.1 の同種写像問題に基づいて定義される.

定義 5.1 において, φ の次数が多項式サイズであれば, この問題は簡単に解けるので, φ の次数は通常は指数サイズのものを考える．また, Galbraith ら [50] は, j 不変量を使って, この問題を定式化しているが, CSIDH 鍵共有では, $\mathbb{F}_p$-有理な楕円曲線のみを対象とするので, $\overline{\mathbb{F}}_p$-同型であるが $\mathbb{F}_p$-同型でないツイスト曲線を判別して扱う必要性が生じるため, 上ではあえて, より素朴な形を採用して, 2 つの同種な楕円曲線 E, E' を使って同種写像問題を提示した.

同種写像問題の初期の考察には, 自己準同型環計算を扱った Kohel の博士論文 [56] や Galbraith による同種写像問題に関する研究 [47] 及び Couveignes と Rostovtsev–Stolbunov による初期の暗号応用への提案 [26, 79] がある．その後, Charles らによる同種写像に基づいたハッシュ関数の提案 [19] は, 同種写像一方向性関数を一方向性の観点からだけでなく, 衝突困難性の観点からも見直すことになり, 初期の同種写像暗号の研究では重要な役割を果たした．特に, 同種写像グラフがエクスパンダーグラフであることに着目して暗号に応用した意義は大きい.

■超特異同種写像問題と通常同種写像問題　標数 p の有限体上の楕円曲線 E の p-ねじれ部分群 $E[p]$ が, $E[p] = \{O_E\}$ の時, E を超特異楕円曲線といい, そうでない時, E を通常楕円曲線という．超特異楕円曲線の j 不変量は, $\mathbb{F}_{p^2}$ の要素である．つまり, 超特異 j 不変量の個数は, 有限個であり, 具体的に $p/12 + \epsilon$ (但し $\epsilon \in \{0, 1, 2\}$) で与えられる．超特異, 通常という楕円曲線の性質は, 同種写像によって保存されるため, 同種写像問題も, この 2 つの性質によって, 超特異同種写像問題と通常同種写像問題という 2 つの問題に分類される.

■超特異同種写像問題の計算困難性　超特異同種写像問題は，5.1.3.2 節で述べられる CSIDH 鍵共有や CSI-FiSh 署名の安全性に関する計算問題の一般形であり，その計算困難性を評価することは重要である．また，自己準同型環計算問題との関係性については 5.1.4 節を参照のこと．

超特異同種写像問題の古典計算機による解読時間は $\tilde{O}(\sqrt{p})$，量子計算機による解読時間は $\tilde{O}(\sqrt[4]{p})$ と見積もられている．古典解読アルゴリズムは Galbraith [47] による中間一致攻撃で，解読時間は $\tilde{O}(\sqrt{p})$ であり，量子解読アルゴリズムは Biasse ら [4] によって時間計算量が $\tilde{O}(\sqrt[4]{p})$ の量子アルゴリズムが知られている．これは，$\mathbb{F}_p$ 上の超特異楕円曲線の同種写像問題に対する準指数時間量子アルゴリズム [17] と Grover アルゴリズムに基づく $\tilde{O}(\sqrt[4]{p})$ の道探索アルゴリズムを結合したものである．

また，Costello ら [21], Longa ら [63] による報告，Udovenko–Vitto [84] による$IKEp182 Challenge [25] 解読報告，Jaques–Schanck [53] による同種写像問題に対する（量子）安全性評価報告は，いずれも SIDH 鍵共有（及び SIKE 暗号化 [52]）法への攻撃として提案されているが，多くの部分は一般的な超特異同種写像問題に関する知見としても有効であることに注意する．

5.1.2　SIDH 同種写像問題に対する解法の最近の進展

SIDH 鍵共有（及び SIKE 暗号化）は，これまで同種写像暗号の中核方式と位置づけられてきたが，SIDH 同種写像問題（定義 5.2）に対して，2022 年に Castryck–Decru [11] に始まる一連の鍵導出解法が発表されて，暗号として完全に破れてしまった．しかし，これらの解法は，SIDH 同種写像問題という補助点の情報を入力に含む問題に対する解法であって，一般の同種写像問題（定義 5.1）には適用できないことに注意する．B-SIDH 鍵共有 [24]，Séta 暗号化方式 [38] も本攻撃法により同様に解読可能である．SIDH 鍵共有に対して修正を図ろうとする試みについては，5.2.2 節で述べる．

■SIDH 同種写像問題　SIDH 鍵共有の公開パラメータは $pp_{\mathtt{sidh}} := (\ell_{\mathtt{A}}, \ell_{\mathtt{B}}, e_{\mathtt{A}}, e_{\mathtt{B}}, f; E, P_{\mathtt{A}}, Q_{\mathtt{A}}, P_{\mathtt{B}}, Q_{\mathtt{B}})$ で与えられる．ここで，$p+1 = f \cdot \ell_{\mathtt{A}}^{e_{\mathtt{A}}} \ell_{\mathtt{B}}^{e_{\mathtt{B}}}$ で，p は素数，$\ell_{\mathtt{A}}, \ell_{\mathtt{B}}$ は 2 つの異なる小素数である（f は小さい正整数で，多くの場合 $f = 1$）．例えば，$\ell_{\mathtt{A}} = 2, \ell_{\mathtt{B}} = 3$．$E$ は，$\mathbb{F}_{p^2}$ 上定義された超特異楕円曲線であり，$P_{\mathtt{A}}, Q_{\mathtt{A}}$ は，$E[\ell_{\mathtt{A}}^{e_{\mathtt{A}}}]$ の基底，$P_{\mathtt{B}}, Q_{\mathtt{B}}$ は，$E[\ell_{\mathtt{B}}^{e_{\mathtt{B}}}]$ の基底である．

定義 5.2 (SIDH 同種写像問題 [31, 50]) *SIDH 鍵共有公開パラメータ $pp_{\mathtt{sidh}}$ と，そこで定義された E と $\ell_{\mathtt{B}}^{e_{\mathtt{B}}}$-同種な $E_{\mathtt{B}}$ と $P'_{\mathtt{A}}, Q'_{\mathtt{A}} \in E_{\mathtt{B}}[\ell_{\mathtt{A}}^{e_{\mathtt{A}}}]$ が与えられた時，$P'_{\mathtt{A}} = \varphi_{\mathtt{B}}(P_{\mathtt{A}}), Q'_{\mathtt{A}} = \varphi_{\mathtt{B}}(Q_{\mathtt{A}})$ となる次数 $\ell_{\mathtt{B}}^{e_{\mathtt{B}}}$ の同種写像 $\varphi_{\mathtt{B}} : E \to E_{\mathtt{B}}$ を計算せよ．特に，$\varphi_{\mathtt{B}}$ の核 $\ker(\varphi_{\mathtt{B}})$ の生成元 $R_{\mathtt{B}} \in E[\ell_{\mathtt{B}}^{e_{\mathtt{B}}}]$ を計算せよ．*

表 5.1 に，2022 年に発表された SIDH 同種写像問題（定義 5.2）に対する攻撃法に関して，その始点曲線に対する条件，時間計算量，及び公開されているソフトウェア実装例についてまとめた．Robert [78] に従って，攻撃に用いられるアーベル多様体の次元 $2g$ により「$2g$ 次元攻撃法」と呼んで分類している．Maino–Martindale 法及び Robert 法に関しては [71] に現在実装が進んでいると述べられている．特に，Maino–Martindale 法実装公開は近々行われるとのことであり，現在，表 5.1 のアルゴリズム・実装の研究は進展が著しい．

以下，表 5.1 に記載した各攻撃法に関して概説する．

■Castryck–Decru による鍵導出攻撃 [11]　始点曲線 E が特殊極値整環 $\mathcal{O}_0$ を自己準同型環にもつ場合に主に限られるが[*1]，アーベル曲面間の同種写像が分解するかどうかという事実を使って SIDH 問題にアプローチした最初の論文で

[*1] 特殊極値整環 $\mathcal{O}_0$ に関しては 5.1.4.2 節を参照のこと．

表 5.1: SIDH 鍵共有に対する $2g$ 次元攻撃法．Castryck–Decru の 2 次元攻撃法の時間計算量評価 [90] では GRH（一般化されたリーマン予想）を仮定している．また，Robert の 4 次元攻撃法の時間計算量に関しては，[78] のヒューリスティック仮定 4.4 の下で多項式時間で動作することを意味する．その詳細は [78] の命題 4.6 を参照のこと．

$2g$	提案論文	始点曲線 E	時間計算量	公開実装例
2	Castryck–Decru [11]	特殊極値曲線	多項式時間 [90]	[12, 72]
2	Maino–Martindale [64]	一般曲線	準指数関数時間	–
4	Robert [78]	一般曲線	ヒューリスティック仮定付き多項式時間	–
8	Robert [78]	一般曲線	多項式時間	–

ある．特筆すべきは，その公開実装 [12] によると，SIKE Challenge [25] に挙げられていたパラメータ $IKE217 や SIKE パラメータ SIKEp434, SIKEp503, SIKEp610, SIKEp751 をいずれも現実的な時間内で解読することに成功したことである．後続の実装報告 [71] では，NIST 安全性レベル 5 に相当するとされていたパラメータ SIKEp751 が通常の PC（Intel Core i7-9750H CPU）で 1-2 時間程度で解けることが報告されている．

以下に示す Kani の定理が鍵となっている： 秘密同種写像の次数 $N_{\mathtt{B}} := \ell_{\mathtt{B}}^{e_{\mathtt{B}}}$ と互いに素な次数 a の同種写像 $\alpha : E \to E'$ が与えられれば，α と SIDH 同種写像問題の入力から楕円曲線積の間の $(a + N_{\mathtt{B}}, a + N_{\mathtt{B}})$-同種写像 $F : E \times E'' \to E_{\mathtt{B}} \times E'$ が構成可能になる．ここで Castryck–Decru [11] は，$N_{\mathtt{A}} := \ell_{\mathtt{A}}^{e_{\mathtt{A}}}, \ell_{\mathtt{A}} = 2, \ell_{\mathtt{B}} = 3$ として $N_{\mathtt{A}} > N_{\mathtt{B}}$ の下で，$a := N_{\mathtt{A}} - N_{\mathtt{B}}$ が $a = a_1^2 + 4a_2^2$ と 2 整数 a_1, a_2 による表現をもつ場合に有効な攻撃法を始点曲線 E が特殊な場合に示した．更に，その攻撃では，$a + N_{\mathtt{B}} = (N_{\mathtt{A}} - N_{\mathtt{B}}) + N_{\mathtt{B}} = N_{\mathtt{A}} = 2^{e_{\mathtt{A}}}$ であるので，F は $(2^{e_{\mathtt{A}}}, 2^{e_{\mathtt{A}}})$-同種写像，つまり Richelot 同種写像の列となっている．Richelot 同種写像列の計算や Richelot 同種写像のなすグラフに関する研究はこれまでも種数 2 同種写像暗号と関連して行われてきており [81, 83, 57, 14, 45]，それらの研究と関連した形で Castryck–Decru 攻撃法が実現されている．

■Maino–Martindale による始点曲線に依らない攻撃法 [64]　Maino–Martindale [64] では，始点曲線に依らない攻撃法を実現するために，Kani の定理で必要な次数 a が平滑（smooth）になる場合の解析を進めて，De Feo の寄与も取り込んで準指数時間攻撃法を実現した．更に，Castryck–Decru 法では秘密同種写像 $\varphi_{\mathtt{B}}$ を徐々に部分的な同種写像を決定していく方法であったが，Maino–Martindale [64] は，より直接的に 1 度の Kani の定理適用により $\varphi_{\mathtt{B}}$ を見つけ出すアルゴリズムに改良した．

■Robert による SIDH 問題に対する多項式時間攻撃 [78]　Robert [78] では，楕円曲線 8 つの直積の間の同種写像 F を上述の Kani の定理の一般化より得て，その分解性を使って多項式時間が証明可能な SIDH 攻撃アルゴリズムを構成した．さらに，[78] では，既に表 5.1 で見たように Castryck–Decru 法，Maino–Martindale 法を「$2g$ 次元攻撃法」という形で統合した．それによりそれぞれの方法の利点と課題が分かりやすい形で把握できるようになった（表 5.1 参照）．

例えば，[78] の主結果は，上述のように 8 次元アーベル多様体間の同種写像計算に基づいた証明可能多項式時間「8 次元攻撃法」アルゴリズムの提示にあるが，一方，ヒューリスティック多項式時間の「4 次元攻撃法」アルゴリズム [78, 系 4.5, 命題 4.6] についても詳細な解析を与えており，実装可能性という点から貴重な考察が含まれている．例えば，これら攻撃法では Kummer モデルの座標が用いられるが，8 次元攻撃法では $2^8 = 256$ 個の座標が必要であったのが，4 次元攻撃法では $2^4 = 16$ 個の座標で十分になる．このように実際的見地からの 4 次元攻撃法の更なる検討は，Maino–Martindale 法の更なる検討と共に重要な今後の研究課題である．

また，Robert の論文 [78] においては，Petit [74] から始まり de Quehen ら [75] により発展させられた SIDH に対

する「ねじれ点攻撃（torsion point attack）」との関連性もまとめられており，さらに広い見地から補助点情報を使った攻撃法の全体像も概観できるようになっている．

5.1.3 同種写像に基づく一方向性群作用（暗号学的群作用）に関する計算問題

素体 $\mathbb{F}_p$ 上定義された超特異楕円曲線間の同種写像問題の困難性に基づく鍵共有法として，CSIDH 鍵共有 (5.2.1 節参照) が 2018 年になって Castryck らによって提案された [20]．その安全性の根拠となる計算問題を [26, 82] に従ってまとめる．以降では，整数 a, b $(a < b)$ に対して $\mathbb{Z}$ の部分集合 $[a, b]$ を $[a, b] := \{a, a+1, \ldots, b-1, b\}$ とする．

5.1.3.1 2 種の一方向性群作用：REGA と EGA

■CSIDH 鍵共有・CSI-FiSh 署名の公開パラメータ CSIDH 鍵共有で，公開パラメータは $pp_{\texttt{csidh}} := (\mathfrak{O}, (\mathfrak{l}_1, \mathfrak{l}_2, \ldots, \mathfrak{l}_n), E, B)$ で与えられる．ここで，$\mathfrak{O}$ は虚 2 次代数体の整環，$\mathfrak{l}_1, \mathfrak{l}_2, \ldots, \mathfrak{l}_n$ はノルムが小さい奇素数 ℓ_i になる $\mathfrak{O}$ の素イデアルで，ℓ_i は $(\ell_i) := \mathfrak{l}_i\bar{\mathfrak{l}}_i$ $(i = 1, 2, \ldots, n)$ と 2 個の異なる素イデアル $\mathfrak{l}_i, \bar{\mathfrak{l}}_i$ の積に分解している．そして $p+1 = 4 \cdot \ell_1 \cdots \ell_n$ とした時，p は素数である必要がある．小奇素数 ℓ_i は，例えば，$\ell_1 = 3, \ell_2 = 5, \ldots$ である．E は，$\mathbb{F}_p$ 上定義されて，$\mathfrak{O}$ を $\mathbb{F}_p$-自己準同型環にもつ超特異楕円曲線である．B は指数 e_i のノルムの上限値，すなわち $-B \leq e_i \leq B$ となる指数 e_i を CSIDH 鍵共有では使う．

CSI-FiSh 署名の公開パラメータには，イデアル類群 $\mathrm{cl}(\mathfrak{O})$ の構造計算がなされた $pp_{\texttt{csidh}}$ が含まれる．そして，$\mathrm{cl}(\mathfrak{O})$ の生成元の間の関係式の情報 $\mathcal{R}$ も付された公開パラメータ $pp_{\texttt{csi-fish}} := (pp_{\texttt{csidh}}, \mathcal{R})$ を用いることで，有限アーベル群 $\mathrm{cl}(\mathfrak{O})$ 上での効率的な一様サンプリングが可能になる．特にその代表的なパラメータである CSIDH-512 では，$\mathrm{cl}(\mathfrak{O})$ がノルム 3 のイデアル $\mathfrak{l}_1$ により生成される巡回群であることが [6] において示された．現在，イデアル類群 $\mathrm{cl}(\mathfrak{O})$ の構造計算は（古典）準指数時間必要であるので，CSIDH-512 のように生成元が既知の十分大きいイデアル類群を得ることは一般に困難で，公開パラメータ $pp_{\texttt{csi-fish}}$ 生成に対する大きな制約となっている．

■CSIDH・CSI-FiSh 群作用の抽象形としての REGA・EGA CSIDH 鍵共有，及び CSI-FiSh 署名での基本演算は，$\mathbb{F}_p$-自己準同型環に虚 2 次代数体の整環 $\mathfrak{O}$ をもつ楕円曲線集合 X に対する $\mathfrak{O}$ のイデアル類群 $G := \mathrm{cl}(\mathfrak{O})$ の群作用 $(g, x) \mapsto gx \in X$ (但し $g \in G, x \in X$) として理解できる．その群作用は，自由かつ推移的である．群作用の詳細については 5.2.1.1 節を参照のこと．その記法に従えば，CSIDH・CSI-FiSh における同種写像問題は，この群作用の (G に関する) 逆関数 $(x, gx) \mapsto g$ を計算する問題と理解できる．このことから，CSIDH・CSI-FiSh における群作用は，一方向性群作用（または暗号学的群作用）と呼ばれる [26, 82].

但し，CSIDH 鍵共有と CSI-FiSh 署名では，$pp_{\texttt{csidh}}$ と $pp_{\texttt{csi-fish}}$ の違いに応じて G からのサンプリング方法が異なる．CSI-FiSh 署名ではすでに述べたように G から効率的に一様サンプリングするのに対して，CSIDH では $[-B, B]^n \subset \mathbb{Z}^n$ から適切に選んだ $(e_1, e_2, \ldots, e_n)$ により計算した $\prod_{i=1}^{n} \mathfrak{l}_i^{e_i}$ によって G からのサンプリングを行う．

これらサンプリング方法の違いに基づいて，最近の一方向性群作用の研究 [1, 68] では，CSI-FiSh 署名の場合の一方向性群作用を EGA（Effective Group Action：有効群作用）と呼び，CSIDH 鍵共有の場合の群作用を REGA（Restricted Effective Group Action：制限的有効群作用）と呼んでいる．以降では，基本構成としての「CSIDH」を接頭辞に付けて，それぞれの一方向性群作用を CSIDH-EGA，CSIDH-REGA と呼ぶことにする．

5.1.3.2 CSIDH-(R)EGA 上の計算問題

■CSIDH-EGA 上の DL,CDH 計算問題とそれらの量子帰着同値性 CSIDH-EGA に関して離散対数問題（DL: Discrete Logarithm）にあたる基本問題は，以下の CSIDH-EGA-DL 問題であり，更に，それに基づいた CDH（Computational

Diffie–Hellmann）問題は, CSIDH-EGA-CDH 問題である．それらの問題は，それぞれ，CSIDH ベクトル化問題及び CSIDH 並列化問題と呼ばれることもある [26, 82, 22]．以下，イデアル類 $[\mathfrak{a}]$ の群作用 $[\mathfrak{a}]E$ に関しては 5.2.1.1 節を参照のこと．

定義 5.3 (CSIDH-EGA-DL 問題 [26, 20, 1]) *CSI-FiSh* 署名公開パラメータ $pp_{\texttt{csi-fish}}$ と, $\mathbb{F}_p$ 上定義されており $\mathbb{F}_p$-自己準同型環 $\mathfrak{O}$ をもつ超特異楕円曲線 $E, E_{\mathtt{A}}$ が与えられた時, $E_{\mathtt{A}} = [\mathfrak{a}]E$ となる $\mathfrak{O}$ のイデアル $\mathfrak{a}$ を計算せよ．但し, $\mathfrak{a}$ の E への作用が効率的に計算可能な場合に限る．例えば, $\mathfrak{a}$ が小さい次数のイデアル積で与えられる場合などである．

定義 5.4 (CSIDH-EGA-CDH 問題 [26, 20, 1]) *CSI-FiSh* 署名公開パラメータ $pp_{\texttt{csi-fish}}$ と, $\mathbb{F}_p$ 上定義されており $\mathbb{F}_p$-自己準同型環 $\mathfrak{O}$ をもつ超特異楕円曲線 $E, E_{\mathtt{A}} := [\mathfrak{a}]E, E_{\mathtt{B}} := [\mathfrak{b}]E$（但し, $\mathfrak{a}, \mathfrak{b}$ は群作用が効率的に計算できる $\mathfrak{O}$ のイデアル）が与えられた時, $[\mathfrak{ab}]E = [\mathfrak{b}]E_{\mathtt{A}} = [\mathfrak{a}]E_{\mathtt{B}}$ を計算せよ．

通常の DL 問題と CDH 問題の場合のように，CSIDH-EGA-CDH 問題を CSIDH-EGA-DL 問題に帰着させることができるが，[48] において，その逆，CSIDH-EGA-CDH 問題を解くオラクルを用いて CSIDH-EGA-DL 問題を解く多項式時間量子帰着アルゴリズムが提案されている．[48] の帰着では，CSIDH-EGA-CDH 問題を完全に解くオラクルを仮定していた．Montgomery ら [68] により有意な（non-negligible）確率で CSIDH-EGA-CDH 問題に答えるオラクルを用いても [48] の帰着が成り立つことが示された．

更に, [68] では, CSIDH-REGA 上では [48] の帰着結果が成り立たないことも示されており, 判定版の CSIDH-EGA-DDH 問題へ拡張する事に関しても否定的な結果が示されている．また，CSIDH-EGA-DL 問題と CSIDH-EGA-CDH 問題の古典帰着に関して，Castryck ら [22] は,（古典）一方向性の準同型写像が存在するという妥当な仮定の下に，一般には EGA-DL 問題を EGA-CDH 問題に古典帰着させることができないことを簡潔な反例によって示した．

■CSIDH-REGA 上の DL,CDH 計算問題　既に述べたように，現在, イデアル類群 $G := \mathrm{cl}(\mathfrak{O})$ の構造計算を多項式時間で行う (古典) アルゴリズムは知られていないため, G 上の一様分布からの効率的なサンプリング法も知られていない. よって, 近似的にその一様サンプリングを行う効率的な (秘密鍵) サンプリング法を用いて CSIDH 鍵共有は与えられる (5.2.1 節参照). それに従って, CSIDH-EGA-DL 問題と CSIDH-EGA-CDH 問題もそれぞれ修正されて, それらを CSIDH-REGA-DL 問題, CSIDH-REGA-CDH 問題として以下に与える．これらの問題も，前段落と同様に，CSIDH ベクトル化問題及び CSIDH 並列化問題と呼ばれることもある [26, 82, 22].

定義 5.5 (CSIDH-REGA-DL 問題 [20, 1]) *CSIDH* 鍵共有公開パラメータ $pp_{\texttt{csidh}}$ と, $\mathbb{F}_p$ 上定義されており $\mathbb{F}_p$-自己準同型環 $\mathfrak{O}$ をもつ超特異楕円曲線 E, 及び $[-B, B]^n \subset \mathbb{Z}^n$ から一様に選んだ $(e_1, e_2, \ldots, e_n)$ により $\mathfrak{a} := \prod_{i=1}^{n} \mathfrak{l}_i^{e_i}$ となる $\mathfrak{a}$ によって $E_{\mathtt{A}} = [\mathfrak{a}]E$ となる $E_{\mathtt{A}}$ が与えられた時, $\mathfrak{a}$ と同じイデアル類に属する $\mathfrak{a}'$, *i.e.*, $\mathfrak{a}' \in [\mathfrak{a}]$ を計算せよ．

定義 5.6 (CSIDH-REGA-CDH 問題 [20, 1]) *CSIDH* 鍵共有公開パラメータ $pp_{\texttt{csidh}}$ と, $\mathbb{F}_p$ 上定義されており $\mathbb{F}_p$-自己準同型環 $\mathfrak{O}$ をもつ超特異楕円曲線 $E, E_{\mathtt{A}} := [\mathfrak{a}]E, E_{\mathtt{B}} := [\mathfrak{b}]E$(但し, $\mathfrak{a}, \mathfrak{b}$ は共に, $[-B, B]^n \subset \mathbb{Z}^n$ から一様に選んだ $(e_1, e_2, \ldots, e_n)$ により $\prod_{i=1}^{n} \mathfrak{l}_i^{e_i}$ と表されるイデアル) が与えられた時, $[\mathfrak{ab}]E = [\mathfrak{b}]E_{\mathtt{A}} = [\mathfrak{a}]E_{\mathtt{B}}$ を計算せよ．

■CSIDH-(R)EGA-DL 問題の古典計算機による計算困難性　CSIDH-(R)EGA-DL 問題に関しては，$\mathbb{F}_p$ 上の超特異楕円曲線の同種写像問題に対する Delfs–Galbraith [29] の (古典) アルゴリズムを適用するのが, 漸近的解読時間が最速の古典アルゴリズムとされており，その解読時間は $\tilde{O}(\sqrt[4]{p})$ である．

■CSIDH-(R)EGA-DL 問題に対する準指数時間での量子攻撃　G の X への作用が自由かつ推移的であるなら, 群作用 DL 問題は, 隠れシフト問題に帰着されて, それは更に二面体群に関する隠れ部分群問題 (DHSP: Dihedral Hidden Shift Problem) に帰着する. DHSP には, 準指数時間で動く量子アルゴリズムが知られているので, 一般に一方向性群作用に基づいた暗号方式は, 量子計算機に対して準指数時間安全性しかもたない. つまり, Childs ら [17] による通常同種写像問題に対する量子準指数時間アルゴリズムは, CSIDH-(R)EGA-DL 問題に対しても有効であり, CSIDH 群作用に関する DL 問題に対する量子計算機による漸近的な解読時間は準指数関数 $L_p[1/2, \sqrt{3}/2]$ で与えられる. [*2]

実用的には, 漸近的ではないその正確な見積もりが, 与えられた安全性レベルを達成する p の bit 長を決めるのに重要である. まず, EUROCRYPT 2019 において, Bernstein ら [7] は, CSIDH 群作用を行う量子回路のサイズを具体的に見積もることで, 上述の準指数時間アルゴリズムが, 従来考えられていたより計算オーバーヘッドが大きいのでないか, つまり, 攻撃するのはより困難であろうと主張している.

更に, EUROCRYPT 2020 において, Bonnetain–Schrottenloher [8] と Peikert [73] により独立に 2 つの研究成果が報告された. [8] では, 詳細に CSIDH 攻撃量子アルゴリズムを検討して, これまで考えられていたより効率的に攻撃可能であると主張している. それにより, 彼らは, Castryck ら [20] が 56-bit 量子安全性レベルと主張していたパラメータが, 実際には 38 bits レベルの量子安全性しか確保できないのでないか, という試算を述べている. また, [8] では, Kuperberg の 2005 年の論文 [59] に基づいた攻撃法に関して特に詳細な解析がなされたが, Peikert [73] では, その後の DHSP 解法の進展 [76, 60] も取り入れた安全性評価の改善がなされた.

そして, Chávez-Saab ら [10] の最近の評価結果では, NIST 安全性レベル 1 を満たすために素数 p を 4096 bits または 5120 bits 程度に大きくする必要性が示されており, 安全性レベル 2 には 6144 bits, 安全性レベル 3 には 8192 bits または 9216 bits 程度の大きさが必要であるという評価結果も報告されている.

■CSIDH・CSI-FiSh パラメータ以外のイデアル類群作用 DDH 問題の古典解法　上で見た EGA 上の DL 問題と CDH 問題の同値性と関係した興味深い DDH（Decisional Diffie–Hellman）問題に関する研究成果が Castryck ら [27, 16] により発表された. ある種のイデアル類群作用においては, 虚 2 次整環における「種の理論（genus theory)」を用いて, その作用に関する DDH 問題を効率的に解くことができることが示された.

但し, CSIDH 鍵共有・CSI-FiSh 署名に対して重要なこととして, その攻撃が有効になるためにはイデアル類群 $G := \mathrm{cl}(\mathfrak{O})$ が 2-ねじれ点を持つ必要があり, $p \equiv 3 \bmod 4$ である CSIDH パラメータに対しては無効であることも示されている.

5.1.4　自己準同型環計算問題と SQISign 署名方式の安全性に関する計算問題

5.1.4.1　自己準同型環計算問題

同種写像暗号は, Kohel [56], Galbraith [47], Couveignes [26] らの先駆的研究にその起源をもつが, 特に , Kohel は有限体上の楕円曲線の自己準同型環を計算するアルゴリズムを探求しており, そのために楕円曲線の同種写像からなる「同種写像グラフ」の性質を見極めることから始めて, 目的とする自己準同型環計算を同種写像グラフ上のアルゴリズム構成に帰着していく. その後, Kohel–Lauter–Petit–Tignol [55] は, この「同種写像計算」と「自己準同型環計算」を並置しながら考察する視点を,「構成的 Deuring 対応」として計算論的観点から捉え直した（表 5.2 参照）. そこでは, 四元数環側での ℓ-同種写像道探索問題を解く KLPT アルゴリズムが鍵となるアルゴリズムである. そして, この構成的 Deuring 対応に基づき「同種写像計算」と「自己準同型環計算」の等価性が示されており [33, 34, 89], 現在,

[*2] ここで, $L_p[\alpha, c] := \exp\left((c + o(1))\,(\log p)^{\alpha}(\log\log p)^{1-\alpha}\right)$ とする.

自己準同型環計算問題の困難性に基づいた暗号構成の研究が進められている [49, 39, 41].

■自己準同型環計算問題とその超特異同種写像計算問題との同値性　以下の記述に関しては，例えば [61, 62] を参照する．有理数体 $\mathbb{Q}$ 上 $\{1, i, j, k\}$ を基底とするベクトル空間でありかつ $a, b \in \mathbb{Q}$ により $i^2 = a, j^2 = b, k = ij = -ji$ という積構造が入った $\mathbb{Q}$ 上の代数（環）を四元数環 $\mathcal{B}$ と呼ぶ．各素点 ν（素数または ∞）における $\mathbb{Q}$ の完備化 $\mathbb{Q}_\nu$ による $\mathcal{B} \otimes \mathbb{Q}_\nu$ が $\nu = p, \infty$ の時にのみ斜体（可除環）になる四元数環 $\mathcal{B} = \mathcal{B}_{p,\infty}$ を扱う．これを，$\mathcal{B}_{p,\infty}$ は p, ∞ の 2 点のみで分岐する四元数環であるといい，$\mathcal{B}_{p,\infty}$ は同型を除いて一意的に決まる．この同じ素数 p を標数とする有限体上の超特異楕円曲線 E の自己準同型写像がなす環 $\mathrm{End}(E)$ は E の自己準同型環と呼ばれて，$\mathrm{End}(E)$ は $\mathcal{B}_{p,\infty}$ の極大整環 $\mathcal{O}$ になっている[*3]．ここで，（四元数環の）整環とは $\mathbb{Z}$ 上階数 4 の加群でありかつ環であるものであり，極大整環とは，そのような整環の中で包含関係に関して極大になっているものを指す．この自己準同型環 $\mathrm{End}(E)$ を計算する以下の問題が基本である．

定義 5.7 (自己準同型環計算問題 [56]) 超特異楕円曲線 E が与えられて，E の自己準同型環 $\mathrm{End}(E)$ を計算せよ．

Eisenträger らの研究 [33, 34] により，超特異同種写像計算問題と（超特異）自己準同型環計算問題の間に多項式時間帰着による計算問題としての同値性が示された．そこではヒューリスティックな仮定が使われていたが，Wesolowski [89] は，一般化されたリーマン予想に基づいて，その同値性に対して厳密な証明を与えた．

5.1.1 節で，超特異同種写像問題の古典計算機による現在最速の解読時間は $\tilde{O}(\sqrt{p})$ と見積もられていたので，この同値性により，自己準同型環計算問題も同等の計算時間であるが，直接に，自己準同型環計算問題を解く研究も進められており，[34] において，$\tilde{O}(\sqrt{p})$ 時間の自己準同型環計算（古典）アルゴリズムが報告されている．

■Deuring 対応　自己準同型環計算問題で与えられる楕円曲線 E から極大整環 $\mathcal{O}$ への対応は，表 5.2 に掲げたように，楕円曲線に関する様々な概念から四元数環に関する概念への対応に拡張される．その詳細に関しては，例えば [62, 第 2 章] を参照していただきたいが，特に基本的な対応としては，同種写像 $\varphi : E \to E_1$ が，極大整環の間の同型 $\mathcal{O} \cong \mathrm{End}(E), \mathcal{O}_1 \cong \mathrm{End}(E_1)$ を通して，左 $\mathcal{O}$-整イデアルかつ右 $\mathcal{O}_1$-整イデアルである I_φ に対応していることである．これにより始点曲線 E を固定すると，同種写像 $\varphi : E \to E_1$ の終点曲線 E_1 が $\mathcal{O}$ のイデアル類と対応することがわかり，超特異 j 不変量 ($\in \mathbb{F}_{p^2}$) の集合がイデアル類集合 $\mathrm{cl}(\mathcal{O})$ と一対一に対応していることもわかる．

一般に表 5.2 に示されるように，幾何的な情報から成る楕円曲線側のデータと代数的な情報から成る四元数環側のデータの間に対応関係が存在しており，Deuring 対応と呼ばれる．自己準同型環計算問題（定義 5.7）は Deuring 対応に基づいた問題であり，楕円曲線側の超特異 j 不変量 $j(E)$ から対応する四元数環側の極大整環 $\mathcal{O} = \mathrm{End}(E)$ を計算する問題となっている．そして，この Deuring 対応は，5.2.4 節及び 5.3.1 節での暗号構成を理解する際にも重要な鍵となっている．

5.1.4.2　SQISign 署名の安全性に関する計算問題

次に，SQISign 署名の安全性を示すために必要な計算問題を述べる．

■SQISign 署名の健全性に関する計算問題　まずは，SQISign 署名の健全性（偽造不可能性）を示すための計算問題である超特異平滑自己準同型写像計算問題（SEP：Smooth Endomorphism Problem）を定義する．以下では，核が巡回群となる自己準同型写像を巡回自己準同型写像と呼ぶ．

定義 5.8 (超特異平滑自己準同型写像計算問題 [39, 62]) 超特異楕円曲線 E が与えられて，平滑な整数を次数にもつ E 上の（非自明な）巡回自己準同型写像を見つけよ．

[*3] 自己準同型写像は英語で endomorphism であるので，その全体を $\mathrm{End}(E)$ で表す．

表 5.2: Deuring 対応

楕円曲線側	四元数環側
超特異 j 不変量 $j(E) \in \mathbb{F}_{p^2}$ (の $\mathbb{F}_{p^2}/\mathbb{F}_p$-Galois 共役類)	$\mathcal{B}_{p,\infty}$ 内の極大整環 $\mathcal{O} = \mathrm{End}(E)$ の自己同型類（タイプ）
同種写像 $\varphi : E \to E_1$ で定まる (E_1, φ)	左 $\mathcal{O}$-整イデアルかつ右 $\mathcal{O}_1$-整イデアルである I_φ
自己準同型写像 $\theta \in \mathrm{End}(E)$	主イデアル $\mathcal{O}\theta$
同種写像の次数 $\deg(\varphi)$	イデアルのノルム $n(I_\varphi)$
双対同種写像 $\hat{\varphi}$	共役イデアル $\overline{I_\varphi}$
同じ定義域・値域の同種写像 $\varphi : E \to E_1, \psi : E \to E_1$	同値なイデアル $I_\varphi \sim I_\psi$
超特異 j 不変量 $j(E) \in \mathbb{F}_{p^2}$ の集合	イデアル類の集合 $\mathrm{cl}(\mathcal{O})$
同種写像の合成 $\tau \circ \rho : E \to E_1 \to E_2$	イデアル積 $I_{\tau\circ\rho} = I_\rho \cdot I_\tau$
N-同種写像の同型類	レベル N の Eichler 整環の類集合

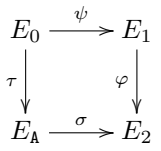
この問題で問うているような非自明な自己準同型写像が計算できれば，[34] で見るように，自己準同型環 $\mathrm{End}(E)$ 全体も計算できることが知られているので，この問題は，本質的に自己準同型環計算問題と同値である [39]．よって，$\tilde{O}(\sqrt{p})$ 時間での古典アルゴリズム [34] が現状最速と見積もられる．

■特殊極値的楕円曲線　次に，SQISign 署名の零知識性を示すための計算問題を述べるが，公開パラメータで重要となる楕円曲線 E_0 を示す．$p = 3 \bmod 4$ の時，j 不変量 $j = 1728$ となる $E_0 : y^2 = x^3 + x$ の $\mathcal{O}_0 = \mathrm{End}(E_0)$ は $i^2 = -1, j^2 = -p$ となる $\mathcal{O}_0 = \mathbb{Z} + \mathbb{Z}i + \mathbb{Z}\frac{i+j}{2} + \mathbb{Z}\frac{1+ij}{2}$ となることが知られている．更に具体的に自己準同型写像 $\iota : (x, y) \mapsto (-x, \sqrt{-1}y), \pi : (x, y) \mapsto (x^p, y^p)$ により $\mathrm{End}(E_0) = \mathbb{Z} + \mathbb{Z}\iota + \mathbb{Z}\frac{\iota+\pi}{2} + \mathbb{Z}\frac{1+\iota\pi}{2}$ で与えられる．

標数 p と ∞ のみで分岐する四元数環 $\mathcal{B}_{p,\infty} := \mathbb{Q}[i, j]$ における極大整環 $\mathcal{O}_0 \subset \mathcal{B}_{p,\infty}$ は，最小判別式の 2 次整環である $\mathfrak{O} \subset \mathcal{O}_0 \cap \mathbb{Q}[i]$ による $\mathfrak{O} + j\mathfrak{O} \subset \mathcal{O}_0$ が部分整環であり $\mathfrak{O} \subset (j\mathfrak{O})^\perp$ と直交分解しているとき[*4]，特殊極値的 (special extremal) であるという．詳細は [39, 62] を参照．$p = 7 \bmod 12$ の時，上述の E_0 に対して $\mathrm{End}(E_0)$ は特殊極値的であり，この時，E_0 は特殊極値的曲線と呼ばれる．特殊極値的曲線 E_0 は，その自己準同型環の構造が簡単で計算上扱いやすいため GPS 署名 及び SQISign 署名の公開パラメータの一部として必要である．

■SQISign 署名の零知識性に関する計算問題　SQISign 署名では，右図の同種写像 τ が秘密鍵で，超特異楕円曲線 $E_\mathtt{A}$ が公開鍵 (の主要な一部) である．署名生成では，同種写像 ψ, φ を適切に生成して得られた合成写像 $\varphi \circ \psi \circ \hat{\tau}$ を「ランダム化」した同種写像 σ を署名とする[*5]．[39, 41] において定義された E_0 を始点とする同種写像から成るある集合 $\mathcal{P}_{N_\tau}$ を τ によって $E_\mathtt{A}$ を始点とした同種写像に移した集合 $[\tau]_*\mathcal{P}_{N_\tau}$（$\mathcal{P}_{N_\tau}$ の τ による pushforward）を考える．正しく生成された署名同種写像 σ は $[\tau]_*\mathcal{P}_{N_\tau}$ に属するのであるが，それが $E_\mathtt{A}$ を始点とした 2 べき次数 $D\,(= 2^e)$ の巡回同種写像全体 $\mathtt{Iso}_{D,j(E_\mathtt{A})}$ から一様にサンプリングしたのと区別が付くかという問題が以下であり，SQISign 署名の零知識性を示すために必要である．

SQISign 同種写像図式



CSI-FiSh, GPS 図式と同様に可換図式ではない.

[*4] $\mathcal{B}_{p,\infty}$ における内積は $\alpha, \beta \in \mathcal{B}_{p,\infty}$ に対して $\frac{1}{2}\mathrm{tr}(\alpha\bar{\beta})$ で与えられて，ここは，その内積に関する直交分解である ($\mathcal{B}_{p,\infty}$ 内のトレース，共役の定義は，例えば [62] を参照のこと).

[*5] $\hat{\tau}$ は τ の双対同種写像である．表 5.2 も参照のこと.

定義 5.9 (SQISign 署名のランダム識別問題 [39, 62]) $\tau : E_0 \to E_{\mathtt{A}}$ を秘密同種写像として，楕円曲線 E_0 を含む *SQISign* 署名の公開パラメータ $pp_{\mathtt{sqisign}}$（詳しくは *5.3.1* 節参照）と公開鍵 $E_{\mathtt{A}}$ が入力として与えられると共に，$[\tau]_*\mathcal{P}_{N_\tau}$ から一様サンプリングして返すオラクル O_τ への多項式回のアクセスが許される時に，$E_{\mathtt{A}}$ を始点とする同種写像 σ が与えられて σ が $\mathtt{Iso}_{D,j(E_{\mathtt{A}})}$ から一様に選ばれたか，$[\tau]_*\mathcal{P}_{N_\tau}$ から一様に選ばれたかを判定せよ．

SQISign 署名の提案者によると，現在のところ，SQISign 署名のランダム識別問題を解くのに，E_0 と $E_{\mathtt{A}}$ の情報から τ を暴く攻撃法より効率の良い攻撃法はまだ知られていないとのことである [39, 62]．つまり，$\tilde{O}(\sqrt{p})$ 時間を必要とすると見積もられている．

また，上述の SQISign 署名に関する計算問題は，どちらも補助点を問題に含まないことにより，5.1.2 節で見た最近の SIDH 同種写像問題に対する攻撃法が適用できないことに注意する．

5.2 同種写像に基づく代表的な暗号方式

これまで長い間，補助点付きの SIDH 鍵共有に対する安全性が懸念されてきた．2022 年に Castryck らの攻撃により，その懸念が顕在化したが，これまでにも，補助点付きの方式への懸念から，補助点無しの方式を探求する研究が行われてきた．補助点無し暗号方式の代表例が，以下の CSIDH 鍵共有とその署名形，GPS 署名・SQISign 署名である．

5.2.1 CSIDH 鍵共有とその変種

5.2.1.1 CSIDH 鍵共有

Castryck ら [20] により提案された CSIDH 鍵共有を記述する．CSIDH 鍵共有は, 有限アーベル群 G による一方向性群作用をもつ空間 X 上で構成される．ここで, $X = \mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ は, $\mathbb{F}_p$ 上定義されて $\mathbb{F}_p$-有理自己準同型環が固定された虚 2 次整環 $\mathfrak{O}$ と同型であり，かつその同型により p 乗フロベニウス写像が $\pi \in \mathfrak{O}$ に移されるような超特異楕円曲線の $\mathbb{F}_p$-同型類の集合であり, $G = \mathrm{cl}(\mathfrak{O})$ は $\mathfrak{O}$ のイデアル類群である．以下では，$\mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ が空でないと仮定する．Castryck ら [20] は, 5.1.3.2 節で定義した CSIDH-REGA-CDH の判定版問題の困難性に基づいて, CSIDH 鍵共有方式を提案した.

K を虚 2 次代数体, $\mathfrak{O} \subset K$ をその整環とする, すなわちランク 2 の自由 $\mathbb{Z}$-加群である K の部分環である．$\mathfrak{O}$ の分数イデアルは, $\alpha \in K^*$ と $\mathfrak{O}$-イデアル $\mathfrak{a}$ によって $\alpha\mathfrak{a}$ と表される K 内の $\mathfrak{O}$-部分加群である．$\mathfrak{a}\mathfrak{b} = \mathfrak{O}$ となる $\mathfrak{O}$-分数イデアル $\mathfrak{b}$ が存在する時に ($\mathfrak{O}$-分数イデアル) $\mathfrak{a}$ は可逆であるという．そして, そのような $\mathfrak{b}$ が存在するならば, $\mathfrak{a}^{-1} = \mathfrak{b}$ と定義する．可逆分数イデアルの集合 $I(\mathfrak{O})$ はイデアル積に関してアーベル群をなす．この群には主イデアルからなる部分群 $P(\mathfrak{O})$ が含まれており, $\mathfrak{O}$ のイデアル類群は商群 $\mathrm{cl}(\mathfrak{O}) = I(\mathfrak{O})/P(\mathfrak{O})$ によって定義される．どのイデアル類 $[\mathfrak{a}] \in \mathrm{cl}(\mathfrak{O})$ にも整イデアルが存在してその代表として使うことができる．$\mathfrak{O}$ のどの整イデアル $\mathfrak{a}$ も $\mathfrak{O}$-イデアルの積として $\mathfrak{a}_s \not\subseteq \pi\mathfrak{O}$ となる整イデアル $\mathfrak{a}_s$ によって $(\pi\mathfrak{O})^r\mathfrak{a}_s$ と表せる．ここで, π は, p 乗フロベニウス写像．この表示により, 整イデアル $\mathfrak{a}$ に対して楕円曲線 $E/E[\mathfrak{a}]$ とそこへの $N(\mathfrak{a})$ 次同種写像 $\varphi_{\mathfrak{a}} : E \to E/E[\mathfrak{a}]$ が以下のように定義される．ここで, $N(\mathfrak{a}) := \#(\mathfrak{O}/\mathfrak{a})$ は $\mathfrak{a}$ のノルムである．$\varphi_{\mathfrak{a}}$ の分離的な部分は $E[\mathfrak{a}] = \bigcap_{\alpha \in \mathfrak{a}_s} \ker \alpha$ を核にもつ同種写像であり, 純非分離的な部分はフロベニウス写像 π の r 回の繰り返しで与えられる．同種写像 $\varphi_{\mathfrak{a}}$ 及び値域曲線 $E/E[\mathfrak{a}]$ は共に $\mathbb{F}_p$ 上定義されており $\mathbb{F}_p$-同型を除いて一意的に決まる．ここで主イデアルにより定義される同種写像は E 上の自己準同型写像になるので, 2 つのイデアルが同じイデアル類に属することと, 対応する同種写像が $\mathbb{F}_p$-同型な値域曲線を与えることとは同値である．つまり，$E/E[\mathfrak{a}]$ の $\mathbb{F}_p$-同型類はイデアル類 $[\mathfrak{a}]$ のみにより決まり，特に，この対応はイデアル類群 $\mathrm{cl}(\mathfrak{O})$ の $\mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ への作用を与える．更に, $\mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ に属する 2 つの楕円曲線間の $\mathbb{F}_p$-同種写像 ψ はすべて

この対応により可逆な $\mathfrak{O}$-イデアルから得られる．そして分離部分 $\mathfrak{a}_s$ は ψ から $\mathfrak{a}_s = \{\alpha \in \mathfrak{O} \mid \ker\alpha \supseteq \ker\psi\}$ によって復元できる．その対応は以下の定理にまとめられる．

定理 5.10 ([88, 80, 20]) 虚 2 次代数体内の整環 $\mathfrak{O}$ と $\pi \in \mathfrak{O}$ を $\mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ が空集合でないものとする．その時，以下で与えられるイデアル類群 $\mathrm{cl}(\mathfrak{O})$ の $\mathcal{E}\ell\ell_p(\mathfrak{O}, \pi)$ への作用は自由かつ推移的である．

$$\begin{array}{rcl} \mathrm{cl}(\mathfrak{O}) \times \mathcal{E}\ell\ell_p(\mathfrak{O}, \pi) & \to & \mathcal{E}\ell\ell_p(\mathfrak{O}, \pi) \\ ([\mathfrak{a}], E) & \mapsto & E/E[\mathfrak{a}], \end{array}$$

ここで，$\mathfrak{a}$ は類 $[\mathfrak{a}]$ を代表する整イデアルである．

以下では $E/E[\mathfrak{a}]$ を $[\mathfrak{a}]E$ と書くことにする．定理 5.10 で述べた群作用に基づいて，以下のように CSIDH 鍵共有プロトコル (図 5.1) を定義する．下の図で，$\mathfrak{a} \leftarrow \mathrm{cl}(\mathfrak{O})$ と書いたのは，実際にはイデアル類群 $\mathrm{cl}(\mathfrak{O})$ からのサンプリングとして，定義 5.5 の CSIDH-REGA-DL 問題及び定義 5.6 の CSIDH-REGA-CDH 問題に記載された REGA としての $\mathfrak{a}$ のサンプリング法を用いる．モンゴメリ型楕円曲線 $E : y^2 = x^3 + ax^2 + x$ に対して，係数 a は，E のモンゴメリ係数と呼ばれる．CSIDH 鍵共有では，始点曲線 $E : y^2 = x^3 + x$ に対して，アリスとボブによって計算される楕円曲線はすべてモンゴメリ型楕円曲線である．

アリス		**ボブ**
$\mathfrak{a} \leftarrow \mathrm{cl}(\mathfrak{O})$: アリスの秘密鍵,		$\mathfrak{b} \leftarrow \mathrm{cl}(\mathfrak{O})$: ボブの秘密鍵,
$[\mathfrak{a}]E$ (のモンゴメリ係数) を計算,	$\xrightarrow{[\mathfrak{a}]E}$	$[\mathfrak{b}]E$ (のモンゴメリ係数) を計算,
$SK_{\mathtt{Alice}} := [\mathfrak{a}]([\mathfrak{b}]E)$ のモンゴメリ係数.	$\xleftarrow{[\mathfrak{b}]E}$	$SK_{\mathtt{Bob}} := [\mathfrak{b}]([\mathfrak{a}]E)$ のモンゴメリ係数.

図 5.1: CSIDH 鍵共有の概要

イデアル類群 $\mathrm{cl}(\mathfrak{O})$ は可換なので，$[\mathfrak{a}]([\mathfrak{b}]E) = [\mathfrak{a}\mathfrak{b}]E = [\mathfrak{b}\mathfrak{a}]E = [\mathfrak{b}]([\mathfrak{a}]E)$ であり，そのモンゴメリ係数を考えれば $SK_{\mathtt{Alice}} = SK_{\mathtt{Bob}}$ となるので，アリスとボブは同じ鍵を共有できる．その計算アルゴリズムについては，Castryck ら [20], Meyer ら [67], 小貫ら [69] などを参照のこと．CSIDH 鍵共有の安全性は，CSIDH-REGA-CDH 問題の困難性に基づく．

最近，主に CSIDH 系の方式で使う演算を効率化するべき根同種写像（radical isogeny）計算法 [15, 13] や square-root Vélu 計算法 [3] が提案されている．また，Chávez-Saab らによって，効率化された SQALE 鍵共有方式 [10] も提案されており，4096 bits 以上の CSIDH 素数パラメータに関する SQALE 鍵共有の実装報告もなされている．

また，2019 年以降に，実用化を目指して，CSIDH 鍵共有方式を CSIDH 認証鍵共有に変換する研究が [46, 54, 58, 51, 30] などにより進展している．

5.2.1.2 CSIDH 鍵共有の変種

CSIDH 鍵共有の変種には，OSIDH 鍵共有 [18, 28] や SiGamal 暗号化方式 [66]・Sims 暗号化方式 [44] がある．OSIDH 鍵共有では，楕円曲線以外に「向き」と呼ばれる付加情報への群作用も考慮しており，SiGamal 暗号化方式では，楕円曲線とその上のねじれ点への群作用を考慮した暗号化方式の設計になっている．特に，SiGamal 暗号化では，ねじれ点への群作用を取り入れることで，CSIDH 共有鍵であるモンゴメリ係数（図 5.1 参照）ではなく，ねじれ点の離散対数からなる一様ランダムな乱数を送受信者間で共有できるようになっている．

5.2.2 SIDH 型の鍵共有

5.2.2.1 M-SIDH 鍵共有と MD-SIDH 鍵共有

5.1.2 節で見たように，SIDH 鍵共有に対して多項式時間の攻撃法が発見されたが，その直後に，それら攻撃法を回避する SIDH 鍵共有の変種方式が 2 つ提案された．1 つ目は，守谷による「同種写像次数」を隠すことによる次数隠蔽型（masked-degree）SIDH [65] であり，2 つ目は，Fouotsa によるねじれ点隠蔽型（masked torsion point images）SIDH [43] である．2023 年 1 月に，2 論文 [65, 43] を統合して安全性に関して新たな考察が加えられた [42] が発表された．[42] では，ねじれ点隠蔽型 SIDH と次数隠蔽型 SIDH をそれぞれ M-SIDH 鍵共有方式，MD-SIDH 鍵共有方式と呼んでいる．

[42] において，M-SIDH 方式の方が，MD-SIDH 方式より小さい素数 p によって同等の安全性が得られることが述べられているが，これらの方式は，SIDH 方式より効率面では格段に劣ることも指摘されている．例えば，NIST レベル 1,3,5 に対応する M-SIDH 素数 p の bit 長は，それぞれ 5911, 9382, 13000 bits となることが示されており，SIKE 暗号化に提案された素数 bit 長と比べて格段に大きく，それにより計算効率も劣ることになる．また，その安全性検証はこれからの研究課題であり，今後も，研究の進展を注視する必要がある方式である．

5.2.2.2 pSIDH 鍵共有

「部分整環（suborder）表現」と呼ぶ新しい同種写像表現を利用した鍵共有法の提案が Leroux によってなされた [61]．その鍵共有法は pSIDH（prime SIDH）鍵共有と呼ばれるが，これまで暗号構成に取り入れてこなかった大きい素数次数の同種写像を部分整環表現を使うことで暗号構成に取り込んでいる．その安全性は，同種写像の部分整環表現からイデアル表現を求める問題（SOIP: SubOrder to Ideal Problem）の困難性に基づいている．[61] によれば，準指数関数時間の量子攻撃を受けるとされており，それが現在最速の pSIDH 鍵共有への攻撃法となっている．

5.2.3 CSIDH ベース署名方式

5.2.3.1 SeaSign 署名

De Feo と Galbraith [37] により, CSIDH ベースの SeaSign 署名が提案された．これは, CSIDH 鍵共有の数学的な構造を利用したものであり, まだ実用的とは言い難いが, 現実的な計算時間に収まる署名方式となっている．以下では, [37] で, 「基本形」と呼ばれる SeaSign 署名方式を記載する．[37] では, 更に, 基本形でも使われたパラメータ t と共に, パラメータ s を導入して, 署名が短い方式や公開鍵が短い方式といった変形方式を定義している．

後続研究 [32] において, 実用化を目指して SeaSign 署名の高速化が図られている．SeaSign 署名を改良することで, 実用的な速度性能を達成できるかどうかを見極めることは, 今後の同種写像ベース署名研究にとって重要な課題の一つである．以下では, ベクトル $\mathbf{e}$ の各成分は e_i $(i = 1, 2, \ldots, n)$ とする, 即ち, $\mathbf{e} = (e_1, e_2, \ldots, e_n)$ である．ベクトル $\mathbf{f}_k, \mathbf{z}_k$ についても同様の記法を用いる．また，整数 a, b $(a < b)$ に対して $[a, b] := \{a, a+1, \ldots, b-1, b\} \subset \mathbb{Z}$ とする．

鍵生成： 公開パラメータ $pp_{\mathtt{csidh}} := (\mathfrak{O}, (\mathfrak{l}_1, \mathfrak{l}_2, \ldots, \mathfrak{l}_n), E, B)$, すなわち, 虚 2 次整環 $\mathfrak{O}$, イデアル $\mathfrak{l}_1, \mathfrak{l}_2, \ldots, \mathfrak{l}_n$, $\mathbb{F}_p$-有理な超特異楕円曲線 E, 上限値 B を入力とする．係数ベクトル $\mathbf{e} \leftarrow_R [-B, B]^n$ を生成する．$E_{\mathtt{A}} := \left[\prod_{i=1}^{n} \mathfrak{l}_i^{e_i}\right] E$ を計算して, 秘密鍵 $sk := \mathbf{e}$, 公開鍵 $pk := E_{\mathtt{A}}$ とする．

署名生成： 公開パラメータ $pp_{\mathtt{csidh}}$, メッセージ $\mathtt{msg}$, 公開鍵 $pk := E_{\mathtt{A}}$, 秘密鍵 $sk := \mathbf{e}$ を入力とする．各 $k = 1, 2, \ldots, t$ に対して, 係数ベクトル $\mathbf{f}_k \leftarrow_R [-(nt+1)B, (nt+1)B]^n$ を生成して, $\mathcal{E}_k := \left[\prod_{i=1}^{n} \mathfrak{l}_i^{f_{k,i}}\right] E$ を計算す

る．ハッシュ値を $b_1 \| \cdots \| b_t := H(j(\mathcal{E}_1), \ldots, j(\mathcal{E}_t), \mathtt{msg})$ と bit 分解する．各 $k = 1, 2, \ldots, t$ に対して，もし $b_k = 0$ であれば，$\mathbf{z}_k := \mathbf{f}_k$ として，もし $b_k = 1$ であれば，$\mathbf{z}_k := \mathbf{f}_k - \mathbf{e}$ として，もし $\mathbf{z}_k \notin [-ntB, ntB]^n$ であれば $\perp$ を出力する．そうでなければ，署名 $\sigma := (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t, b_1, b_2, \ldots, b_t)$ を出力する．

署名検証　公開パラメータ $pp_{\mathtt{csidh}}$，メッセージ $\mathtt{msg}$，公開鍵 $pk := E_{\mathtt{A}}$，署名 σ を入力とする．まず，署名 σ が，$\sigma := (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t, b_1, b_2, \ldots, b_t)$ というデータになっていることを確認する．各 $k = 1, 2, \ldots, t$ に対して，もし $b_k = 0$ であれば，$\mathcal{E}_k := \left[\prod_{i=1}^{n} \mathfrak{l}_i^{z_{k,i}}\right] E$ を計算して，もし $b_k = 1$ であれば，$\mathcal{E}_k := \left[\prod_{i=1}^{n} \mathfrak{l}_i^{z_{k,i}}\right] E_{\mathtt{A}}$ を計算する．ハッシュ値を $b'_1 \| \cdots \| b'_t := H(j(\mathcal{E}_1), \ldots, j(\mathcal{E}_t), \mathtt{msg})$ と bit 分解する．もし，$(b'_1, b'_2, \ldots, b'_t) = (b_1, b_2, \ldots, b_t)$ であれば，受理を出力して，そうでなければ，不受理を出力する．

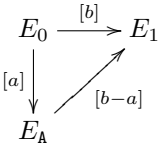
SeaSign 署名方式は，ランダムオラクルモデルにおいて，CSIDH-REGA-CDH 問題困難性の仮定の下で，選択文書攻撃に対して存在的偽造不可 (EUF-CMA) 安全であることが示されている [37].

5.2.3.2　CSI-FiSh 署名

Beullens–Kleinjung–Vercauteren [6] は，CSIDH-512 パラメータに関するイデアル類群 $\mathrm{cl}(\mathfrak{O})$ の構造計算を遂行することで，効率を高めた CSIDH ベースの CSI-FiSh 署名方式を提案した．CSIDH-512 パラメータでは，512 bits 素数 $p = 4 \cdot \ell_1 \cdot \cdots \cdot \ell_{74} - 1$, $\ell_{74} = 587$ を用いており，Beullens ら [6] は，$\mathbb{Q}(\sqrt{-p})$ のイデアル類群 $\mathrm{cl}(\mathfrak{O})$ が，ノルム 3 のイデアル $\mathfrak{l}_1$ により生成される位数 $N := \#\mathrm{cl}(\mathfrak{O}) = 37 \cdot 1407181 \cdot 51593604295295867744293584889 \cdot 3159941450468199585300827874558783220490 9$ の有限巡回群になることを示した．つまり，$[a] \in \mathbb{Z}/N\mathbb{Z}$ に対してイデアル類 $[\mathfrak{l}_1]^a = [\mathfrak{l}_1^a]$ を対応させることで同型 $\mathbb{Z}/N\mathbb{Z} \cong \mathrm{cl}(\mathfrak{O})$ が得られる．更に，イデアル類群の作用 $[\mathfrak{l}_1^a]E$ を同型 $\mathbb{Z}/N\mathbb{Z} \cong \mathrm{cl}(\mathfrak{O})$ を使って $[a]E$ と表し，$\mathbb{Z}/N\mathbb{Z}$ の作用と見なすことにする．これにより 5.1.3.1 節に述べた，より望ましい CSIDH ベースの一方向性群作用 EGA が得られた．

CSI-FiSh 署名の構成法は SeaSign 署名と変わらないので，Σ-プロトコルにおける記述の差異のみを右図に従って以下に述べる：位数 N の巡回群 $\mathbb{Z}/N\mathbb{Z}\,(\cong \mathrm{cl}(\mathfrak{O}))$ の作用に基づき，秘密鍵（証拠）を乱数 $a \in \mathbb{Z}/N\mathbb{Z}$，公開鍵 $E_0, E_{\mathtt{A}} := [a]E_0$ とする．証明者は乱数 $b \in \mathbb{Z}/N\mathbb{Z}$ によりコミットメント $E_1 := [b]E_0$ を計算して検証者に送る．検証者はチャレンジ $c \in \{0, 1\}$ をランダムに選び証明者に送付，証明者はレスポンス $r := b - ca \bmod N$ を検証者に送る．最後に，検証者は $c = 0$ であれば $E_1 = [r]E_0$ であること，$c = 1$ であれば $E_1 = [r]E_{\mathtt{A}}$ であることが成り立つかどうか検証して検証結果を出力する．

CSI-FiSh 同種写像図式



実際に，CSI-FiSh 署名にするには，Fiat–Shamir 変換に則り，チャレンジを $(c_i \in \{0, 1\})_{i \in [t]}$ と t 個にするとともに，それらをハッシュ関数を用いて計算して非対話化することで得られる．

上に示したフレームワークをより広範囲の CSIDH パラメータに拡張できれば望ましいが，5.1.3.2 節に述べたように，CSIDH 問題の最近の安全性検討状況によると，NIST 安全性レベル 1 を満たすためには，素数 p を 4096 bits または 5120 bits 程度に大きくする必要が示されている．そして，それに対応するイデアル類群計算を遂行するのは現状では困難と思われており，このことが，実適用における CSI-FiSh 署名アプローチの限界を示している．

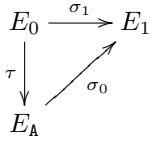
CSI-FiSh 署名提案後に，El Kaafarani ら [35] により，タイト安全な Lossy CSI-FiSh 署名の提案もなされた．そして，リング署名・グループ署名など高機能暗号系への拡張研究 [5, 2] があるのが CSI-FiSh 署名の利点の一つとなっている．

5.2.4 GPS 署名

Galbraith–Petit–Silva (GPS) [49] によって始めて自己準同型環の知識証明に基づく署名方式が提案された．GPS 署名は実際に利用するのは困難であろうと思われているが，現在，GPS 署名は，SQISign 署名の原型を与えているという点で重要である．5.1.4 節で述べた Deuring 対応と KLPT アルゴリズム [55] が GPS 署名の理論的基礎を与える．

右図において E_0 は 5.1.4.2 節で与えた $j(E_0) = 1728$ なる楕円曲線（特殊極値的楕円曲線）であり，そこで見たようにその E_0 に関しては $\mathrm{End}(E_0)$ の構造が簡明な形で与えられている．その楕円曲線 E_0 からの秘密鍵同種写像 $\tau : E_0 \to E_{\mathtt{A}}$ を知っている証明者（署名生成者）は，$E_{\mathtt{A}}$ から別の楕円曲線 E_1 への同種写像 $\sigma_0 : E_{\mathtt{A}} \to E_1$ と τ との合成 $\sigma_0 \circ \tau : E_0 \to E_1$ を KLPT アルゴリズムによって「ランダム化」して同じ始点 E_0 と終点 E_1 をもつ $\sigma_0 \circ \tau$ とは異なる同種写像 $\sigma_1 : E_0 \to E_1$ を得ることができる．

GPS 同種写像図式



さらに，自己準同型環 $\mathrm{End}(E_{\mathtt{A}})$ を計算する問題の困難性に基づけば，このようなランダム化ができるのは，τ を知っている証明者に限られるので，チャレンジ bit $c \in \{0,1\}$ を送って証明者に同種写像 σ_c を答えさせることにより，τ に関する知識の有無を検査することができて，認証・署名方式が構成できる．それが GPS 認証方式，そしてその Fiat–Shamir 変換署名が GPS 署名である．ここでは [49, 第 4 章] と [62, 5.1.2 節] に基づいて GPS 署名を記述する．また，[49, 第 4 章] では，通常の Fiat–Shamir 変換を施した署名方式と Unruh 変換を施した署名方式の 2 方式が記述されているが，ここでは記述の簡便さを考慮して前者の記述を基にして以下に署名方式を与える．

鍵生成： 既知の特殊極値的自己準同型環 $\mathcal{O}_0$ をもつ超特異楕円曲線 E_0，互いに素な B-べき平滑数 S_1, S_2[*6]を，S_1, S_2 次の同種写像グラフ上ランダムウォークが急攪拌性定理により一様分布を導く程度に十分大きくとる．セキュリティパラメータ λ に対して $t := \lambda$（または $t := 2\lambda$）として，t bits 出力のハッシュ関数 H を選ぶ．$pp_{\mathtt{gps}} := (E_0, S_1, S_2, H)$ を公開パラメータとする．さらに，E_0 を始点とする S_1 次のランダムな同種写像 $\tau : E_0 \to E_{\mathtt{A}}$ を計算して，$pp_{\mathtt{gps}}$ と $E_{\mathtt{A}}$ を公開鍵として，τ を秘密鍵とする．

署名生成： 各 $i = 1, \ldots, t$ に関して $E_{\mathtt{A}}$ を始点とする S_2 次のランダムな同種写像 $\sigma_{0,i} : E_{\mathtt{A}} \to E_{1,i}$ を計算する．署名対象メッセージ $\mathtt{msg}$ に対してチャレンジ bit 列 $h := b_1 \| \cdots \| b_t := H(j(E_{1,1}), \ldots, j(E_{1,t}), \mathtt{msg}) \in \{0,1\}^t$ をハッシュ関数 H で計算する．各 $i = 1, \ldots, t$ に対して，もし $b_i = 1$ なら KLPT アルゴリズムによって「ランダム化」したランダム同種写像 $\sigma_{1,i} : E_0 \to E_{1,i}$ を計算する．署名を $\sigma := (h, \sigma_{b_1,1}, \ldots, \sigma_{b_t,t})$ とする．

署名検証： 公開鍵 $(pp_{\mathtt{gps}}, E_{\mathtt{A}})$，メッセージ $\mathtt{msg}$ と署名 $\sigma = (h, \sigma_1, \ldots, \sigma_t)$ を入力として，各 $i = 1, \ldots, t$ に対して，同種写像 σ_i を計算して，その終点曲線 $E_{1,i}$ を得る．次に $H(j(E_{1,1}), \ldots, j(E_{1,t}), \mathtt{msg})$ を計算して署名内の h と一致するかどうか検証して，全ての $i = 1, \ldots, t$ に対して検証が成功すれば受理を出力して，そうでなければ，不受理を出力する．

GPS 署名は，超特異楕円曲線同種写像計算問題またはそれと同値な自己準同型環計算問題（定義 5.7）の困難性を仮定すればランダムオラクルモデルの下で EUF-CMA 安全であることが示されている [49, 定理 10]．GPS 署名では，1 bit のチャレンジを用いた Σ-プロトコルに基づいているため，署名サイズが大きくなるのが欠点である．また，署名生成で使われた KLPT アルゴリズムの計算時間改善も課題であった [62, 5.1.2 節]．以上，GPS 署名には (1) 署名サイズ 及び (2) KLPT アルゴリズム計算時間 に関する 2 つの課題が存在する．

[*6] S_k（$k = 1, 2$）が B-べき平滑数（powersmooth number）とは，S_k が $\ell_{k,i}^{e_{k,i}} < B$ なる $\ell_{k,i}^{e_{k,i}}$ の積で表される（i.e., $S_k = \prod_i \ell_{k,i}^{e_{k,i}}$）ことである．

5.3 同種写像に基づく主要な暗号方式

本節では，公開鍵と署名サイズが小さいことを特長にもつ SQISign 署名について述べる（表 5.3 参照）.

表 5.3: 同種写像に基づく暗号の分類

文献	暗号化	鍵交換	署名
SQISign [39]			○

5.3.1 SQISign 署名

以下，自己準同型環計算問題（定義 5.7）の困難性に安全性の根拠を置く SQISign 署名を概説する．SQISign 署名は公開鍵と署名を合わせたサイズが小さい方式として注目されている．5.2.4 節で述べた GPS 署名を基にして改良を加えた署名方式が SQISign 署名であり，ASIACRYPT 2020 で De Feo–Kohel–Leroux–Petit–Wesolowski [39] により提案された．5.2.4 節末尾に付した GPS 署名の 2 つの課題を克服している．チャレンジ空間に同種写像の空間を用いることで，そのサイズをセキュリティパラメータ λ まで大きくして，Σ-プロトコルを 1 度適用するだけで十分な Fiat–Shamir 署名構成とした．これで署名サイズが格段に小さくなった．また，GPS 署名生成においては，表 5.2 の Deuring 対応に基づいて，同種写像のイデアル表現（表 5.2 の四元数環側）をねじれ点を使った表現（表 5.2 の楕円曲線側）に変換する部分で時間が費やされていたが，SQISign 署名ではその処理を速度改善したサブルーチン（IdealToIsogeny）に置き換えるのに成功して演算効率も大きく改善した [39, 41].

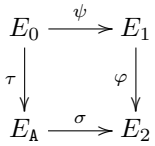
また，安全性に関しては，健全性は超特異平滑自己準同型写像計算問題（定義 5.8）の困難性に基づき，零知識性は定義 5.9 で述べた SQISign 署名のランダム識別問題の困難性に基づいている．初期提案 [39] では，ノルム方程式を解くサブルーチンに不備があり，生成される署名同種写像 σ に偏りが生じていたことが [41] において指摘された．そして，更に [41] でその不備を除去したアルゴリズム提案が行われた.

具体的なパラメータ，特に適切な SQISign 素数（SQISign-friendly prime）p を生成する問題は非自明であり，初期提案 [39] から始まり，[23, 41, 9] と現在も進行中の研究テーマである．その現状報告を後ほど行う．また，小貫 [70] により，SQISign 鍵生成で得られる鍵の分布の理論・実験による解析がなされており，[39] で述べられた従来仕様の不備を指摘して，改善アルゴリズムを提案している.

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

■SQISign 署名方式記述　SQISign 署名では，右図の同種写像 τ が秘密鍵で，超特異楕円曲線 $E_{\mathtt{A}}$ が公開鍵 (の主要な一部) である．署名生成では，コミットメント同種写像 ψ とチャレンジ同種写像 φ を適切に生成して得られた合成写像 $\varphi \circ \psi \circ \hat{\tau}$ を一般化された KLPT アルゴリズムでランダム化した同種写像 σ を署名（Σ-プロトコルのレスポンス）とする．チャレンジ φ によりセキュリティパラメータ分のランダムネスを与えることができるので，1 度の Σ-プロトコル適用で十分な安全性が達成できる．よって，GPS 署名と比べて格段に短い署名サイズが実現できる.

SQISign 同種写像図式



鍵生成：　既知の特殊極値的自己準同型環 $\mathcal{O}_0$ をもつ超特異楕円曲線 E_0，λ bits の平滑奇数 D_c（λ はセキュリティパラメータ），超特異 2-同種写像グラフの直径より大きな e による $D := 2^e$ を生成して，$pp_{\mathtt{sqisign}} := (E_0, D_c, D)$

を公開パラメータとする．さらに，E_0 を始点とするランダムな同種写像 $\tau : E_0 \to E_{\mathtt{A}}$ を計算して，$pp_{\mathtt{sqisign}}$ と $E_{\mathtt{A}}$ を公開鍵として，τ を秘密鍵とする．

署名生成： E_0 を始点とするランダムな同種写像 $\psi : E_0 \to E_1$ を計算．署名対象メッセージ $\mathtt{msg}$ に対してハッシュ関数 H で計算した $H(j(E_1), \mathtt{msg})$ から決まる D_c 次の巡回同種写像 $\varphi : E_1 \to E_2$ を計算．同種写像の合成 $\varphi \circ \psi \circ \hat{\tau} : E_{\mathtt{A}} \to E_2$ から（一般化された KLPT アルゴリズムを用いて）同じ始点・終点を有して $\hat{\varphi} \circ \sigma$ が巡回同種写像になる D 次のランダム同種写像 $\sigma : E_{\mathtt{A}} \to E_2$ を計算．(E_1, E_2, σ) を $\mathtt{msg}$ の署名として出力．

署名検証： 公開鍵 $(pp_{\mathtt{sqisign}}, E_{\mathtt{A}})$，メッセージ $\mathtt{msg}$ と署名 (E_1, E_2, σ) を入力として，E_1 から E_2 への同種写像 $\varphi := H(j(E_1), \mathtt{msg})$ を計算する．σ が $E_{\mathtt{A}}$ から E_2 への D 次同種写像であることと $\hat{\varphi} \circ \sigma$ が $E_{\mathtt{A}}$ から E_1 への巡回同種写像であることを検証して，共に成立すれば受理を出力して, そうでなければ, 不受理を出力する．

既に述べたように，SQISign 署名の安全性は，超特異平滑自己準同型写像計算問題（定義 5.8）の困難性と，定義 5.9 で述べた SQISign 署名 σ のランダム識別問題の困難性に基づいている．

■SQISign 署名パラメータ 署名同種写像 σ の次数は $D = 2^e$，チャレンジ同種写像 φ の次数は平滑奇数 D_c であるので，それら同種写像を小さい拡大次数の有限体で効率的に計算するために，超特異楕円曲線の位数 $\#E(\mathbb{F}_{p^2}) = p^2 - 1$ を考慮して，できるだけ大きい正整数 f，正奇数 T に関して $2^f \cdot T \,|\, p^2 - 1$ が満たされる素数 p（SQISign 素数）を生成することが必要である．具体的には，ある B に対して B-平滑な T，$T \approx p^{5/4+\epsilon}$ ([9] では例えば $0.02 < \epsilon < 0.1$ とする) に対して $2^f \cdot T \,|\, p^2 - 1$ となる素数 p を探索する必要がある．SQISign 素数の選択基準として，署名検証の効率化には f をできるだけ大きくして，署名生成の効率性にとっては $\sqrt{B}/f$ をできるだけ小さくするのが望ましい [41].

NIST 安全性レベル 1： SQISign 素数は，NIST 安全性レベル 1 については，[41] において，XGCD アルゴリズムに基づいて，$B = 3923$ に関して B-平滑な T を持つ以下の 254 bits 素数 p が生成された．$f = 66$ であり，T は式 (5.1) で灰色でない B 以下の奇素因数の積で与えられる．T に関しては他の SQISign 素数についても同様である．また，[39, 62] によればレベル 1 パラメータでは署名 σ の次数 $D = 2^e$ の指数 e が $e = 1000$ で与えられている．

$$\begin{aligned} p + 1 &= 2^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 19 \cdot 29^2 \cdot 37^2 \cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461 \cdot 521 \cdot 3923 \cdot 62731 \cdot 96362257 \cdot 3924006112952623, \\ p - 1 &= 2 \cdot 3^{65} \cdot 13 \cdot 17 \cdot 43 \cdot 79 \cdot 157 \cdot 239 \cdot 271 \cdot 283 \cdot 307 \cdot 563 \cdot 599 \cdot 607 \cdot 619 \cdot 743 \cdot 827 \cdot 941 \cdot 2357 \cdot 10069. \end{aligned} \tag{5.1}$$

同じく NIST 安全性レベル 1 で最良の SQISign 素数として，[9] において，$B = 523$ に関して B-平滑な T を持つ以下の 254 bits 素数 $p = 2r^3 - 1$ ($r = 204614491255000374748856320$) が生成されている．$f = 47$ である．

$$\begin{aligned} p + 1 &= 2^{46} \cdot 5^3 \cdot 13^3 \cdot 31^3 \cdot 73^3 \cdot 83^3 \cdot 103^3 \cdot 107^3 \cdot 137^3 \cdot 239^3 \cdot 271^3 \cdot 523^3, \\ p - 1 &= 2 \cdot 3^3 \cdot 7 \cdot 11^2 \cdot 17^2 \cdot 19 \cdot 101 \cdot 127 \cdot 149 \cdot 157 \cdot 167 \cdot 173 \cdot 199 \cdot 229 \cdot 337 \cdot 457 \cdot 479 \cdot \\ &\quad 141067 \cdot 3428098456843 \cdot 484047594531861479165862 1. \end{aligned} \tag{5.2}$$

NIST 安全性レベル 3：NIST 安全性レベル 3,5 に関しては [41] までは適切なパラメータ例が与えられていなかったが，レベル 3 について，後続研究である [9] において最良の SQISign 素数として，$B = 10243$ に関して B-平滑な T を持つ以下の 382 bits 素数 $p = 2r^6 - 1$ ($r = 11896643388662145024$) が生成されている．$f = 80$ である．

$$\begin{aligned} p + 1 &= 2^{79} \cdot 3^6 \cdot 23^{12} \cdot 107^6 \cdot 127^6 \cdot 307^6 \cdot 401^6 \cdot 547^6, \\ p - 1 &= 2 \cdot 5^2 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 47 \cdot 71 \cdot 79 \cdot 109 \cdot 149 \cdot 229 \cdot 269 \cdot 283 \cdot 349 \cdot 449 \cdot 463 \cdot 1019 \cdot 1033 \cdot 1657 \cdot 2179 \cdot \\ &\quad 2293 \cdot 4099 \cdot 5119 \cdot 10243 \cdot 381343 \cdot 19115518067 \cdot 740881808972441233 \cdot 832321437914821351639 21. \end{aligned} \tag{5.3}$$

NIST 安全性レベル 5：また，レベル 5 についても，[9] において最良の SQISign 素数として，$B = 150151$ に関して B-平滑な T を持つ以下の 508 bits 素数 $p = 2r^6 - 1$ ($r = 2669797390044648368060825 6$) が生成されている．$f = 86$

である．

$$p+1 = 2^{85} \cdot 17^{12} \cdot 37^6 \cdot 59^6 \cdot 97^6 \cdot 233^6 \cdot 311^{12} \cdot 911^6 \cdot 1297^6, \tag{5.4}$$
$$p-1 = 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11^2 \cdot 23^2 \cdot 29 \cdot 127 \cdot 163 \cdot 173 \cdot 191 \cdot 193 \cdot 211 \cdot 277 \cdot 347 \cdot 617 \cdot 661 \cdot 761 \cdot 1039 \cdot 4637 \cdot 5821 \cdot 15649 \cdot 19139 \cdot 143443 \cdot 150151 \cdot 3813769 \cdot 358244059 \cdot 992456937347 \cdot 353240481781965369823897507 \cdot 860102006951457440137 1658891403021.$$

■SQISign 署名実装報告　NIST レベル 3,5 パラメータ (式 (5.3), (5.4)) に関しては，まだ実装報告が公開されていない．以下では，NIST レベル 1 パラメータ (式 (5.1), (5.2)) に関する実装報告をまとめる．
データサイズ：式 (5.1), (5.2) の NIST レベル 1 パラメータに関して，秘密鍵サイズ 16 Bytes，公開鍵サイズ 64 Bytes，署名サイズ 204 Bytes が報告されている [39, 41].
演算時間：式 (5.1) の NIST レベル 1 パラメータに関して，演算時間中央値は，鍵生成 218 ms（ミリ秒），署名生成 1081 ms，署名検証 19 ms と報告されている [41]．詳細は [41] を参照のこと．

5.4　同種写像に基づく暗号技術に関するまとめ

本章では, 同種写像に基づいた暗号技術をまとめた．特に，2 つの柱となる署名方式である SQISign 署名と CSIDH ベース署名（SeaSign 署名・CSI-FiSh 署名）に関して, 関連する GPS 署名や CSIDH 鍵共有なども含めた方式記述と安全性研究についてまとめてきた.

[26] によると, Couveignes は, 1997 年の École Normale Supérieure でのセミナーで既に同種写像に基づく暗号技術を提案しており, ほぼ同時期に Kohel [56] や Galbraith [47] も, 同種写像問題に関する研究を始めていた. つまり, 同種写像暗号技術の研究は既に 25 年の歴史をもつ. そして, 最近になり, 耐量子計算機暗号の必要性が高まることで, 同種写像暗号技術は注目されて研究が進み, NIST PQC 第 4 ラウンドコンペティションにも選ばれた SIKE 暗号化及びその基本形である SIDH 鍵共有は，最近まで堅調に安全性評価を積み重ねてきた．しかし，2022 年の Castryck–Decru の攻撃法 [11] を始めとする一連の攻撃法 [64, 78] は SIDH 鍵共有に対して決定的な結果をもたらした.

一方，それは，この 25 年の研究の一つの到達点として，同種写像暗号全体に対して基本的な安全性指標を提示することになった．現在，その新しい安全性指標に基づいて更に安全性解析が進展していると共に，また新しい方式提案も含む活発な研究活動が引き続いて行われている．例えば，最近新規提案された M-SIDH 鍵共有・MD-SIDH 鍵共有の安全性研究は，今後の重要な研究課題の一つである.

現在，特に，公開鍵と署名を合わせたサイズが短い SQISign 署名が注目されていると共に，5.1.3 節で見たような CSIDH ベースの一方向性群作用に関する研究も注目されており，種々の暗号プロトコルへの応用も視野に入れた研究も進んでいる．それらも含めて，今後, 特に注意すべきこと数点について以下にまとめておく.

- SQISign 署名は，公開鍵と署名のサイズの小ささ，補助点なしの署名構成，そして短署名に対する強い社会的ニーズなどを踏まえると，現在有望な同種写像暗号技術と思われる．その一方，零知識性に関する計算問題（定義 5.9）の安全性検討などに関して，まだ安全性評価が不十分であり，その安全性評価は今後の重要な課題の一つである．さらに，今後，標準化などを考慮するのであれば，実装研究を進める必要があり，特にさまざまなプラットフォームでの実装結果を蓄えていく必要もある.
- CSIDH ベース署名である SeaSign 署名と CSI-FiSh 署名についても，今後の耐量子計算機署名として研究が進められているが，5.2.3 節で述べたように，安全性評価が定まった実用的な署名を得るためには，まだ今後の安全性・実装研究が必要である．そして，リング署名・グループ署名などの高機能暗号系への応用研究も重要であ

り，今後の研究動向に注目する必要がある．

- （一般的な）超特異同種写像問題及びそれと同値な自己準同型環計算問題に関しては，これまで主に，SIKE 暗号化パラメータに関係して，具体的な安全性評価が行われてきた．SQISign 署名パラメータを具体的に決めていくためには，SQISign 署名パラメータに対して，これらの問題に対する古典・量子アルゴリズムの詳細な解析・見積もりを行うことが重要であり, 今後の課題である．
- 上で述べたように現在研究が進展している新しい安全性指標に基づいて，全体に，同種写像暗号技術は，まだまだ研究の余地があり，鍵・暗号文・署名サイズの小ささの点で他の耐量子暗号技術にない特長があるので，さまざまな利用用途を見据えて今後も継続的な研究が望まれる．

第 5 章の参照文献

[1] N. Alamati, L. D. Feo, H. Montgomery, S. Patranabis. Cryptographic group actions and applications. *ASIACRYPT 2020*, Part II volume 12492 of LNCS, pp. 411–439, Springer, 2020.

[2] W. Beullens, S. Dobson, S. Katsumata, Y. Lai, F. Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. *EUROCRYPT 2022*, Part II, volume 13276 of LNCS, pp. 95–126, Springer, 2022.

[3] D. J. Bernstein, L. D. Feo, A. Leroux, B. Smith. Faster computation of isogenies of large prime degree. *ANTS-XIV*, volume 4 of The Open Book Series, pp. 39–55, Mathematical Sciences Publishers, 2020.

[4] J. Biasse, D. Jao, A. Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. *INDOCRYPT 2014*, volume 8885 of LNCS, pp. 428–442, Springer, 2014.

[5] W. Beullens, S. Katsumata, F. Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. *ASIACRYPT 2020*, Part II, volume 13276 of LNCS, pp. 464–492, Springer, 2020.

[6] W. Beullens, T. Kleinjung, F. Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. *ASIACRYPT 2019*, Part I, volume 11921 of LNCS, pp. 227–247, Springer, 2019.

[7] D. J. Bernstein, T. Lange, C. Martindale, L. Panny. Quantum circuits for the CSIDH: optimizing quantum evaluation of isogenies. *EUROCRYPT 2019*, Part II, volume 11477 of LNCS, pp. 409–441, Springer, 2019.

[8] X. Bonnetain, A. Schrottenloher. Quantum security analysis of CSIDH. *EUROCRYPT 2020*, Part II, volume 12106 of LNCS, pp. 493–522, Springer, 2020.

[9] G. Bruno, M. C.-R. Santos, C. Costello, J. K. Eriksen, M. Meyer, M. Naehrig, B. Sterner. Cryptographic smooth neighbors. *IACR Cryptology ePrint Archive*, 2022/1439.

[10] J. Chávez-Saab, J. Chi-Domínguez, S. Jaques, F. Rodríguez-Henríquez. The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. *Journal of Cryptographic Engineering*, volume 12, number 3, pp. 349–368, Springer, 2022.

[11] W. Castryck, T. Decru. An efficient key recovery attack on SIDH (preliminary version). *IACR Cryptology ePrint Archive*, 2022/975.

[12] W. Castryck, T. Decru. Magma code of an efficient key recovery attack on SIDH, 2022. `https://homes.esat.kuleuven.be/~wcastryc/`. （2023-04-11 閲覧）

[13] W. Castryck, T. Decru, M. Houben, F. Vercauteren. Horizontal racewalking using radical isogenies. *ASIACRYPT 2022*, volume 13792 of LNCS, Springer, 2022.

[14] W. Castryck, T. Decru, B. Smith. Hash functions from superspecial genus-2 curves using Richelot isogenies. *Journal of Mathematical Cryptology*, volume 14, number 1, pp. 268–292, De Gruyter, 2020.

[15] W. Castryck, T. Decru, F. Vercauteren. Radical isogenies. *ASIACRYPT 2020*, Part II, volume 12492 of

LNCS, pp. 493–519, Springer, 2020.

[16] W. Castryck, M. Houben, F. Vercauteren, B. Wesolowski. On the decisional Diffie–Hellman problem for class group actions on oriented elliptic curves. *ANTS-XV*, volume 8 of Research in Number Theory, pp. 39–55, Springer, 2022.

[17] A. M. Childs, D. Jao, V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, volume 8, number 1, pp. 1–29, De Gruyter, 2014.

[18] L. Colò, D. Kohel. Orienting supersingular isogeny graphs. *Journal of Mathematical Cryptology*, volume 14, number 1, pp. 414–437, De Gruyter, 2020.

[19] D. Charles, K. Lauter, E. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, volume 22, number 1, pp. 93–113, Springer, 2009.

[20] W. Castryck, T. Lange, C. Martindale, L. Panny, J. Renes. CSIDH: An efficient post-quantum commutative group action. *ASIACRYPT 2018*, Part III, volume 11274 of LNCS, pp. 395–427, Springer, 2018.

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

[22] W. Castryck, N. V. Meeren. Two remarks on the vectorization problem. *INDOCRYPT 2022*, volume 13774 of LNCS, pp. 658–678, Springer, 2022.

[23] C. Costello, M. Meyer, M. Naehrig. Sieving for twin smooth integers with solutions to the Prouhet-Tarry-Escott problem. *EUROCRYPT 2021*, Part I, volume 12696 of LNCS, pp. 272–301, Springer, 2021.

[24] C. Costello. B-SIDH: Supersingular isogeny Diffie–Hellman using twisted torsion. *ASIACRYPT 2020, Part II*, volume 12492 of LNCS, pp. 440–463, Springer, 2020.

[25] C. Costello. The case for SIKE: A decade of the supersingular isogeny problem. *IACR Cryptology ePrint Archive*, 2021/543.

[26] J. Couveignes. Hard homogeneous spaces. *IACR Cryptology ePrint Archive*, 2006/291.

[27] W. Castryck, J. Sotáková, F. Vercauteren. Breaking the decisional Diffie–Hellman problem for class group actions using genus theory: Extended version. *Journal of Cryptology*, volume 35, issue 4, Springer, 2022.

[28] P. Dartois, L. D. Feo. On the security of OSIDH. *PKC 2022*, Part I, volume 13177 of LNCS, pp. 52–81, Springer, 2022.

[29] C. Delfs, S. Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, volume 78, number 2, pp. 425–440, Springer, 2016.

[30] J. Duman, D. Hartmann, E. Kiltz, S. Kunzweiler, J. Lehmann, D. Riepel. Group action key encapsulation and non-interactive key exchange in the QROM. *ASIACRYPT 2022*, Part II, volume 13792 of LNCS, pp. 36-66, Springer, 2022.

[31] L. De Feo, D. Jao, J. Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, volume 8, number 3, pp. 209–247, De Gruyter, 2014.

[32] T. Decru, L. Panny, F. Vercauteren. Faster SeaSign signatures through improved rejection sampling. *PQCrypto 2019*, volume 11505 of LNCS, pp. 271–285, Springer, 2019.

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

[34] K. Eisenträger, S. Hallgren, C. Leonardi, T. Morrison, J. Park. Computing endomorphism rings of supersin-

gular elliptic curves and connections to pathfinding in isogeny graphs. *ANTS-XIV*, volume 4 of The Open Book Series, pp. 215–232, Mathematical Sciences Publishers, 2020.

[35] A. El Kaafarani, S. Katsumata, F. Pintore. Lossy CSI-FiSh: Efficient signature scheme with tight reduction to decisional CSIDH-512. *PKC 2020*, Part II, volume 13276 of LNCS, pp. 157–186, Springer, 2020.

[36] L. D. Feo. Mathematics of isogeny based cryptography. *arXiv*: 1711.04062, 2017.

[37] L. D. Feo, S. D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. *EUROCRYPT 2019*, Part III, volume 11478 of LNCS, pp. 759–789, Springer, 2019.

[38] L. D. Feo, C. D. de Saint Guilhem, T. B. Fouotsa, P. Kutas, A. Leroux, C. Petit, J. Silva, B. Wesolowski. Séta: Supersingular encryption from torsion attacks. *ASIACRYPT 2021*, Part IV, volume 13092 of LNCS, pp. 249–278, Springer, 2021.

[39] L. D. Feo, D. Kohel, A. Leroux, C. Petit, B. Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. *ASIACRYPT 2020*, Part I, volume 12492 of LNCS, pp. 64–93, Springer, 2020.

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

[41] L. D. Feo, A. Leroux, B. Wesolowski. New algorithms for the Deuring correspondence: SQISign twice as fast. *IACR Cryptology ePrint Archive*, 2022/234.

[42] T. B. Fouotsa, T. Moriya, C. Petit. M-SIDH and MD-SIDH: countering SIDH attacks by masking information. *IACR Cryptology ePrint Archive*, 2023/013.

[43] T. R. Fouotsa. SIDH with masked torsion point images. *IACR Cryptology ePrint Archive*, volume 2022/1054.

[44] T. B. Fouotsa, C. Petit. Sims: A simplification of SiGamal. *PQCrypto 2021*, volume 12841 of LNCS, pp. 277–295, Springer, 2021.

[45] E. Florit, B. Smith. An atlas of the Richelot isogeny graph. *RIMS Kôkyûroku Bessatsu*, volume B90, pp. 195–219, RIMS, Kyoto University, June 2022. `https://repository.kulib.kyoto-u.ac.jp/dspace/handle/2433/276282`.

[46] A. Fujioka, K. Takashima, K. Yoneyama. One-round authenticated group key exchange from isogenies. *ProvSec 2019*, volume 11821 of LNCS, pp. 330–338, Springer, 2019.

[47] S. Galbraith. Constructing isogenies between elliptic curves over finite fields. *Journal of Computational Mathematics*, volume 2, pp. 118–138, Global Science Press, 1999.

[48] S. Galbraith, L. Panny, B. Smith, F. Vercauteren. Quantum equivalence of the DLP and CDHP for group actions. *Mathematical Cryptology*, volume 1, number 1, pp. 40–44, 2021.

[49] S. D. Galbraith, C. Petit, J. Silva. Identification protocols and signature schemes based on supersingular isogeny problems. *Journal of Cryptology*, volume 33, number 1, pp. 130–175, Springer, 2020.

[50] S. D. Galbraith, F. Vercauteren. Computational problems in supersingular elliptic curve isogenies. *Quantum Information Processing*, volume 17, number 10, article 265, Springer, 2018.

[51] R. Ishibashi, K. Yoneyama. Post-quantum anonymous one-sided authenticated key exchange without random oracles. *PKC 2022*, Part II, volume 13178 of LNCS, pp. 35–65, Springer, 2022.

[52] D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. D. Feo, B. Hess, A. Jalali, B. Koziel et al. SIKE: Supersingular isogeny key encapsulation. *submission to the NIST's PQC standardization, round 3*, `https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/SIKE-Round3.zip`, 2020-10.（2023-04-11 閲覧）

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

[54] B. de Kock, K. Gjøsteen, M. Veroni. Practical isogeny-based key-exchange with optimal tightness. *SAC 2020*, volume 12804 of LNCS, pp. 451–479, Springer, 2020.

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

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

[57] T. Katsura, K. Takashima. Counting Richelot isogenies between superspecial abelian surfaces. *ANTS-XIV*, volume 4 of The Open Book Series, pp. 283–300, Mathematical Sciences Publishers, 2020.

[58] T. Kawashima, K. Takashima, Y. Aikawa, T. Takagi. An efficient authenticated key exchange from random self-reducibility on CSIDH. *ICISC 2020*, volume 12593 of LNCS, pp. 58–84, Springer, 2020.

[59] G. Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, volume 35, number 1, pp. 170–188, SIAM, 2005.

[60] G. Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)*, volume 22 of Leibniz International Proceedings in Informatics (LIPIcs), pp. 20–34, Schloss Dagstuhl, 2013.

[61] A. Leroux. A new isogeny representation and applications to cryptography. *ASIACRYPT 2022*, Part II, volume 13792 of LNCS, pp. 3–35, Springer, 2022.

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

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

[64] L. Maino, C. Martindale. An attack on SIDH with arbitrary starting curve (draft). *IACR Cryptology ePrint Archive*, 2022/1026.

[65] T. Moriya. Masked-degree SIDH. *IACR Cryptology ePrint Archive*, 2022/1019.

[66] T. Moriya, H. Onuki, T. Takagi. SiGamal: A supersingular isogeny-based PKE and its application to a PRF. *ASIACRYPT 2020*, Part II, volume 12492 of LNCS, pp. 551–580, Springer, 2020.

[67] M. Meyer, S. Reith. A faster way to the CSIDH. *INDOCRYPT 2018*, volume 11356 of LNCS, pp. 137–152, Springer, 2018.

[68] H. Montgomery, M. Zhandry. Full quantum equivalence of group action DLog and CDH, and more. *ASIACRYPT 2022*, Part I, volume 13791 of LNCS, pp. 3-32, Springer, 2022.

[69] H. Onuki, Y. Aikawa, T. Yamazaki, T. Takagi. A constant-time algorithm of CSIDH keeping two points. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences.*, volume E103-A, number 10, pp. 1174–1182, IEICE, 2020.

[70] H. Onuki. On the key generation in SQISign. *Number-Theoretic Methods in Cryptology (NutMic 2021).*

[71] R. Oudompheng, G. Pope. A note on reimplementing the Castryck–Decru attack and lessons learned for SageMath. *IACR Cryptology ePrint Archive*, 2022/1283.

[72] R. Oudompheng, L. Panny, G. Pope, et al. Sagemath reimplementation of the SIDH key recovery attack. `https://github.com/jack4818/Castryck-Decru-SageMath` (2023-04-10 閲覧).

[73] C. Peikert. He gives c-sieves on the CSIDH. *EUROCRYPT 2020*, Part II, volume 12106 of LNCS, pp. 463–492, Springer, 2020.

[74] C. Petit. Faster algorithms for isogeny problems using torsion point images. *ASIACRYPT 2017*, Part II, volume 10625 of LNCS, pp. 330–353, Springer, 2017.

[75] V. de Quehen, P. Kutas, C. Leonardi, C. Martindale, L. Panny, C. Petit, K. E. Stange. Improved torsion-point attacks on SIDH variants. *CRYPTO 2021*, Part III, volume 12827 of LNCS, pp. 432–470, Springer, 2021.

[76] O. Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. *arXiv*: 0406151, 2004.

[77] J. Renes. Computing isogenies between Montgomery curves using the action of (0, 0). *PQCrypto 2018*, volume 10786 of LNCS, pp. 229–247, Springer, 2018.

[78] D. Robert. Breaking SIDH in polynomial time. *IACR Cryptology ePrint Archive*, 2022/1038.

[79] A. Rostovtsev, A. Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptology ePrint Archive*, 2006/1450.

[80] R. Schoof. Nonsingular plane cubic curves over finite fields. *Journal of Combinatorial Theory, Series A*, volume 46, number 2, pp. 183–208, Academic Press, 1987.

[81] B. Smith. Explicit endomorphisms and correspondences. PhD thesis, The University of Sydney, 2005.

[82] B. Smith. Pre- and post-quantum Diffie–Hellman from groups, actions, and isogenies. *WAIFI 2018*, volume 11321 of LNCS, pp. 3–40, Springer, 2018.

[83] K. Takashima. Efficient algorithms for isogeny sequences and their cryptographic applications. *Mathematical Modelling for Next-Generation Cryptography: CREST Crypto-Math Project*, pp. 97–114, Springer, 2017.

[84] A. Udovenko, G. Vitto. Breaking the $IKEp182 challenge. *IACR Cryptology ePrint Archive*, 2021/1421.

[85] J. Vélu. Isogénies entre courbes elliptiques. *C.R. Acad. Sc. Paris, Séries A.*, volume 273, pp. 238–241, 1971.

[86] J. Voight. *Quaternion algebras*, Springer, June 2022.

[87] L. Washington. *Elliptic Curves: Number Theory and Cryptography*, CRC Press, 2nd edition, 2008.

[88] W. C. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'É.N.S., 4^e série*, volume 2, number 4, pp. 521–560, 1969.

[89] B. Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. *FOCS 2021*, pp. 1100–1111, IEEE, 2022.

[90] B. Wesolowski. Understanding and improving the Castryck–Decru attack on SIDH, August 2022. Available at the author's website.

第 6 章

ハッシュ関数に基づく署名技術

本章ではハッシュ関数に基づく署名技術についてまとめる．ハッシュ関数に基づく署名技術の安全性はハッシュ関数の第二原像攻撃に対する安全性に依存している．

ハッシュ関数に基づく署名技術は，最初に Lamport により one-time signature として提案された [10, 26]．また，この方式を改良した Winternitz one-time signature が Merkle [31] により述べられている．これらの方式は一組の公開鍵と秘密鍵を用いて一つのメッセージに署名を行う 1 回署名方式である．1 回署名方式とマークル木とを用いて複数回署名を行うことを可能とする方式が Merkle [30, 31] により述べられている．

6.1 ハッシュ関数に基づく署名技術の安全性の根拠となる問題

ハッシュ関数は任意長あるいは実用上十分な長さ以下の入力 $\{0,1\}$ 系列に対して固定長の $\{0,1\}$ 系列を出力する関数である．ハッシュ関数を $H : \mathcal{D} \to \mathcal{R}$ とする．ここで，$\mathcal{D}$ は任意長の $\{0,1\}$ 系列の集合 $\{0,1\}^*$ の部分集合であり，$\mathcal{R}$ は固定長の $\{0,1\}$ 系列の集合である．ハッシュ関数の第二原像攻撃は，第一原像 $X \in \mathcal{D}$ が与えられたとき，$X \neq X'$ かつ $H(X) = H(X')$ を満たす第二原像 $X' \in \mathcal{D}$ を求めるという問題を解くことを目的とする攻撃である．なお，第二原像攻撃に対する安全性は，しばしば，ハッシュ関数の各入力に対する出力が無作為に選択されるようなランダム関数であると仮定して評価される．このようなランダム関数はランダムオラクルとも呼ばれる．H がランダムオラクルであるとき，第二原像を得るのに必要な計算時間は $\Theta(|\mathcal{R}|)$ である．また，量子コンピュータでは，Grover の探索アルゴリズム [16] を用いることにより，第二原像を得るのに必要な計算時間は $\Theta(\sqrt{|\mathcal{R}|})$ となる．

本章で取り上げるハッシュ関数に基づく署名技術では，SHA-2 [34], SHA-3 [35], Haraka [24] のうちのいくつかのハッシュ関数を用いることが想定されている．これらのうち，SHA-2, SHA-3 は米国 NIST の指定する標準ハッシュ関数族である．

SHA-2 は固定長入出力の圧縮関数からなる Merkle-Damgård 構造 [9, 32] を有するハッシュ関数の族であり，Secure Hash Standard [34] のうち，SHA-1 を除く SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 からなる．SHA-2 の各ハッシュ関数の名称の末尾の数値は出力の bit 長を表す．SHA-3 は固定長入出力の置換を用いたスポンジ構造 [4] を有するハッシュ関数の族であり，SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256 からなる．SHA3-224, SHA3-256, SHA3-384, SHA3-512 については，末尾の数値は出力の bit 長を表す．なお，出力長は内部状態のうちスポンジ構造を有するハッシュ関数の安全性に大きく関わるキャパシティと呼ばれる部分の bit 長の半分である．SHAKE128, SHAKE256 については，出力長は任意に設定できる．なお，末尾の数値はキャパシティと呼ばれる部分の bit 長の半分である．SHA-2, SHA-3 の出力長および安全性の一覧を表 6.1 に示す．この表では，第二原像攻撃に対する安全性のみでなく，それとともにハッシュ関数の主な安全性要件である衝突攻撃，原

像攻撃に対する安全性も示されている．この表で，攻撃に対する安全性を表す数値 μ は bit 安全性と呼ばれ，攻撃に成功するために必要な計算時間（ハッシュ関数を構成する圧縮関数あるいは置換の計算回数）がおよそ 2^{μ} であることを示している．なお，SHA-224, SHA-256, SHA-512 の第二原像攻撃に対する安全性は Kelsey と Schneier により提案された攻撃 [25] に基づいて評価されており，$L(m) := \lceil \log_2(m/B) \rceil$ である．ここで，m は第一原像の bit 長であり，SHA-224, SHA-256 については $B = 512$，SHA-512 については $B = 1024$ である．

表 6.1: SHA-2, SHA-3 の安全性 [35]

ハッシュ関数	出力長	攻撃に対する安全性		
		衝突攻撃	原像攻撃	第二原像攻撃
SHA-224	224	112	224	$\min\{224, 256 - L(m)\}$
SHA-512/224	224	112	224	224
SHA-256	256	128	256	$256 - L(m)$
SHA-512/256	256	128	256	256
SHA-384	384	192	384	384
SHA-512	512	256	512	$512 - L(m)$
SHA3-224	224	112	224	224
SHA3-256	256	128	256	256
SHA3-384	384	192	384	384
SHA3-512	512	256	512	512
SHAKE128	d	$\min\{d/2, 128\}$	$\min\{d, 128\}$ 以上	$\min\{d, 128\}$
SHAKE256	d	$\min\{d/2, 256\}$	$\min\{d, 256\}$ 以上	$\min\{d, 256\}$

Haraka はブロック暗号 AES [12, 33] に基づく置換を用いた Davies-Meyer 構造 [37] に基づくハッシュ関数の族であり，Haraka-256, Haraka-512 からなる．Haraka-256, Haraka-512 の末尾の数値は入力の bit 長を表す．出力長はいずれも 256 bits である．Haraka は，ハッシュ関数に基づく署名技術での使用を想定し，短い入力に対して高い処理性能を達成するよう設計されている．Haraka の安全性については Haraka-256, Haraka-512 に対する原像攻撃が示されており [2]，特に，Haraka-512 が（第二）原像攻撃に対して 256 bit 安全性を有しないことが示されている．

本章で使用する記号・用語を以下にまとめる．

- $\{0,1\}$ 系列 α, β の連接を $\alpha \| \beta$ と表記する．
- $\ll$ は左論理シフトを表す．
- 整数 ν について $[\nu]_l$ は ν の長さ l bits の 2 進数表記を表す．
- $\mathbb{B} := \{0,1\}^8$ とする．

6.2 ハッシュ関数に基づく代表的な署名方式

6.2.1 Winternitz One-Time Signature

Winternitz one-time signature [31] は，一組の公開鍵と秘密鍵を用いて一つのメッセージに署名を行う 1 回署名方式である．この方式では，署名対象のメッセージのハッシュ値 N を b 進数表記の整数とみなす．N が ℓ_{m} 桁の b

進数 $N_{\ell_{\mathrm{m}}-1}N_{\ell_{\mathrm{m}}-2}\cdots N_1N_0$ で表記されるとする．このとき，$0 \le k \le \ell_{\mathrm{m}}-1$ について $N_k \in \{0,1,\ldots,b-1\}$ であり，$N = \sum_{k=0}^{\ell_{\mathrm{m}}-1} N_k 2^k$ である．さらに，N のチェックサムを $C := \sum_{k=0}^{\ell_{\mathrm{m}}-1}(b-1-N_k)$ と定義する．C が ℓ_{c} 桁の b 進数 $N_{\ell_{\mathrm{m}}+\ell_{\mathrm{c}}-1}N_{\ell_{\mathrm{m}}+\ell_{\mathrm{c}}-2}\cdots N_{\ell_{\mathrm{m}}+1}N_{\ell_{\mathrm{m}}}$ で表記されるとする．$\ell := \ell_{\mathrm{m}} + \ell_{\mathrm{c}}$ とする．

■鍵生成アルゴリズム　秘密鍵 $(x_0, x_1, \ldots, x_{\ell-1})$，公開鍵 $(pub_0, pub_1, \ldots, pub_{\ell-1})$ は以下のように生成される．

1. $x_0, x_1, \ldots, x_{\ell-1} \in \mathcal{D}$ を無作為に選択する．
2. $0 \le k \le \ell-1$ について $pub_k := H^{b-1}(x_k) := \underbrace{H(H(\cdots(H}_{b-1 \text{ times}}(x_k))\cdots))$ とする．

■署名アルゴリズム　メッセージのハッシュ値 N の署名 $(s_0, s_1, \ldots, s_{\ell-1})$ は以下のように生成される．

1. $0 \le k \le \ell-1$ について $s_k := H^{N_k}(x_k)$ とする．

■検証アルゴリズム　メッセージのハッシュ値 N とその署名 $(s_0, s_1, \ldots, s_{\ell-1})$ の検証は以下のように行われる．

1. $0 \le k \le \ell-1$ について $pub_k = H^{b-1-N_k}(s_k)$ かつそのときに限り，$(s_0, s_1, \ldots, s_{\ell-1})$ は N の正しい署名である．

仮にチェックサムが導入されていないとすると，N の署名 $s_0, s_1, \ldots, s_{\ell_{\mathrm{m}}-1}$ が得られたとき，$0 \le k \le \ell_{\mathrm{m}}-1$ について $N'_k \ge N_k$ を満たす N' について，$s'_k := H^{N'_k-N_k}(s_k)$ によって，署名 $(s'_0, s'_1, \ldots, s'_{\ell_{\mathrm{m}}-1})$ が容易に偽造できる．

Winternitz one-time signature の偽造不能性は，Dods ら [13] により論じられている．Winternitz one-time signature に基づく方式については，Lafrance と Menezes [28] によりまとめられている．

6.2.2　マークル木を用いた署名方式

1 回署名方式を用いて複数のメッセージに署名を行う場合，メッセージの個数と同じ個数の公開鍵と秘密鍵の組が必要となる．マークル木を用いることにより，このような複数回署名方式の公開鍵の大きさを削減できる [30].

2^h 個のメッセージに署名を行うための 1 回署名の公開鍵を $pk_0, pk_1, \ldots, pk_{2^h-1}$ とする．このとき，高さが h，すなわち，葉の個数が 2^h のマークル木は以下のように構成される．高さ $j(\ge 0)$ の左から $i(\ge 0)$ 番目の節点を $v_{i,j}$ と表記する．$v_{i,j}$ は以下のように計算される．

1. $0 \le i \le 2^h-1$ について，$v_{i,0} := H(pk_i)$ とする．
2. $1 \le j \le h$ に対し，$0 \le i \le 2^{h-j}-1$ について，$v_{i,j} := H(v_{2i,j-1} \| v_{2i+1,j-1})$ とする．

この署名方式の公開鍵は $v_{0,h}$ である．秘密鍵は 1 回署名の公開鍵 $pk_0, pk_1, \ldots, pk_{2^h-1}$ に対応するすべての秘密鍵である．i 個目のメッセージの署名を検証するためには，$v_{0,h}$ を用いて pk_i が正しいことを検証する必要がある．このために，i 個目のメッセージの署名には，マークル木の $v_{i,0}$ から $v_{0,h}$ に至る経路上の各節点の，経路上にない子節点が含まれる．これらの節点の列は認証パスと呼ばれる．

6.2.3　マークル木の階層構造による署名方式

前節で述べた一つのマークル木を用いた署名方式では，鍵生成時にすべての 1 回署名の公開鍵と秘密鍵を生成する必要があり，例えば，2^{50} 個の署名を行うために高さ 50 のマークル木を構成することは，所要計算時間の観点から非実用

的である．このような多数のメッセージに署名を行う際には，マークル木の階層構造による署名方式が提案されている [19].

この署名方式のマークル木の階層構造の階層数を L とする．この署名方式では，$0 \leq i \leq L-1$ について，第 i 層のマークル木の高さはすべて等しく h_i であると仮定する．このとき，この署名方式は $2^{\sum_{i=0}^{L-1} h_i}$ 個のメッセージに署名できる．

この署名方式で，$0 \leq i \leq L-1$ について，第 i 層のマークル木は $2^{\sum_{j=0}^{i-1} h_j}$ 個存在する．第 0 層（最上層）のマークル木は 1 個であり，その根がこの署名方式の公開鍵となる．したがって，この公開鍵を生成する際には，1 回署名の公開鍵と秘密鍵の組を 2^{h_0} 個だけ生成すれば良い．$0 \leq i < L-1$ について，第 i 層の各マークル木は第 $(i+1)$ 層の 2^{h_i} 個のマークル木の根を署名するために使用される．第 $(L-1)$ 層（最下層）のマークル木は，それぞれ $2^{h_{L-1}}$ 個のメッセージの署名に使用される．

この署名方式では，一つのメッセージの署名の際に，各層についてそれぞれ一つのマークル木を生成しておけば十分である．各メッセージの署名は，そのメッセージに対する最下層のマークル木による署名と，$0 \leq i < L-1$ について，そのメッセージの署名の際に使用された第 i 層のマークル木による第 $(i+1)$ 層のマークル木の根の署名からなる．この署名方式について，階層数 $L = 3$，各階層のマークル木の高さ $h_0 = h_1 = h_2 = 3$ の模式図を図 6.1 に示す．灰色の節点は認証パスをなす節点である．

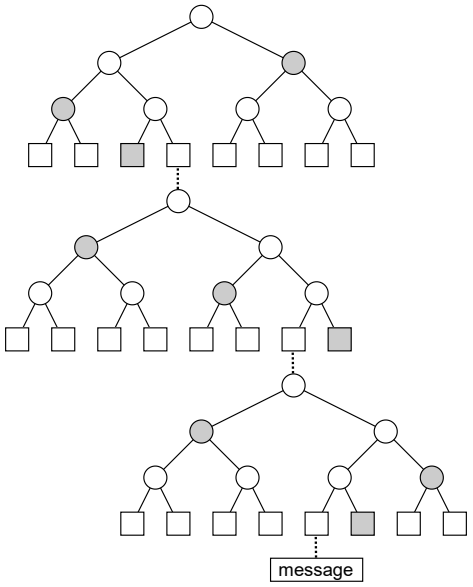


図 6.1: マークル木の階層構造による署名方式

6.2.4 プレフィクスとビットマスク

プレフィクスは，ハッシュ関数に基づく署名方式の処理において，すべてのハッシュ関数の計算がそれぞれ異なる入力に対して行われるよう入力に付加される系列である．プレフィクスは，Lighton と Micali [27] により，security string という名称で，ハッシュ関数に基づく署名方式の安全性をハッシュ関数の第二原像攻撃に対する安全性にタイトに帰着するために導入された．

ビットマスクは，Dahmen ら [11] により，ハッシュ関数に基づく署名方式の安全性をハッシュ関数の第二原像攻撃

に対する安全性に帰着するために導入された．ビットマスクは乱数系列であり，ハッシュ関数への入力をランダム化するために，bit ごとの排他的論理和により入力に加えられる．

6.3 ハッシュ関数に基づく主要な署名方式

本章で取り上げるハッシュ関数に基づく署名方式を表 6.2 に示す．

表 6.2: ハッシュ関数に基づく署名方式

文献	暗号化	鍵交換	署　名
Lighton-Micali Hash-Based Signatures [29, 8]			○
eXtended Merkle Signature Scheme (XMSS) [17, 8]			○
SPHINCS$^+$ [1]			○

NIST SP 800-208 [8] は，以下のハッシュ関数に基づく stateful な署名方式を規定している．

- Lighton-Micali Signatures (LMS), Hierarchical Signature System (HSS) [29]
- eXtended Merkle Signature Scheme (XMSS), multi-tree XMSS (XMSS^{MT}) [17]

LMS は Lighton と Micali による署名方式 [27] に基づく．HSS, XMSS^{MT} はそれぞれ，6.2.3 節で述べられたような，LMS, XMSS の階層構造による署名方式である．ハッシュ関数に基づく stateful な署名方式では，同一の秘密鍵が複数のメッセージの署名に使用されることがないように秘密鍵を管理することが必須である．

SPHINCS$^+$ [1] は 2022 年 7 月に NIST Post Quantum Cryptography Standardization Process で標準化の候補アルゴリズムの一つに選出された．SPHINCS$^+$ はハッシュ関数に基づく stateless な署名方式であり，stateful な方式に求められるような秘密鍵の管理が不要な方式である．SPHINCS$^+$ は SPHINCS [5] の改良版として提案され [6, 7]，その後も NIST Post Quantum Cryptography Standardization Process で改良が行われた．

6.3.1 Lighton-Micali Hash-Based Signatures

IRTF RFC 8554 [29] では，LMS, HSS が述べられている．LMS, HSS では Winternitz one-time signature に基づく LM-OTS が用いられる．LMS は LM-OTS とマークル木とを用いて構成され，この構造は LMS 木と呼ばれる．HSS は LMS 木の階層構造による署名方式である．LM-OTS, LMS, HSS にはそれぞれ，それらのアルゴリズムで用いられるハッシュ関数，パラメータセットに対応する長さ 4 Bytes の符号なし整数が割り当てられる．これは typecode と呼ばれる．

本節では，ハッシュ関数 $H : \mathcal{D} \to \mathcal{R}$ について，$\mathcal{D} = \bigcup_{i \geq 0} \{0,1\}^{8i}, \mathcal{R} = \{0,1\}^{8n}$ とする．すなわち，H は任意長の Byte 系列を入力とし，長さ n Bytes の系列を出力する．なお，本節の表記は概ね [15] の表記法に基づいている．

6.3.1.1 LM-OTS

$w \in \{1, 2, 4, 8\}$ を Winternitz 係数の幅（bit 長）とする．p を LM-OTS を構成する長さ n Bytes の系列の個数とする．*type* を typecode とする．ハッシュ関数 H を用いて以下の関数が定義される．

$$H^i_{I,q,d}(x; j) := \begin{cases} x & i = 0 \text{ のとき} \\ H(I \| [q]_{32} \| [d]_{16} \| [i + j - 1]_8 \| H^{i-1}_{I,q,d}(x; j)) & i \geq 1 \text{ のとき} \end{cases}$$

ここで，I は長さ 16 Bytes の系列であり，LM-OTS が，LMS や HSS においてどのマークル木で使用されるかを表す．q は長さ 4 Bytes の整数であり，LM-OTS の公開鍵が対応するマークル木の葉を表す．

■鍵生成アルゴリズム　与えられた I, q に対応する秘密鍵と公開鍵の組を生成するアルゴリズムを以下に示す．

1. $x_0, x_1, \ldots, x_{p-1} \in \{0,1\}^{8n}$ を無作為に選択する．
2. $0 \leq i \leq p-1$ について，$y_i := H_{I,q,i}^{2^w-1}(x_i; 0)$ とする（図 6.2）．
3. $K := H(I \| [q]_{32} \| [\mathtt{8080}]_{16} \| y_0 \| y_1 \| \cdots \| y_{p-1})$ とする．

秘密鍵は $(type, I, q, x_0, x_1, \ldots, x_{p-1})$ である．公開鍵は $[type]_{32} \| I \| [q]_{32} \| K$ である．

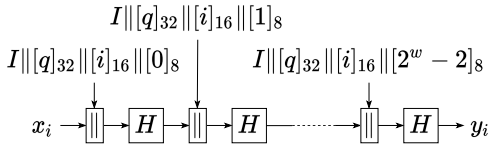


図 6.2: y_i の計算

■署名アルゴリズム　$\mathsf{Checksum} : \{0,1\}^{8n} \to \{0,1\}^{16}$ は以下のように定義される関数である．

$$\mathsf{Checksum}(S) := \Big(\sum_{i=0}^{8n/w-1} (2^w - 1 - d_i) \Big) \ll ls$$

ここで，$S = d_0 \| d_1 \| \cdots \| d_{8n/w-1}$ であり，$0 \leq i \leq 8n/w - 1$ について $d_i \in \{0,1\}^w$ である．上式で d_i は整数とみなされている．また，ls は n, w に応じて決まる整数であり，$16 - ls$ が w の倍数であり，かつ，$\mathsf{Checksum}(S)$ の 2 進数表記の長さが常に $16 - ls$ bits 以下となるよう定められる．

メッセージ M に対する署名アルゴリズムを以下に示す．

1. アルゴリズムのパラメータセットに応じて $type, n, p, w$ の値を定める．
2. $C \in \{0,1\}^{8n}$ を無作為に選択し，$Q := H(I \| [q]_{32} \| [\mathtt{8181}]_{16} \| C \| M)$，$c := \mathsf{Checksum}(Q)$ とする．
3. $Q \| c \in \{0,1\}^{wp}$ をそれぞれ長さ w bits のブロック $V_0, V_1, \ldots, V_{p-1}$ に分割する．
4. $0 \leq i \leq p-1$ について，$\sigma_i := H_{I,q,i}^{V_i}(x_i; 0)$ とする．ここで，V_i は整数とみなされる．

メッセージ M に対する署名は $\sigma := [type]_{32} \| C \| \sigma_0 \| \sigma_1 \| \cdots \| \sigma_{p-1}$ である．

■検証アルゴリズム　鍵生成と署名のアルゴリズムより容易に導出されるので，詳細は [15, 29] を参照のこと．

6.3.1.2　LMS

LMS は LM-OTS と同じハッシュ関数を用いることが推奨されている．LMS のマークル木の各節点には番号が付される．根の番号は 1 であり，番号 ν の節点の左の子と右の子の番号はそれぞれ $2\nu, 2\nu + 1$ である．

h はマークル木の高さを表す．m はマークル木の各節点に対応する系列の Byte 長を表し，これはハッシュ関数の出力長である．h, m の値は LMS の typecode によって定められる．

■鍵生成アルゴリズム

1. $I \in \{0,1\}^{128}$ を無作為に選択する．

2. $0 \leq q \leq 2^h - 1$ について，LM-OTS の公開鍵と秘密鍵の組 (pk^q, sk^q) を生成する．
3. マークル木の番号 r の節点に対応する系列 $T[r]$ は以下のように定義される．

$$T[r] := \begin{cases} H(I \| [r]_{32} \| [\mathsf{8282}]_{16} \| pk^{r-2^h}) & 2^h \leq r \leq 2^{h+1} - 1 \text{ のとき} \\ H(I \| [r]_{32} \| [\mathsf{8383}]_{16} \| T[2r] \| T[2r+1]) & 1 \leq r \leq 2^h - 1 \text{ のとき} \end{cases}$$

公開鍵は $[type]_{32} \| [otstype]_{32} \| I \| T[1]$ である．秘密鍵は $sk^0, sk^1, \ldots, sk^{2^h-1}$ である．ここで $type$, $otstype$ はそれぞれ LMS, LMS-OTS の typecode を表す．

■署名アルゴリズム　以下では $0 \leq q \leq 2^h - 1$ である．最初の署名では $q = 0$ とし，1 回の署名ごとに q の値を 1 だけ増やすことにより，LM-OTS の各秘密鍵を複数回使用しないようにしなければならない．

メッセージ M に対する署名アルゴリズムを以下に示す．

1. 秘密鍵 sk^q を用いて LM-OTS による M の署名 σ を計算する．
2. $0 \leq i \leq h - 1$ について，$p_i := T[\lfloor (q + 2^h)/2^i \rfloor \oplus 1]$ とする．

M の署名は $[q]_{32} \| \sigma \| [type]_{32} \| p_0 \| p_1 \| \cdots \| p_{h-1}$ である．$p_0, p_1, \ldots, p_{h-1}$ は LM-OTS の公開鍵 pk^q の認証パスである．

■検証アルゴリズム　鍵生成と署名のアルゴリズムより容易に導出されるので，詳細は [15, 29] を参照のこと．

6.3.1.3 HSS

HSS は 6.2.3 節で述べられたような LMS 木の階層構造による署名方式である．階層数 L は $1 \leq L \leq 8$ を満たす．

HSS は stateful な署名方式なので，メッセージの署名の際に最下層の LMS 木の秘密鍵が使い尽くされたとき，そのメッセージの署名で使用された L 個の LMS 木のうち，第 d 層から下のすべての LMS 木の秘密鍵が使い尽くされた最小の d を求める．$d = 0$ のときは，新たな署名を作成しない．$d \geq 1$ のとき，$d \leq i \leq L - 1$ について，秘密鍵の使い尽くされた第 i 層の LMS 木を破棄し，それぞれに替わる新たな LMS 木を使用して新しいメッセージへの署名を行う．

HSS の鍵生成，署名，検証の各アルゴリズムについての詳細は [15, 29] を参照のこと．

6.3.1.4 パラメータの設定と安全性

LMS の適応的選択メッセージ攻撃に対する存在偽造不能性（EUF-CMA）については，Katz [22] や Fluhrer [15] によりランダムオラクルモデルを仮定して示されており，また，Eaton [14] により量子ランダムオラクルモデルを仮定して示されている．なお，IRTF RFC 8554 [29] には，ハッシュ関数は第二原像攻撃に対する安全性を満たさなければならないと記されている．

NIST SP 800-208 [8] では，ハッシュ関数として SHA-256, SHA-256/192, SHAKE256/256, SHAKE256/192 を使用することが認可されている．ここで，SHA-256/192 は SHA-256 の出力の上位 192 bits を出力とするハッシュ関数である．SHAKE256/256, SHAKE256/192 はそれぞれ，出力長を 256 bits，192 bits とする SHAKE256 である．NIST SP 800-208 [8] と IRTF RFC 8554 [29] の両方に掲載されている SHA-256 を用いる場合の LM-OTS, LMS のパラメータセットの値の一覧をそれぞれ表 6.3, 6.4 に示す．

6.3.2 XMSS: eXtended Merkle Signature Scheme

XMSS は [3, 19] で提案された方式の改良版 [20] に基づく署名方式であり，Winternitz one-time signature に基づく 1 回署名方式 [21] を用いる．

表 6.3: LM-OTS のパラメータセットと署名長（単位は Byte）

名称	n	w	p	ls	署名長
LMOTS_SHA256_N32_W1	32	1	265	7	8516
LMOTS_SHA256_N32_W2	32	2	133	6	4292
LMOTS_SHA256_N32_W4	32	4	67	4	2180
LMOTS_SHA256_N32_W8	32	8	34	0	1124

表 6.4: LMS のパラメータセット

名称	m	h
LMS_SHA256_M32_H5	32	5
LMS_SHA256_M32_H10	32	10
LMS_SHA256_M32_H15	32	15
LMS_SHA256_M32_H20	32	20
LMS_SHA256_M32_H25	32	25

XMSS では三つの鍵付きハッシュ関数 $\mathsf{F}, \mathsf{H}, \mathsf{H}_{\mathrm{msg}}$ と擬似ランダム関数 R が用いられる．いずれも出力の Byte 長は等しく，これを n とする．F の入力は Byte 長 n の鍵と Byte 長 n の系列である．H の入力は Byte 長 n の鍵と Byte 長 $2n$ の系列である．$\mathsf{H}_{\mathrm{msg}}$ の入力は Byte 長 $3n$ の鍵と任意 Byte 長の系列である．R の入力は Byte 長 n の鍵と Byte 長 32 の系列である．これらの関数は SHA-2 [34] または SHA-3 [35] を用いて定義される．例えば，$n = 32$ のとき，SHA-256 を用いて以下のように定義される．

$$\begin{aligned}
\mathsf{F}(k, x) &:= \text{SHA-256}([0]_{256} \| k \| x) \\
\mathsf{H}(k, x) &:= \text{SHA-256}([1]_{256} \| k \| x) \\
\mathsf{H}_{\mathrm{msg}}(k, x) &:= \text{SHA-256}([2]_{256} \| k \| x) \\
\mathsf{R}(k, x) &:= \text{SHA-256}([3]_{256} \| k \| x)
\end{aligned}$$

XMSS では，ハッシュ関数の呼び出しをランダム化するために，それぞれのハッシュ関数の呼び出しで，鍵とビットマスクが用いられる．これらは擬似ランダム関数を用いて生成され，入力として Byte 系列の seed と長さ 32 Bytes のアドレス ADRS が与えられる．アドレスは 3 種あり，それぞれ OTS ハッシュアドレス，L 木アドレス，ハッシュ木アドレスと呼ばれる．それらの構造を図 6.3 に示す．

layer address	(32 bits)
tree address	(64 bits)
type = 0	(32 bits)
OTS address	(32 bits)
chain address	(32 bits)
hash address	(32 bits)
keyAndMask	(32 bits)

(a) OTS ハッシュアドレス

layer address	(32 bits)
tree address	(64 bits)
type = 1	(32 bits)
L-tree address	(32 bits)
tree height	(32 bits)
tree index	(32 bits)
keyAndMask	(32 bits)

(b) L 木アドレス

layer address	(32 bits)
tree address	(64 bits)
type = 2	(32 bits)
Padding = 0	(32 bits)
tree height	(32 bits)
tree index	(32 bits)
keyAndMask	(32 bits)

(c) ハッシュ木アドレス

図 6.3: アドレスの構造

6.3.2.1 WOTS$^+$

$w \in \{4, 16\}$ は Winternitz パラメータと呼ばれる．$\ell := \ell_1 + \ell_2$ は公開鍵，秘密鍵，署名を構成する Byte 長 n の要素の個数を表す．ここで，

$$\ell_1 := \lceil 8n / \log_2 w \rceil, \quad \ell_2 := \lfloor \log_2(\ell_1(w-1)) / \log_2 w \rfloor + 1$$

である．

■チェイニング関数　チェイニング関数 chain の入力は，長さ n Bytes の系列 X，スタートインデクス i，ステップ数 s，長さ 32 Bytes のアドレス ADRS，長さ n Bytes のシード seed であり，以下のように定義される．

$$\mathsf{chain}(X, i, s, \mathsf{seed}, \mathrm{ADRS}) := \begin{cases} X & s = 0 \text{ のとき} \\ \mathtt{NULL} & i + s \geq w \text{ のとき} \\ \mathsf{F}(Key, \mathsf{chain}(X, i, s-1, \mathsf{seed}, \mathrm{ADRS}) \oplus BM) & \text{それ以外のとき} \end{cases}$$

ここで，

$$Key := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [i+s-1]_{32} \| [0]_{32}), \quad BM := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [i+s-1]_{32} \| [1]_{32})$$

である．なお，ADRS′ は ADRS の上位 24 Bytes であり，例えば，$\mathrm{ADRS}' \| [i+s-1]_{32} \| [0]_{32}$ は図 6.3a の ADRS の hash address, keyAndMask の値をそれぞれ，$[i+s-1]_{32}, [0]_{32}$ とすることを表している．

■鍵生成アルゴリズム　入力は ADRS, seed である．

1. $0 \leq i \leq \ell - 1$ について，$sk_i \in \{0,1\}^{8n}$ を無作為に選択する．
2. $0 \leq i \leq \ell - 1$ について，ADRS の chain address の値を $[i]_{32}$ とし，

   $$pk_i := \mathsf{chain}(sk_i, 0, w-1, \mathsf{seed}, \mathrm{ADRS})$$

   とする．この計算を図 6.4 に示す．この図で

   $$Key_j := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [j]_{32} \| [0]_{32}), \quad BM_j := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [j]_{32} \| [1]_{32})$$

   である．

公開鍵は $pk := (pk_0, pk_1, \ldots, pk_{\ell-1})$ である．秘密鍵は $sk := (sk_0, sk_1, \ldots, sk_{\ell-1})$ である．

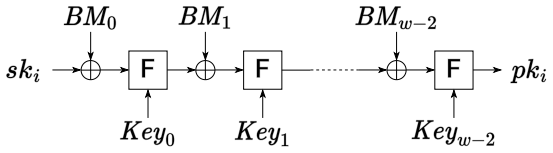


図 6.4: pk_i の計算

■署名アルゴリズム　入力は Byte 長 n のメッセージ M，秘密鍵 sk，アドレス ADRS，シード seed である．

1. M をそれぞれ長さ $\log_2 w$ bits の ℓ_1 個のブロックに分割し，先頭から順に $M_0, M_1, \ldots, M_{\ell_1-1}$ とする．これらを整数とみなすと，$0 \leq i \leq \ell_1 - 1$ について，$M_i \in \{0, 1, \ldots, w-1\}$ である．
2. $C := \sum_{i=0}^{\ell_1-1} (w - 1 - M_i)$ とする．
3. $C \cdot 2^{8-(\ell_2 \log_2 w \bmod 8)}$ を長さ $\lceil (\ell_2 \log_2 w)/8 \rceil$ Bytes の系列とみなし，それぞれ長さ $\log_2 w$ bits の ℓ_2 個のブロックに分割し，先頭から順に $M_{\ell_1}, M_{\ell_1+1}, \ldots, M_{\ell-1}$ とする．
4. $0 \leq i \leq \ell - 1$ について，ADRS の chain address の値を i とし，

   $$sig_i := \mathsf{chain}(sk_i, 0, M_i, \mathsf{seed}, \mathrm{ADRS})$$

   とする．

メッセージ M に対する署名は $sig_0, sig_1, \ldots, sig_{\ell-1}$ である．

■検証アルゴリズム　鍵生成と署名のアルゴリズムより容易に導出されるので，詳細は [17] を参照のこと．

6.3.2.2　XMSS

XMSS はマークル木を用いた署名方式であり，公開鍵と秘密鍵の各組は完全二分木に対応付けられる．

XMSS のハッシュ木の構成のために，ランダム化ハッシュ関数 RH が導入されている．RH の入力は長さ n Bytes の $LEFT, RIGHT$，長さ n Bytes のシード seed，長さ 32 Bytes のアドレス ADRS であり，以下のように定義される．

$$\mathsf{RH}(LEFT, RIGHT, \mathsf{seed}, \mathrm{ADRS}) := \mathsf{H}(Key, (LEFT \oplus BM_0) \| (RIGHT \oplus BM_1))$$

ここで，

$$Key := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [0]_{32}), \quad BM_0 := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [1]_{32}), \quad BM_1 := \mathsf{R}(\mathsf{seed}, \mathrm{ADRS}' \| [2]_{32})$$

である．なお，ADRS$'$ は ADRS の上位 28 Bytes であり，例えば，ADRS$' \| [0]_{32}$ は ADRS の図 6.3 の keyAndMask の値を $[0]_{32}$ とすることを表している．

秘密鍵の生成には [3] に示されているような擬似ランダム鍵生成法を用いることが許容されているが，その安全性は少なくとも XMSS の安全性と同等でなければならない．

■鍵生成アルゴリズム　鍵生成アルゴリズムではマークル木が構成され，その各葉には WOTS$^+$ の公開鍵が対応する．WOTS$^+$ の公開鍵に対して L 木と呼ばれるハッシュ木が構成され，その木の根のハッシュ値が XMSS のマークル木の葉に割り当てられる．L 木の高さ $j(\geq 0)$ の左から $i(\geq 0)$ 番目の節点を $\mathrm{Node}_{i,j}$ と表記する．L 木は以下にしたがって構成される．入力は WOTS$^+$ の公開鍵 $pk := (pk_0, pk_1, \ldots, pk_{\ell-1})$，L 木アドレス ADRS，シード seed である．

1. $0 \leq i \leq \ell - 1$ について，$\mathrm{Node}_{i,0} := pk_i$ とする．
2. $j \geq 0$ について，根が得られるまで以下にしたがって $\mathrm{Node}_{i,j+1}$ を計算する．なお，値の定義された $\mathrm{Node}_{i,j}$ の個数を ℓ' とする．
   (a) $0 \leq i < \lfloor \ell'/2 \rfloor$ について，$\mathrm{Node}_{i,j+1} := \mathsf{RH}(\mathrm{Node}_{2i,j}, \mathrm{Node}_{2i+1,j}, \mathsf{seed}, \mathrm{ADRS})$ とする．ここで，ADRS の tree height を $[j]_{32}$，tree index を $[i]_{32}$ とする．さらに，ℓ' が奇数のとき，$\mathrm{Node}_{\lfloor \ell'/2 \rfloor, j+1} := \mathrm{Node}_{\ell'-1,j}$ とする．
   (b) $j \leftarrow j + 1$ とする．

鍵生成アルゴリズムで構成されるマークル木の高さを h とすると，このマークル木には 2^h 個の葉が存在する．このマークル木に対応する 2^h 個の WOTS$^+$ の公開鍵，それらの L 木，さらに，このマークル木の計算に用いられる OTS ハッシュアドレス，L 木アドレス，ハッシュ木アドレスの layer address，tree address はすべて，それぞれ $[0]_{32}$，$[0]_{64}$ である．左から $k\,(\geq 0)$ 番目の葉に対応する OTS ハッシュアドレスの OTS address，L 木アドレスの L-tree address は $[k]_{32}$ である．

鍵生成アルゴリズムで構成されるマークル木の葉は対応する L 木の根である．葉以外の節点は L 木の節点と同じ方法で計算される．なお，このマークル木は完全二分木なので，上述の L 木の計算手続きで，ℓ' は常に偶数となる．

秘密鍵は，2^h 個の WOTS$^+$ の秘密鍵，次の署名に使用される WOTS$^+$ の秘密鍵に対応するマークル木の葉の番号 idx，署名されるメッセージのハッシュの計算に使用される SK_{PRF}，マークル木の根 $root$，seed である．公開鍵は，マークル木の根，seed である．ここで，SK_{PRF} と seed はこの鍵生成アルゴリズムで無作為に選択される長さ n Bytes の系列である．また，公開鍵には識別子 OID が付される．

■署名アルゴリズム　メッセージ M の署名は，署名に使用される WOTS$^+$ の秘密鍵の番号 idx，M のダイジェストの計算に使用される乱数 r，WOTS$^+$ による署名，マークル木の idx 番目の葉の認証パスからなる．

1. M のダイジェストを $M' := \mathsf{H}_{\mathrm{msg}}(r \| root \| [idx]_{8n}, M)$ とする．ここで，$r := \mathsf{R}(SK_{\mathrm{PRF}}, [idx]_{32})$ である．
2. WOTS$^+$ の idx 番目の秘密鍵を用いて M' に署名し，マークル木の idx 番目の葉の認証パスを計算する．

WOTS$^+$ の同じ秘密鍵が 2 回以上使用されないよう，idx は $idx \leftarrow idx + 1$ により更新される．

■検証アルゴリズム　鍵生成と署名のアルゴリズムより容易に導出されるので，詳細は [17] を参照のこと．

6.3.2.3　XMSS^{MT}

XMSS^{MT} は，6.2.3 節のマークル木の階層構造による署名方式に相当する．XMSS^{MT} 木はハイパー木と呼ばれ，d 層の XMSS 木からなる．ここで，XMSS 木は 6.3.2.2 節の鍵生成アルゴリズムで生成される L 木とマークル木からなる木を表す．第 $(d-1)$ 層と第 0 層はそれぞれ，XMSS^{MT} 木の根と葉に相当する[*1]．すべての XMSS 木の高さは等しく，Winternitz パラメータもすべて同じ値が用いられる．第 x 層の左から y 番目の XMSS 木の構成で使用される OTS ハッシュアドレス，L 木アドレス，ハッシュ木アドレスの layer address と tree address は，それぞれ $[x]_{32}$，$[y]_{32}$ である．

XMSS^{MT} の鍵生成，署名，検証の各アルゴリズムについての詳細は [17] を参照のこと．

6.3.2.4　パラメータの設定と安全性

Kampanakis と Fluhrer [23] により，LMS と XMSS の比較が論じられている．

Hülsing [20] らは，XMSS について安全性証明を与え，適応的選択メッセージ攻撃に対する存在偽造不能性（EUF-CMA）を満たすことを鍵付きハッシュ関数 $\mathsf{F}, \mathsf{H}, \mathsf{H}_{\mathrm{msg}}$ と擬似ランダム関数 R の以下の安全性に帰着している．

- F が以下の性質を満たすこと
  - multi-function, multi-target second preimage resistance (MM-SPR)
  - すべての出力が 2 個以上の原像を持つこと
- H が MM-SPR を満たすこと
- $\mathsf{H}_{\mathrm{msg}}$ が multi-target extended target collision resistance (M-ETCR) を満たすこと
- R が擬似ランダム関数 (PRF) であること

ここで，MM-SPR, M-ETCR は，$\mathsf{F}, \mathsf{H}, \mathsf{H}_{\mathrm{msg}}$ の構成に用いられるハッシュ関数の第二原像攻撃に対する安全性に基づく性質である．一方，PRF は，秘密鍵入力を有するハッシュ関数が擬似ランダム関数であることを要求する．さらに，R による鍵とビットマスクの生成については，ハッシュ関数がランダムオラクルであることが仮定される．

IRTF RFC 8391 [17] では，上述の XMSS の安全性に関する結果に基づいて，$n = 32, 64$ のとき，それぞれ，256 bit 安全性，512 bit 安全性が提供されると記されている．また，量子計算機を用いた攻撃に対してはそれぞれ，128 bit 安全性，256 bit 安全性が提供されると記されている．

IRTF RFC 8391 [17] では，ハッシュ関数として SHA-256 を用いることが要求されているが，オプションとして SHAKE128/256, SHA-512, SHAKE256/512 を用いることが記されている．一方, NIST SP 800-208 では, SHA-256, SHA-256/192, SHAKE256/256, SHAKE256/192 を用いることが認可されている．NIST SP 800-208 [8] と IRTF

[*1] IRTF RFC 8391 [17] では，各層の番号付けが 6.2.3 節の番号付けとは逆順であり，本稿でもそれに従って記述する．

RFC 8391 [17] の両方に掲載されている SHA-256 を用いる場合の WOTS$^+$, XMSS, XMSS^{MT} のパラメータセットの値の一覧をそれぞれ表 6.5, 6.6, 6.7 に示す.

表 6.5: WOTS$^+$ のパラメータセット

名称	n	w	ℓ
WOTSP-SHA2_256	32	16	67

表 6.6: XMSS のパラメータセットと署名長（単位は Byte）

名称	n	w	ℓ	h	署名長
XMSS-SHA2_10_256	32	16	67	10	2,500
XMSS-SHA2_16_256	32	16	67	16	2,692
XMSS-SHA2_20_256	32	16	67	20	2,820

表 6.7: XMSS^{MT} のパラメータセットと署名長（単位は Byte）

名称	n	w	ℓ	h	d	署名長
XMSSMT-SHA2_20/2_256	32	16	67	20	2	4,963
XMSSMT-SHA2_20/4_256	32	16	67	20	4	9,251
XMSSMT-SHA2_40/2_256	32	16	67	40	2	5,605
XMSSMT-SHA2_40/4_256	32	16	67	40	4	9,893
XMSSMT-SHA2_40/8_256	32	16	67	40	8	18,469
XMSSMT-SHA2_60/3_256	32	16	67	60	3	8,392
XMSSMT-SHA2_60/6_256	32	16	67	60	6	14,824
XMSSMT-SHA2_60/12_256	32	16	67	60	12	27,688

6.3.3 SPHINCS$^+$

SPHINCS$^+$[1] は 6.2.3 節のマークル木の階層構造による署名方式に基づく方式である．ただし，6.3.1 節と 6.3.2 節で述べられた HSS, XMSS^{MT} とは異なり，stateless な署名方式である．

SPHINCS$^+$ では, 以下のような, いくつかの tweakable ハッシュ関数 $\mathbf{T}_\ell$, 二つの擬似ランダム関数 $\mathbf{PRF}, \mathbf{PRF_{msg}}$, 一つの鍵付きハッシュ関数 $\mathbf{H_{msg}}$ が用いられる．

$$\mathbf{T}_\ell : \mathbb{B}^n \times \mathbb{B}^{32} \times \mathbb{B}^{\ell n} \to \mathbb{B}^n \qquad \mathbf{PRF} : \mathbb{B}^n \times \mathbb{B}^{32} \to \mathbb{B}^n \qquad \mathbf{H_{msg}} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^* \to \mathbb{B}^m$$
$$\mathbf{PRF_{msg}} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^* \to \mathbb{B}^n$$

以下では，$\mathbf{T}_1, \mathbf{T}_2$ について，$\mathbf{F} := \mathbf{T}_1$, $\mathbf{H} := \mathbf{T}_2$ の表記が用いられる．

SPHINCS$^+$ では，図 6.5 に示す 7 種のアドレスが用いられる．どのアドレスも長さは 32 Bytes である．

SPHINCS$^+$ は XMSS^{MT} とよく似た構造を有していることから，以下では両者の相違点を中心に述べる．

6.3.3.1 WOTS$^+$

SPHINCS$^+$ でも WOTS$^+$ が用いられているが，XMSS の WOTS$^+$ と以下の点で異なる．

- Winternitz パラメータは $w \in \{4, 16, 256\}$ である．

layer address	(32 bits)
tree address	(96 bits)
type = 0	(32 bits)
key pair address	(32 bits)
chain address	(32 bits)
hash address	(32 bits)

(a) WOTS$^+$ ハッシュアドレス

layer address	(32 bits)
tree address	(96 bits)
type = 1	(32 bits)
key pair address	(32 bits)
$00\cdots0$	(32 bits)
$00\cdots0$	(32 bits)

(b) WOTS$^+$ 公開鍵圧縮アドレス

layer address	(32 bits)
tree address	(96 bits)
type = 2	(32 bits)
$00\cdots0$	(32 bits)
tree height	(32 bits)
tree index	(32 bits)

(c) ハッシュ木アドレス

layer address	(32 bits)
tree address	(96 bits)
type = 3	(32 bits)
key pair address	(32 bits)
tree height	(32 bits)
treen index	(32 bits)

(d) FORS 木アドレス

layer address	(32 bits)
tree address	(96 bits)
type = 4	(32 bits)
key pair address	(32 bits)
$00\cdots0$	(32 bits)
$00\cdots0$	(32 bits)

(e) FORS 木根圧縮アドレス

layer address	(32 bits)
tree address	(96 bits)
type = 5	(32 bits)
key pair address	(32 bits)
chain address	(32 bits)
$00\cdots0$	(32 bits)

(f) WOTS$^+$ 鍵生成アドレス

layer address	(32 bits)
tree address	(96 bits)
type = 6	(32 bits)
key pair address	(32 bits)
$00\cdots0$	(32 bits)
tree index	(32 bits)

(g) FORS 鍵生成アドレス

図 6.5: アドレスの構造

- チェイニング関数は以下のように定義される．

$$\mathsf{chain}(X, i, s, \mathbf{PK}.\mathsf{seed}, \mathbf{ADRS}) := \begin{cases} X & s = 0 \text{ のとき} \\ \mathtt{NULL} & i + s \geq w \text{ のとき} \\ \mathbf{F}(\mathbf{PK}.\mathsf{seed}, \mathbf{ADRS}' \| [i+s-1]_{32}, \mathsf{chain}(X, i, s-1, \mathbf{PK}.\mathsf{seed}, \mathbf{ADRS})) & \text{それ以外のとき} \end{cases}$$

なお，**ADRS** は WOTS$^+$ ハッシュアドレスであり，$\mathbf{ADRS}' \| [i+s-1]_{32}$ は **ADRS** の hash address の値を $[i+s-1]_{32}$ とすることを表している．

■鍵生成アルゴリズム　入力は $\mathbf{SK}.\mathsf{seed}$, $\mathbf{PK}.\mathsf{seed}$, WOTS$^+$ ハッシュアドレス **ADRS** である．

1. $0 \leq i \leq \ell - 1$ について，**ADRS** の chain address の値を $[i]_{32}$，hash address の値を $[0]_{32}$ とし，

$$sk_i := \mathbf{PRF}(\mathbf{SK}.\mathsf{seed}, \mathbf{skADRS}) \qquad pk_i := \mathsf{chain}(sk_i, 0, w-1, \mathbf{PK}.\mathsf{seed}, \mathbf{ADRS})$$

とする．なお，**skADRS** は WOTS$^+$ 鍵生成アドレスであり，layer address, tree address, key pair address,

chain addresss については **ADRS** と同じ値が用いられる．

2. $pk := \mathbf{T}_\ell(\mathbf{PK}.\mathsf{seed}, \mathbf{wotspkADRS}, pk_0 \| \cdots \| pk_{\ell-1})$ とする．ここで，**wotspkADRS** は WOTS$^+$ 公開鍵圧縮アドレスであり，layer address, tree address, key pair address については **ADRS** と同じ値が用いられる．

公開鍵は pk である．秘密鍵は $sk := (sk_0, sk_1, \ldots, sk_{\ell-1})$ である．

■署名アルゴリズム　詳細は [1] を参照のこと．

■検証アルゴリズム　詳細は [1] を参照のこと．

6.3.3.2　HT

SPHINCS$^+$ では，ハイパー木 HT と呼ばれる XMSS^{MT} と同様の XMSS 木の階層構造が用いられる．HT が d 層の XMSS 木からなるとき，第 $(d-1)$ 層と第 0 層はそれぞれ，HT の根と葉に相当する．すべての XMSS 木の高さは等しく，Winternitz パラメータもすべて同じ値が用いられる．HT の各 XMSS 木の各葉は，6.3.3.1 節の WOTS$^+$ の公開鍵である．各 XMSS 木の葉以外の節点の計算にはハッシュ関数 **H** が用いられる．第 x 層の左から y 番目の XMSS 木の構成で使用される WOTS$^+$ ハッシュアドレス，WOTS$^+$ 公開鍵圧縮アドレス，WOTS$^+$ 鍵生成アドレス，ハッシュ木アドレスの layer address と tree address はそれぞれ $[x]_{32}$，$[y]_{96}$ である．

HT の鍵生成，署名，検証の各アルゴリズムについての詳細は [1] を参照のこと．

6.3.3.3　FORS (Forest of Random Subsets)

SPHINCS$^+$ では，メッセージの署名に WOTS$^+$ ではなく，FORS と呼ばれる方式が用いられる．FORS では一組の公開鍵と秘密鍵を用いて複数個のメッセージに署名できる．FORS は，数回（few-time）署名方式 HORS [38] に基づく HORST [5] の改良版である．FORS は $k, t := 2^a$ をパラメータとし，長さ ka bits の系列に署名を行う．

■鍵生成アルゴリズム　入力は $\mathbf{SK}.\mathsf{seed}$, $\mathbf{PK}.\mathsf{seed}$, FORS 木アドレス **ADRS** である．

1. $0 \le i < kt$ について，

$$sk_i := \mathbf{PRF}(\mathbf{SK}.\mathsf{seed}, \mathbf{skADRS}) \qquad \mathrm{Node}_{i,0} := \mathbf{F}(\mathbf{PK}.\mathsf{seed}, \mathbf{ADRS}, sk_i)$$

とする．ここで，**skADRS** は FORS 鍵生成アドレスであり，layer address, tree address, key pair address については **ADRS** と同じ値が用いられ，tree index の値は $[i]_{32}$ である．また，**ADRS** の tree index の値は $[i]_{32}$ である．

2. $1 \le j \le a$ について，それぞれ，$0 \le i < kt/2^j$ について，

$$\mathrm{Node}_{i,j} := \mathbf{H}(\mathbf{PK}.\mathsf{seed}, \mathbf{ADRS}, \mathrm{Node}_{2i,j-1} \| \mathrm{Node}_{2i+1,j-1})$$

とする．ここで，**ADRS** の tree height の値は $[j]_{32}$，tree index の値は $[i]_{32}$ である．

3. $pk := \mathbf{T}_k(\mathbf{PK}.\mathsf{seed}, \mathbf{forspkADRS}, \mathrm{Node}_{0,a} \| \cdots \| \mathrm{Node}_{k-1,a})$ とする．ここで，**forspkADRS** は FORS 木根圧縮アドレスであり，layer address, tree address, key pair address については **ADRS** と同じ値が用いられる．

このアルゴリズムにより，$\mathrm{Node}_{0,a}, \mathrm{Node}_{1,a}, \ldots, \mathrm{Node}_{k-1,a}$ を根とする k 個のマークル木が構成されている．公開鍵は pk である．秘密鍵は $sk_0, sk_1, \ldots, sk_{kt-1}$ である．

■署名アルゴリズム　長さ ka bits のメッセージ M をそれぞれ長さ a bits の k 個のブロック $M_0, M_1, \ldots, M_{k-1}$ に分割する．すなわち，$M = M_0 \| M_1 \| \cdots \| M_{k-1}$ である．さらに，M_i を 2 進数表記の非負整数とみなす．M の署名は $sk_{0\cdot t+M_0}, sk_{1\cdot t+M_1}, \ldots, sk_{(k-1)t+M_{k-1}}$ と，$0 \leq i < k$ について，$\mathrm{Node}_{i,a}$ を根とするマークル木の $\mathrm{Node}_{it+M_i,0}$ の認証パスである．

■検証アルゴリズム　詳細は [1] を参照のこと．

6.3.3.4　SPHINCS$^+$

前節までの構成要素を用いて SPHINCS$^+$ の署名が構成される．SPHINCS$^+$ のパラメータは以下のとおりである．

- セキュリティパラメータ n（単位は Byte）
- Winternitz パラメータ w
- ハイパー木の高さ h と階層数 d
- FORS の木の個数 k と各木の葉の個数 t

メッセージダイジェストの Byte 長は $m := \lfloor (k\log_2 t + 7)/8 \rfloor + \lfloor (h - h/d + 7)/8 \rfloor + \lfloor (h/d + 7)/8 \rfloor$ となる．

■鍵生成アルゴリズム　$\mathbf{SK}.\mathtt{seed}, \mathbf{SK}.\mathtt{prf} \in \mathbb{B}^n$ はいずれも無作為に選択される．$\mathbf{PK}.\mathtt{seed} \in \mathbb{B}^n$ は無作為に選択される．$\mathbf{PK}.\mathtt{root} \in \mathbb{B}^n$ は HT の第 $(d-1)$ 層の XMSS 木の根である．秘密鍵は $\mathbf{SK}.\mathtt{seed}, \mathbf{SK}.\mathtt{prf}, \mathbf{PK}.\mathtt{seed}, \mathbf{PK}.\mathtt{root}$ である．公開鍵は $\mathbf{PK}.\mathtt{seed}, \mathbf{PK}.\mathtt{root}$ である．

■署名アルゴリズム　メッセージ M の署名は以下のように生成される．

1. $\mathbf{R} := \mathbf{PRF_{msg}}(\mathbf{SK}.\mathtt{prf}, opt, M)$ とする．$opt = \mathbf{PK}.\mathtt{seed}$ であるが，opt を乱数とするオプションも用意されている．
2. $digest := \mathbf{H_{msg}}(\mathbf{R}, \mathbf{PK}.\mathtt{seed}, \mathbf{PK}.\mathtt{root}, M)$ とする．$digest$ の最初の $\lfloor (ka+7)/8 \rfloor$ Bytes，次の $\lfloor (h - h/d + 7)/8 \rfloor$ Bytes，その次の $\lfloor (h/d + 7)/8 \rfloor$ Bytes をそれぞれ tmp_0, tmp_1, tmp_2 とする．さらに，tmp_0 の先頭 ka bits を md，tmp_1 の先頭 $(h - h/d)$ bits を idx_{tree}，tmp_2 の先頭 h/d bits を idx_{leaf} とする．
3. HT の第 0 層の左から idx_{tree} 番目の XMSS 木の左から idx_{leaf} 番目の葉に対応する FORS の鍵を用いて md の署名を生成する．このとき，FORS の **ADRS** の layer address は $[0]_{32}$，tree address は idx_{tree}，key pair address は idx_{leaf} である．さらに，tree height, tree index はともに $[0]_{32}$ である．
4. 上の署名で用いられた FORS の公開鍵への HT による署名を生成する．

M の署名は $\mathbf{R}$，md への FORS による署名，md への署名の検証に用いられる FORS の公開鍵への HT による署名からなる．

■検証アルゴリズム　詳細については [1] を参照のこと．

SPHINCS$^+$ の秘密鍵，公開鍵，署名のサイズはそれぞれ，$4n$ Bytes，$2n$ Bytes，$(h + k(\log_2 t + 1) + d\ell + 1)n$ Bytes である．ここで，$\ell := \ell_1 + \ell_2$ であり，$\ell_1 := \lceil 8n/\log_2 w \rceil$, $\ell_2 := \lfloor (\log_2(\ell_1(w-1)))/\log_2 w \rfloor + 1$ である．

6.3.3.5　パラメータの設定と安全性

Hülsing と Kudinov [18] は，SPHINCS$^+$ が適応的選択メッセージ攻撃に対する存在偽造不能性（EUF-CMA）を満たすことを tweakable ハッシュ関数 $\mathbf{T}_\ell$，鍵付きハッシュ関数 $\mathbf{H_{msg}}$，擬似ランダム関数 $\mathbf{PRF}, \mathbf{PRF_{msg}}$ の以下の安

全性に帰着している．

- $\mathbf{T}_\ell$ が以下の性質を満たすこと
  - single-function, multi-target collision resistance (SM-TCR)
  - single-function, multi-target preimage resistance (SM-PRE)
  - single-function, multi-target decisional second preimage resistance (SM-DSPR)
  - single-function, multi-target undetectability (SM-UD)
- $\mathbf{H_{msg}}$ が interleaved target subset resilience (ITSR) を満たすこと
- $\mathbf{PRF}, \mathbf{PRF_{msg}}$ が擬似ランダム関数 (PRF) であること

ここで，SM-TCR, SM-DSPR, ITSR は，$\mathbf{T}_\ell, \mathbf{H_{msg}}$ の構成に用いられるハッシュ関数の第二原像攻撃に対する安全性に基づく性質であり，SM-PRE は原像攻撃に対する安全性に基づく性質である．一方，SM-UD, PRF は，秘密鍵入力を有するハッシュ関数が擬似ランダム関数であることを要求する．さらに，ビットマスクの生成については，ハッシュ関数がランダムオラクルであることが仮定される．

SPHINCS$^+$ については，表 6.8 のパラメータセットが示されている．この表の最左欄のラベルの s と f はそれぞれ，署名サイズ，計算時間について最適化されたパラメータセットであることを示している．ただし，一方の最適化で他方が非実用的にならないよう配慮されている．また，安全性レベルは NIST Post Quantum Cryptography Standardization Process の Call for Proposals に記された安全性強度のカテゴリである．なお，Haraka を用いる場合については，表 6.8 とは異なり，$n = 24, 32$ のときの安全性はレベル 2 とされており，したがって，bit 安全性はおよそ 128 程度となる．Bao らによる原像攻撃 [2] は安全性がレベル 2 であることを否定するものではない．

表 6.8: SPHINCS$^+$ のパラメータセットの例．署名長の単位は Byte である．

名称	n	h	d	$\log_2 t$	k	w	bit 安全性	安全性レベル	署名長
SPHINCS$^+$-128s	16	63	7	12	14	16	133	レベル 1	7,856
SPHINCS$^+$-128f	16	66	22	6	33	16	128	レベル 1	17,088
SPHINCS$^+$-192s	24	63	7	14	17	16	193	レベル 3	16,224
SPHINCS$^+$-192f	24	66	22	8	33	16	194	レベル 3	35,664
SPHINCS$^+$-256s	32	64	8	14	22	16	255	レベル 5	29,792
SPHINCS$^+$-256f	32	68	17	9	35	16	255	レベル 5	49,856

6.3.3.6 ハッシュ関数の実現法

SPHINCS$^+$ のハッシュ関数はすべて，SHAKE256，SHA-2，Haraka のうちのいずれかを用いて定義される．なお，tweakable ハッシュ関数については robust と simple の二つの実現が示されている．robust な実現では 6.2.4 節で述べられたビットマスクが用いられるが，simple な実現では用いられない．

SHA-2 を用いた実現では，当初は SHA-256 のみが用いられていたが，SHA-256 を用いた実現では安全性のレベル 5 が達成できないことを示す攻撃 [36] が示されたため，$n = 24, 32$ については，一部の関数が SHA-512 を用いて実現されることとなった．

SHAKE256 を用いた構成は以下のとおりである．

$$\mathbf{H_{msg}}(\mathbf{R}, \mathbf{PK}.\mathtt{seed}, \mathbf{PK}.\mathtt{root}, M) := \mathrm{SHAKE256}(\mathbf{R}\|\mathbf{PK}.\mathtt{seed}\|\mathbf{PK}.\mathtt{root}\|M, 8m)$$
$$\mathbf{PRF}(\mathbf{PK}.\mathtt{seed}, \mathbf{SK}.\mathtt{seed}, \mathbf{ADRS}) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|\mathbf{SK}.\mathtt{seed}, 8n)$$
$$\mathbf{PRF_{msg}}(\mathbf{SK}.\mathtt{prf}, \mathtt{OptRand}, M) := \mathrm{SHAKE256}(\mathbf{SK}.\mathtt{prf}\|\mathtt{OptRand}\|M, 8n)$$

robust な実現では

$$\mathbf{F}(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M_1) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M_1^{\oplus}, 8n)$$
$$\mathbf{H}(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M_1\|M_2) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M_1^{\oplus}\|M_2^{\oplus}, 8n)$$
$$\mathbf{T}_\ell(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M^{\oplus}, 8n)$$

simple な実現では

$$\mathbf{F}(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M_1) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M_1, 8n)$$
$$\mathbf{H}(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M_1\|M_2) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M_1\|M_2, 8n)$$
$$\mathbf{T}_\ell(\mathbf{PK}.\mathtt{seed}, \mathbf{ADRS}, M) := \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}\|M, 8n)$$

である．ここで，$M \in \{0,1\}^l$ のとき，$M^{\oplus} := M \oplus \mathrm{SHAKE256}(\mathbf{PK}.\mathtt{seed}\|\mathbf{ADRS}, l)$ である．

6.4 ハッシュ関数に基づく署名技術に関するまとめ

本章では，ハッシュ関数に基づく署名技術として，Lighton-Micali hash-based signatures, XMSS, SPHINCS$^+$ を取り上げた．これらはいずれも 6.2 節で述べた代表的なハッシュ関数に基づく署名方式に基づく構造を有する．Lighton-Micali hash-based signatures [29] と XMSS [17] は NIST の推奨アルゴリズムであり [8]，SPHINCS$^+$[1] は NIST Post Quantum Cryptography Standardization Process で標準化の候補アルゴリズムの一つに選出された．

ハッシュ関数に基づく署名技術の安全性はハッシュ関数の第二原像攻撃に対する安全性に依存しているが，XMSS, SPHINCS$^+$ については，秘密鍵入力を有するハッシュ関数が擬似ランダム関数であることにも依存する．さらに，ビットマスクの生成についてはハッシュ関数がランダムオラクルであることが仮定される．また，偽造攻撃の計算量は，ハッシュ関数がランダムオラクルであることを仮定して見積もられている．

ハッシュ関数に基づく署名技術については，stateful であること，すなわち，各メッセージの署名に用いられる 1 回署名の秘密鍵を 2 回以上使用することのないよう管理しなければならないことが問題であった．Lighton-Micali hash-based signatures と XMSS はいずれもハッシュ関数に基づく stateful な署名方式であり，それらを推奨アルゴリズムとする NIST SP 800-208 [8] には，ハッシュ関数に基づく stateful な署名方式は一般的な使用には適するものでなく，近い将来に実装が必要であり，その実装が長期間の使用を予定されており，かつ，使用開始後に他の署名方式への移行が実用的でないような応用での使用が意図されていると述べられている．

SPHINCS$^+$ は XMSS の設計で得られた知見に基づいて設計されており，HSS, XMSS^{MT} と同様の構造を有するが，各メッセージの署名に一つの秘密鍵で数回署名可能な FORS を用いることによって署名可能な回数を増加させることにより，stateless であることを達成している．

第 6 章の参照文献

[1] J.-P. Aumasson, D. J. Bernstein, W. Beullens, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, A. Hülsing et al. SPHINCS$^+$. Submission to the NIST post-quantum project, v.3.1, 2022. `https://sphincs.org/resources.html`.（2023-04-10 閲覧）

[2] Z. Bao, X. Dong, J. Guo, Z. Li, D. Shi, S. Sun, X. Wang. Automatic search of meet-in-the-middle preimage attacks on AES-like hashing. *EUROCRYPT 2021*, Part I, volume 12696 of LNCS, pp. 771–804, Springer, 2021.

[3] J. Buchmann, E. Dahmen, A. Hülsing. XMSS – A practical forward secure signature scheme based on minimal security assumptions. *PQCrypto 2011*, volume 7071 of LNCS, pp. 117–129, Springer, 2011.

[4] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche. Sponge functions. *ECRYPT Hash Workshop*, 2007-01.

[5] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, Z. Wilcox-O'Hearn. SPHINCS: Practical stateless hash-based signatures. *EUROCRYPT 2015*, Part I, volume 9056 of LNCS, pp. 368–397, Springer, 2015.

[6] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, P. Schwabe. The SPHINCS$^+$ signature framework. *IACR Cryptology ePrint Archive*, 2019/1086.

[7] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, P. Schwabe. The SPHINCS$^+$ signature framework. *ACM CCS 2019*, pp. 2129–2146, ACM, 2016.

[8] D. Cooper, D. Apon, Q. Dang, M. Davidson, M. Dworkin, C. Miller. Recommendation for stateful hash-based signature schemes. *NIST SP 800-208*, `https://csrc.nist.gov/publications/detail/sp/800-208/final`, 2020-10.（2023-04-10 閲覧）

[9] I. Damgård. A design principle for hash functions. *CRYPTO 1989*, volume 435 of LNCS, pp. 416–427, Springer, 1989.

[10] W. Diffie, M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, volume 22, issue 6, pp. 644–654, IEEE, 1976.

[11] E. Dahmen, K. Okeya, T. Takagi, C. Vuillaume. Digital signatures out of second-preimage resistant hash functions. *PQCrypto 2008*, volume 5299 of LNCS, pp. 109–123, Springer, 2008.

[12] J. Daemen, V. Rijmen. The Design of Rijndael: AES – The Advanced Encryption Standard. *Information Security and Cryptography*, Springer, 2002.

[13] C. Dods, N. P. Smart, M. Stam. Hash based digital signature schemes. *Cryptography and Coding: 10th IMA International Conference*, volume 3796 of LNCS, pp. 96–115, Springer, 2005.

[14] E. Eaton. Leighton-Micali hash-based signatures in the quantum random-oracle model. *SAC 2017*, volume 10719 of LNCS, pp. 263–280, Springer, 2017.

[15] S. Fluhrer. Further analysis of a proposed hash-based signature standard. *IACR Cryptology ePrint Archive*, 2017/553.

[16] L. K. Grover. A fast quantum mechanical algorithm for database search. *STOC '96*, pp. 212–219, ACM, 1996.

[17] A. Hülsing, D. Butin, S. Gazdag, J. Rijneveld, A. Mohaisen. XMSS: eXtended Merkle Signature Scheme. *IRTF RFC* 8391, `https://www.rfc-editor.org/rfc/rfc8391`, 2018-05. (2023-04-11 閲覧)

[18] A. Hülsing, M. Kudinov. Recovering the tight security proof of SPHINCS$^+$. *IACR Cryptology ePrint Archive*, 2022/346.

[19] A. Hülsing, L. Rausch, J. Buchmann. Optimal parameters for XMSS^{MT}. *CD-ARES 2013 Workshops: MoCrySEn and SeCIHD*, volume 8128 of LNCS, pp. 194–208, Springer, 2013.

[20] A. Hülsing, J. Rijneveld, F. Song. Mitigating multi-target attacks in hash-based signatures. *PKC 2016*, volume 9614 of LNCS, pp. 387–416, Springer, 2016.

[21] A. Hülsing. W-OTS$^+$ – shorter signatures for hash-based signature schemes. *AFRICACRYPT 2013*, volume 7918 of LNCS, pp. 173–188, Springer, 2013.

[22] J. Katz. Analysis of a proposed hash-based signature standard. *Security Standardisation Research (SSR 2016)*, volume 10074 of LNCS, pp. 261–273, Springer, 2016.

[23] P. Kampanakis, S. Fluhrer. LMS vs XMSS: Comparion of two hash-based signature standards. *IACR Cryptology ePrint Archive*, 2017/349.

[24] S. Kölbl, M. M. Lauridsen, F. Mendel, C. Rechberger. Haraka v2 – efficient short-input hashing for post-quantum applications. *IACR ToSC*, 2016, issue 2, pp. 1–29, Ruhr-Universität Bochum.

[25] J. Kelsey, B. Schneier. Second preimages on n-bit hash functions for much less than 2^n work. *EUROCRYPT 2005*, volume 3494 of LNCS, pp. 474–490, Springer, 2005.

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

[27] F. T. Leighton, S. Micali. Large provably fast and secure digital signature schemes based on secure hash functions. US Patent 5,432,852, 1995.

[28] P. Lafrance, A. Menezes. On the security of the WOTS-PRF signature scheme. *Advances in Mathematics of Communications*, volume 13, number 1, pp.185–193, American Institute of Mathematical Sciences, 2019.

[29] D. A. McGrew, M. Curcio, S. R. Fluhrer. Leighton-Micali hash-based signatures. *IRTF RFC 8554*, `https://datatracker.ietf.org/doc/html/rfc8554`, 2019-04. (2023-04-10 閲覧)

[30] R. C. Merkle. Secrecy, Authentication, and Public Key Systems. PhD thesis, Stanford University, 1979. `https://www.merkle.com/papers/Thesis1979.pdf`.

[31] R. C. Merkle. A certified digital signature. *CRYPTO 1989*, volume 435 of LNCS, pp. 218–238, Springer, 1989.

[32] R. C. Merkle. One way hash functions and DES. *CRYPTO 1989*, volume 435 of LNCS, pp. 428–446, Springer, 1989.

[33] NIST. Advanced encryption standard (AES). *FIPS PUB 197*, `https://csrc.nist.gov/publications/detail/fips/197/final`, 2001-11. (2023-04-10 閲覧)

[34] NIST. Secure hash standard (SHS). *FIPS PUB 180-4*, `https://csrc.nist.gov/publications/detail/fips/180/4/final`, 2015-08. (2023-04-10 閲覧)

[35] NIST. SHA-3 standard: Permutation-based hash and extendable-output functions, *FIPS PUB 202*, `https://csrc.nist.gov/publications/detail/fips/202/final`, 2015-08. （2023-04-10 閲覧）

[36] R. A. Perlner, J. Kelsey, D. A. Cooper. Breaking category five SPHINCS$^+$ with SHA-256. *PQCrypto 2022*, volume 13512 of LNCS, pp. 501–522, Springer, 2022.

[37] J. Quisquater, M. Girault. $2n$-bit hash-functions using n-bit symmetric block cipher algorithms. *EUROCRYPT 1989*, volume 434 of LNCS, pp. 102–109, Springer, 1989.

[38] L. Reyzin, N. Reyzin. Better than BiBa: Short one-time signatures with fast signing and verifying. *ACISP 2002*, volume 2384 of LNCS, pp. 144–153, Springer, 2002.

耐量子計算機暗号の研究動向調査報告書
[CRYPTREC TR-2001-2022]

発行日：2023 年 3 月 31 日（第 1 版）
発行者
- 〒184-8795
  東京都小金井市貫井北町四丁目２番１号
  国立研究開発法人情報通信研究機構
  （サイバーセキュリティ研究所　セキュリティ基盤研究室）
  NATIONAL INSTITUTE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
  4-2-1 NUKUI-KITAMACHI, KOGANEI
  TOKYO, 184-8795 JAPAN
- 〒113-6591
  東京都文京区本駒込二丁目２８番８号
  独立行政法人情報処理推進機構
  （セキュリティセンター セキュリティ技術評価部 暗号グループ）
  INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN
  2-28-8 HONKOMAGOME, BUNKYO-KU
  TOKYO, 113-6591 JAPAN