

CRYPTREC

暗号技術ガイドライン（耐量子計算機暗号）

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	1
1.1	耐量子計算機暗号 (PQC) の必要性について	2
1.2	PQC の研究及び標準化等に関する動向	4
1.3	本調査で対象とした PQC の種類	5
1.4	暗号技術ガイドライン（耐量子計算機暗号）執筆者リスト	6
第 1 章の参考文献		7
第 2 章	PQC の活用方法	11
2.1	暗号の利用形態	11
2.1.1	署名用途での公開鍵暗号の利用	11
2.1.2	守秘用途での公開鍵暗号の利用	12
2.1.3	鍵共有用途での公開鍵暗号の利用	12
2.2	各利用形態における課題	12
2.2.1	署名用途での課題	13
2.2.2	守秘用途での課題	14
2.2.3	鍵共有用途での課題	14
2.3	各利用形態における対策	15
2.3.1	署名用途固有の対策	16
2.3.2	秘匿及び鍵共有用途固有の対策	16
2.3.3	耐量子計算機暗号の活用方法	16
第 2 章の参考文献		18
第 3 章	格子に基づく暗号技術	21
3.1	格子に基づく暗号技術の安全性の根拠となる問題	21
3.1.1	LWE 問題の紹介	21
3.1.2	NTRU 問題の紹介	21
3.1.3	格子問題の公開チャレンジの求解状況	22
3.2	格子に基づく代表的な暗号方式	22
3.2.1	Hash-and-Sign に基づく署名方式の格子問題への拡張	22
3.2.2	Fiat-Shamir 署名方式の格子問題への拡張	23
3.3	格子に基づく主要な暗号方式	24

3.3.1	CRYSTALS-Kyber	25
3.3.2	CRYSTALS-Dilithium	28
3.3.3	FALCON	33
3.4	格子に基づく暗号技術に関するまとめ	37
第 3 章の参考文献		39
第 4 章 符号に基づく暗号技術		47
4.1	符号に基づく暗号技術の安全性の根拠となる問題	48
4.1.1	LPN 問題とは	48
4.1.2	LPN 問題の拡張	49
4.1.2.1	復号問題	49
4.1.2.2	シンδροーム復号問題	49
4.1.2.3	Module-LPN 問題	49
4.1.3	LPN 問題に対する評価	50
4.1.3.1	BKW アルゴリズムおよびその改良	50
4.1.3.2	Arora-Ge アルゴリズム	50
4.1.3.3	SD 問題を経由するアルゴリズム	51
4.1.3.4	量子アルゴリズムへの耐性	51
4.1.3.5	現状の進展	52
4.2	符号に基づく代表的な暗号方式	53
4.2.1	McEliece 暗号	53
4.2.2	Niederreiter 暗号	54
4.3	符号に基づく主要な暗号方式	54
4.3.1	Classic McEliece	55
4.3.2	BIKE	56
4.3.3	暗号方式 3: HQC	57
4.4	まとめ	58
第 4 章の参考文献		60
第 5 章 多変数多項式に基づく暗号技術		63
5.1	多変数多項式に基づく暗号技術の安全性の根拠となる問題	63
5.1.1	MP 問題 (MQ 問題)	63
5.1.2	MinRank 問題	64
5.1.3	IP 問題, EIP 問題	65
5.2	多変数多項式に基づく代表的な暗号方式	65
5.2.1	双極型システム	65
5.2.2	HFE 方式, HFEv-方式	67
5.2.2.1	暗号方式 HFE	67
5.2.2.2	署名方式 HFEv-	67

5.2.3	署名方式 Rainbow	68
5.3	多変数多項式に基づく主要な暗号方式	69
5.3.1	署名方式 UOV	69
5.3.1.1	UOV の概要	69
5.3.1.2	UOV のパラメータ選択	70
5.4	多変数多項式に基づく暗号技術に関するまとめ	70
第 5 章の参考文献		72
第 6 章 同種写像に基づく暗号技術		75
6.1	同種写像に基づく暗号技術の安全性の根拠となる問題	75
6.1.1	同種写像問題の一般形	75
6.1.2	自己準同型環計算問題と SQISign 署名方式の安全性に関する計算問題	77
6.1.2.1	自己準同型環計算問題	77
6.1.2.2	SQISign 署名の安全性に関する計算問題	78
6.2	同種写像に基づく代表的な暗号方式	79
6.2.1	GPS 署名	79
6.3	同種写像に基づく主要な暗号方式	80
6.3.1	SQISign 署名	81
6.4	同種写像に基づく暗号技術に関するまとめ	83
第 6 章の参考文献		85
第 7 章 ハッシュ関数に基づく署名技術		87
7.1	ハッシュ関数に基づく署名技術の安全性の根拠となる問題	87
7.2	ハッシュ関数に基づく代表的な署名方式	88
7.2.1	Winternitz One-Time Signature	88
7.2.2	マークル木を用いた署名方式	88
7.2.3	マークル木の階層構造による署名方式	89
7.2.4	プレフィクスとビットマスク	89
7.3	ハッシュ関数に基づく主要な署名方式	90
7.3.1	XMSS: eXtended Merkle Signature Scheme	91
7.3.1.1	WOTS ⁺	91
7.3.1.2	XMSS	93
7.3.1.3	XMSS ^{MT}	94
7.3.1.4	パラメータの設定と安全性	94
7.3.2	SPHINCS ⁺	95
7.3.2.1	WOTS ⁺	95
7.3.2.2	HT	97
7.3.2.3	FORS (Forest of Random Subsets)	97
7.3.2.4	SPHINCS ⁺	98

7.3.2.5	パラメータの設定と安全性	98
7.3.2.6	ハッシュ関数の実現法	99
7.4	ハッシュ関数に基づく署名技術に関するまとめ	100
第 7 章の参考文献		101

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、本ガイドライン及び調査報告書 [11] 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。

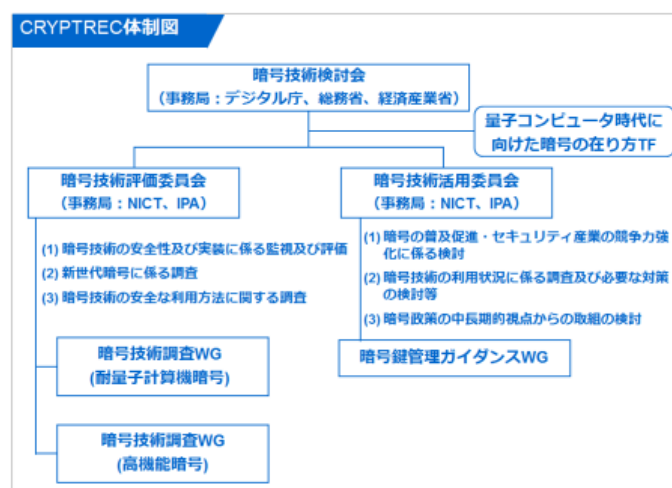


図 1.1: 2022 年度 CRYPTREC 体制図

^{*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 の祖冲之 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, 11, 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 [11] 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」[4] 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ル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 $376289=571 \times 659$ 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いが、[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に関わりがある．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に関わりがある．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 暗号技術ガイドライン（耐量子計算機暗号）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 章の参考文献

- [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) . 耐量子計算機暗号の研究動向調査報告書. CRYPTREC TR-2001-2022, <https://www.cryptrec.go.jp/report/cryptrec-tr-2001-2022.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, 1050403, American Physical Society, 2021.
- [20] IBM. IBM、400 量子ビット超えの量子プロセッサと次世代 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 standardiza-

- tion 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

PQC の活用方法

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

2.1 暗号の利用形態

公開鍵暗号には幾つかの利用形態が存在するが、本書では「電子政府における調達のために参照すべき暗号のリスト」（以下、CRYPTREC 暗号リストと呼ぶ）に合わせ、公開鍵暗号を署名・守秘・鍵共有に分類し、以降その分類に沿って記載する。

2.1.1 署名用途での公開鍵暗号の利用

本節では、署名を付与する行為を「デジタル署名処理」と呼び、付与される署名データを「デジタル署名」と呼ぶ。デジタル署名が付与されたコンテンツを改竄すると、その改竄を確認することができる。そのため、署名用途の公開鍵暗号を用い、コンテンツにデジタル署名を付与することで、コンテンツの改竄を防止することができる。コンテンツはドキュメントデータであることもあれば、鍵情報^{*1}を含むこともある。また、デジタル署名処理に用いられる秘密鍵が、（対応する公開鍵を含む電子証明書等により）所定の人物等と紐づいている場合においては、コンテンツの生成者等を確認（認証）することもできる。このように署名用途の公開鍵暗号は、コンテンツの改竄防止、署名者等の認証、データ元の認証等に利用される。

具体的な署名用途の公開鍵暗号の利用例としては、TLS 通信 [15] におけるクライアント認証やサーバ認証、OS のコードサイン等に広く利用されている。また、公開鍵の配布手段の 1 種である公開鍵暗号基盤（PKI）においても、電子証明書 [5] 等の形態で広く利用されている。加えて、タイムスタンプ署名 [1] では、コンテンツに対して署名が付与された時刻を確認するために利用されている。CRYPTREC 暗号リストには、DSA、ECDSA、RSA-PSS、RSASSA-PKCS1-v1.5 及び EdDSA が署名用途の公開鍵暗号として記載されている。

^{*1} 鍵情報には暗号鍵やメタデータが含まれ [8]、公開鍵暗号の鍵のみではなく共通鍵暗号の鍵に関する情報も含む概念となる。

2.1.2 守秘用途での公開鍵暗号の利用

守秘用途の公開鍵暗号によって暗号化された暗号文は、対応する秘密鍵なしに解読することは困難となる。そのため、守秘用途の公開鍵暗号は、意図した相手にデータを提示する為に利用できる。暗号化処理による保護は、ドキュメントデータ等に対して行われることもあれば、別の暗号鍵の鍵情報^{*2}に対して行われることもある。保護が鍵情報に対して行われる具体的なユースケースとしては、鍵情報を通信当事者間で共有する場合や、暗号鍵所有者がその鍵情報をバックアップする場合等が該当する。

守秘用途及び鍵共有用途の公開鍵暗号の一般的な実装形態として、公開鍵暗号により共通鍵暗号の共通鍵を保護し、またその共通鍵を利用した共通鍵暗号によりコンテンツの秘匿性や完全性を保護するというアプローチが存在する。

守秘用途の公開鍵暗号をこのアプローチで用いる場合においては、共通鍵暗号の共通鍵は送信者により作成され、配送される。このアプローチにおいて仮に、ある時点で秘密鍵が漏洩した場合、過去にその秘密鍵を持つ利用者に対して配送された共通鍵暗号の共通鍵が漏洩するおそれがある。また、受信者は共通鍵の生成に関わることができず、仮に送信者が別の通信相手との共通鍵暗号の共通鍵を使い回し等していても察知することができない。そのため、昨今の TLS 通信等における共通鍵の共有においては、秘匿用途の公開鍵暗号でなく、(次節の) 鍵共有用途での公開鍵暗号を一時的 (Ephemeral) な鍵を用いて利用することが望ましいと考えられるようになった [8, 18]。なお、「TLS 暗号設定ガイドライン」[8]においても、鍵交換 (鍵共有・守秘) においては、Perfect Forward Security (PFS) の特性をもつ DHE (または ECDHE) を選択するのがセキュリティ上望ましいと記載されている。CRYPTREC 暗号リストには、RSA-OAEP 及び RSAES-PKCS1-v1.5^{*3}が守秘用途の公開鍵暗号として記載されている。また、RFC7525 においても [18]、4.1 節において (守秘用途である) RSA key transport は利用すべきでないと記載されており、4.2 節でも一時的 (Ephemeral) な鍵を用いる暗号スイートが推奨されている。

2.1.3 鍵共有用途での公開鍵暗号の利用

鍵共有用途での公開鍵暗号は、鍵共有に参加する二者が、同一の鍵情報^{*4}を共有するために利用される。守秘用途の公開鍵暗号を二者間の通信で利用する場合は、任意のコンテンツに対する暗号処理を実施できる。一方、鍵共有用途の公開鍵暗号を二者間の通信で利用する場合、そこで共有されるものは、(共通鍵暗号や公開鍵暗号等の) 何らかの暗号アルゴリズムで利用するための暗号鍵となる。

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

2.2 各利用形態における課題

Michel Mosca [11] によると、量子コンピュータによる攻撃の脅威は、保護対象の暗号の保護期間を X 年、暗号処理の実装の置き換えに要する期間を Y 年、暗号解読に利用可能な量計算機が実現するまでの期間を Z 年としたときに、

^{*2} 秘密鍵、共通鍵、鍵導出鍵、及びそれら鍵のメタデータを含む。

^{*3} 守秘用途の RSAES-PKCS1-v1.5 は、運用監視暗号リストに記載されており、互換性維持以外での利用は推奨されていない。

^{*4} 共通鍵暗号の共通鍵、鍵導出機能の鍵やパラメータ等

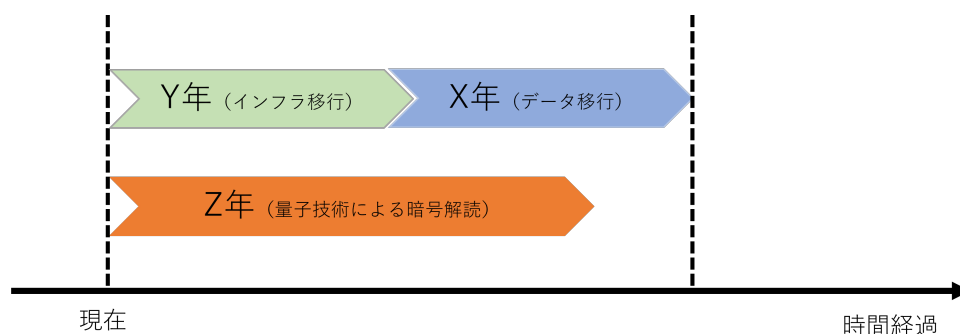


図 2.1: Mosca の発表 [11] より

これらが $X + Y > Z$ である場合には、心配する必要があるとしている（図 2.1）。もっとも、暗号解読が可能な量子コンピュータの実現時期は未だ不透明であり、 Z を予想するのは難しい。また X は、暗号のみならず、保護対象となるデータの性質等によっても大きく異なる。保護対象となるデータに対して、保護期間が設定されていない場合などは、 X を導出することが難しいこともありうる。加えて Y は、暗号の実装形態により大きく変化する。また、 X 及び Y は、暗号システムの運用を通して、将来において変動することもありうる。

このように、ある公開鍵暗号アプリケーションが与えられた時に、その公開鍵暗号アプリケーションが量子コンピュータの脅威について備える必要があるかを判断しようとした場合において、 Z は不確定であり、 X や Y も将来においては変動しうるため、判断が難しいという課題がある。ここで、保護対象となるデータに保護期間が設定されていない場合においては、判断に先駆けて（ X 導出のために）データの保護期間を決定する必要があり、場合によっては判断に必要な情報を収集するために相当の期間を要する可能性もある。

なお、多様な暗号ライブラリが、耐量子計算機暗号を相互運用可能な形態で受け入れるためには、共通の OID (Object Identifier) 体系を利用することが望ましい。この OID 体系については、IETF にて標準化が進んでおり [10, 17]、近い将来に解決されることが見込まれているため、以降は議論しない。

2.2.1 署名用途での課題

署名用途の公開鍵暗号は、コンテンツの改竄防止、認証等に利用されるが、ユースケースによって脅威の性質は大きく異なる。例えば、TLS 通信 [15] におけるクライアント認証やサーバ認証においては、認証用に付与されたデジタル署名の検証を行うのはその場限りとなり、それほど長期間に渡り検証されるわけではない。一方で、電子データに対するドキュメント署名や、バイナリデータに対するコードサインであれば、署名対象のデータを利用する人が存在する限り（数十年に渡り）デジタル署名が検証されることもありうる。特に、後者においては、仮に電子証明書に有効期間等が記載されていたとしても、その有効期間満了後にも検証されることが十分に考えられる。このように、デジタル署名用途において、図 2.1 における X は大きく異なり、個別のアプリケーションごとに判断する必要がある。また、公開鍵の配布のために PKI 等を利用していた場合、トラストアンカーの置き換え等に時間を要することになり、図 2.1 における Y が 10 年以上となることも珍しくない。なお、e-mail を保護する S/MIME プロトコルにおいては、S/MIME 証明書（に対応する秘密鍵）によって保護される対象を通信プロトコルと捉えると（data in transit に対するセキュリティとなり）TLS 同様保護の対象となる期間は短い、S/MIME 証明書（に対応する秘密鍵）によって保護される対象をメールコンテンツと捉えると（data at rest に対するセキュリティとなり）保護の対象期間が非常に長くなりうる。このように、 X を導出する上では、個別のデータの保護期間のみでなく、プロトコルがサポートする機能も明確化する

ことが望ましい。

2.2.2 守秘用途での課題

守秘用途の公開鍵暗号では、保護対象となるコンテンツや鍵情報の保護期間が非常に長いことを想定していることや、場合によっては無期限で保護されることを想定していることがある。具体的には、患者を特定または類推可能な形態で保管された遺伝性の疾病に関する情報や、外交上重要な情報、及びそれらの情報の暗号化に利用される鍵などは、長い保護期間を持つ傾向にある。また、ドキュメントの生成時には無期限に秘匿することを想定しており、公開することを想定していない情報等も存在する。このような場合においては、図 2.1 における X は非常に大きくなるため、 $X + Y > Z$ が成立すると仮定した上で、速やかに量子コンピュータによる攻撃に対する何らかの対策を行うことが望ましい。

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

2.2.3 鍵共有用途での課題

鍵共有用途での課題は、守秘用途の課題と同種の課題を含む。例えば、鍵共有で共有された共通鍵暗号の共通鍵が、非常に長い保管期間を持つデータの暗号化に利用されていた場合、図 2.1 における X は非常に大きくなり、 $X + Y > Z$ が成立すると想定され、速やかに量子コンピュータによる攻撃に対する何らかの対策が必要となる。

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

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

標準化された鍵共有用途の公開鍵暗号が存在するのであれば、1) を選択可能であり、それは比較的容易に実現可能だと考えられる。しかしながら、標準化された鍵共有用途の公開鍵暗号が存在しなく、守秘用途の公開鍵暗号しか標準化されたものがないのであれば、2) を選択することとなり、守秘用途の耐量子計算機暗号を用いて鍵共有部分を構成することとなる。ここで、守秘用途の暗号を単純に導入した場合、PFS の性質を持たなくなる可能性があり、それによりデータ保護及び運用ポリシー策定時に想定していなかった経路からの情報漏洩等が発生する可能性が生じる。また、守秘用途の公開鍵暗号に対して何らかの手を加えて、PFS の性質を持つ暗号プロトコルを構成したとしても、その暗号プロトコルが標準化されていない場合は、運用ポリシー上、利用できないこともあり得る。このような課題は、既存のシステムが鍵共有用途の公開鍵暗号を利用しており、今後採用する耐量子計算機暗号が守秘用途の場合に発生する課題となる。

2.3 各利用形態における対策

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

例えば、NIST は耐量子計算機暗号への移行の一環として情報システムが利用している暗号の棚卸しや整理の方法を検討しており [3]、以下のような流れで対策を行うことを想定していると考えられる。

- ある程度以上の価値のある情報を扱う情報システムをリストアップする。
- 情報システム内において利用される、暗号モジュール及び暗号方式をリストアップする。
- 暗号方式によって保護される情報の保護期間 (X の把握のために必要) を把握する。保護期間が設定されていない場合は設定する。
- 暗号モジュールの移行に要する時間 (Y) を把握する。
- X 又は Y が一定の値以上で、またある程度以上の価値のある情報を扱う情報システムをリストアップする。
- リストに対してプライオリティ付けを行う。
- プライオリティ順に、Cryptographic agility [7] の向上や暗号方式の耐量子計算機暗号への移行を実施する。
- 耐量子計算機暗号への速やかな移行が難しい場合は、以下の公開鍵暗号の用途に応じた対策を行う。

暗号モジュールや暗号方式のリストアップは、既に管理簿や仕様書等が存在するのであればそれを利用できる。管理簿や詳細な仕様書等が存在しない場合は、何らかの自動化ツールを使うことが、効率の面からもミスを減らす面からも望ましい。そのようなツールの利用を検討する上では、(2022 年末時点では最終報告は公開されていないものの、今後出てくるであろう) NIST NCCoE の検討 [3] が参考になるものと思われる。

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

暗号の移行に際しては、速やかに耐量子計算機暗号に移行するというアプローチと、予め Cryptographic agility [7]^{*5} を向上させつつ、ある程度以上の Cryptographic agility が達成された後に移行するというアプローチが存在する。Cryptographic agility が上昇した場合、 Y が減少することとなる。そのため、例えば耐量子計算機暗号の評価が十分にされておらず、暗号移行開始の妨げとなっているケースにおいては、当面の間は Cryptographic agility 上昇に努めるというアプローチが合理的であると考えられる [9]。なお、Cryptographic agility 上昇に伴う Y が減少により、該当する情報システムのプライオリティが下がれば、他の (相対的にプライオリティが高くなった) 情報システムへリソースを集中することもできる。

^{*5} より迅速に暗号を変更可能とする性質。

2.3.1 署名用途固有の対策

署名用途の公開鍵暗号は様々なユースケースで利用されるが、PKI 等のインフラの移行に要する時間 (Y) やコードサイン証明書が利用される期間 (X) が比較的長く、速やかな耐量子計算機暗号への移行が難しい可能性もある。このようなケースにおいても、以下のような対応を取ることが望ましい。

PKI においては一般に Y が大きくなる傾向にあるが、電子証明書の有効期間短縮や、1 枚の電子証明書に対して従来暗号と耐量子計算機暗号の 2 つの公開鍵及びデジタル署名を付与する方式などを採用することで、ある程度は Y も減少することが期待できる [16, 14]。なお、後者の 2 つの公開鍵暗号とデジタル署名を利用する方式においては、実装やポリシー管理の複雑性が大きく増加することが考えられ、その点には注意を払う必要がある。

X を実質的に短縮する技術として、タイムスタンプ更新技術が挙げられる。例えば、ERS[6] 等を利用することで、タイムスタンプの更新や、暗号の更新を行うことができる。 Z 年が経過する前に、既存の公開鍵暗号を耐量子計算機暗号に更新することが可能であれば、 X, Y, Z の関係性によらず、データは保護される。一方で、実装の複雑性が増加する可能性があり、 Y が上昇する可能性がある点は注意が必要となる。

2.3.2 秘匿及び鍵共有用途固有の対策

根本的な対策は、暗号を耐量子計算機暗号に移行することである。しかしながら、2.2.2 節及び 2.2.3 節に記載した通り、秘匿及び鍵共有用途で保護されたコンテンツや鍵情報は、保護期間が非常に長いことや、場合によっては無期限で保護されることを想定していることもある。そのような情報は既に将来における量子コンピュータでの解読リスクに晒されており、一刻も早く対応を始めない限り、リスクに晒された情報は増加し続ける。一方で、全ての秘匿及び鍵共有用途の公開鍵暗号を移行するためには非常に大きなリソースが要求され、全ての暗号文に対して迅速な移行を行うことには困難が想定される。

そのような状況においても、秘匿及び鍵共有用途固有の対策を効率的に行う方法 [9] として、以下のようなアプローチが考えられる。 Z に対して $X + Y$ が非常に小さい ($X + Y \ll Z$) と予測される暗号文に対しては、CRYPTREC による注意喚起情報 [4] に注意を払いつつ、今まで通りの対応を行う。また、 $X + Y > Z$ となることが十分予想される暗号文に対しては、2.3 節前段で述べたような、耐量子計算機暗号への移行や、暗号文の保護期間である X 年の短縮と、情報システムの暗号処理の実装の置き換えに要する期間 Y 年の短縮を行う（その結果 X や Y が十分に小さくすることができるのであれば、今まで通りの対応を行う）。一方で、前述検討の上でも $X + Y > Z$ と予想される又はそうなることが避けられない暗号文に対しては、暗号システムの対量子計算機暗号への移行を進めつつも、既存の公開鍵暗号によって保護されている暗号文は公開ネットワーク等に保管せず、適切にアクセスコントロールを行う。なお、現在 DH を利用している場合は 2.2.3 節で述べたような検討を行い、DH 固有の性質が必要か否かを予め検討することが望ましい。

2.3.3 耐量子計算機暗号の活用方法

耐量子計算機暗号を活用する上では、2.3 節冒頭で述べたとおり、現状において、どのようなデータに対して、どのような暗号技術を利用しているかを把握することが第一歩となる。また、保護対象となるデータの保護期間を予め把握しておくことで、より効率的な対応ができると考えられる [9]。その上で、公開できるデータは公開し、破棄可能なデータは破棄することが望ましい。上述のような検討を予め行うことで、Cryptographic agility[7] の向上も見込むことができ、より効果的に耐量子計算機暗号を活用できることが期待できる。量子コンピュータの脅威への対策を検討するに

あたっては保護されている情報の価値と，図 2.1 における X, Y, Z の関係を踏まえてプライオリティを付けて，そのプライオリティ順に対策を実施することが望ましい。

第 2 章の参考文献

- [1] C. Adams, P. Cain, D. Pinkas, R. Zuccherato. Internet X.509 public key infrastructure time-stamp protocol (TSP). <https://www.rfc-editor.org/rfc/rfc3161>, 2001-08. (2023-04-12 閲覧)
- [2] A. Becker, R. Guthrie, M. Jenkins. Non-composite hybrid authentication in PKIX and applications to internet protocols. <https://datatracker.ietf.org/doc/draft-becker-guthrie-noncomposite-hybrid-auth/>. (2023-04-12 閲覧)
- [3] W. Barker, M. Souppaya, W. Newhouse. Migration to post-quantum cryptography. <https://www.nccoe.nist.gov/sites/default/files/legacy-files/pqc-migration-project-description-final.pdf>, 2021-08. (2023-04-12 閲覧)
- [4] CRYPTREC. 注意喚起一覧. <https://www.cryptrec.go.jp/er.html>. (2023-04-12 閲覧)
- [5] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. <https://www.rfc-editor.org/rfc/rfc5280>, 2008-05. (2023-04-12 閲覧)
- [6] T. Gondrom, R. Brandner, U. Pordes. Evidence record syntax (ERS). <https://www.rfc-editor.org/rfc/rfc4998>, 2007-08. (2023-04-12 閲覧)
- [7] R. Housley. Guidelines for cryptographic algorithm agility and selecting mandatory-to-implement algorithms. <https://www.rfc-editor.org/rfc/rfc7696>, 2015-11. (2023-04-12 閲覧)
- [8] CRYPTREC. TLS 暗号設定ガイドライン. <https://www.cryptrec.go.jp/report/cryptrec-gl-3001-3.0.1.pdf>. (2023-04-12 閲覧)
- [9] 伊藤 忠彦, 宇根 正志, 清藤 武暢. 量子コンピュータによる脅威を見据えた 暗号の移行対応. <https://www.imes.boj.or.jp/research/papers/japanese/19-J-15.pdf>, 2019-08. (2023-04-12 閲覧)
- [10] J. Massimo, P. Kampanakis, S. Turner, B. Westernbaan. Algorithms and identifiers for post-quantum algorithms. <https://datatracker.ietf.org/doc/draft-massimo-lamps-pq-sig-certificates>, 2022-07-08. (2023-04-12 閲覧)
- [11] 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)
- [12] 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)
- [13] National Security Agency. The commercial national security algorithm suite 2.0 and quantum computing

- FAQ. https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI_CNCA_2.0_FAQ_.PDF, 2022-09-07. (2023-04-12 閲覧)
- [14] M. Ounsworth, J. Gray. Composite KEM for use in internet PKI. <https://datatracker.ietf.org/doc/html/draft-ounsworth-pq-composite-kem-00.html>, 2022-07-11. (2023-04-12 閲覧)
- [15] E. Rescorla. The transport layer security (TLS) protocol version 1.3. <https://www.rfc-editor.org/rfc/rfc8446>, 2018-08. (2023-04-12 閲覧)
- [16] D. Stebila, S. Fluhrer, S. Gueron. Hybrid key eXchange in TLS 1.3. <https://datatracker.ietf.org/doc/html/draft-ietf-tls-hybrid-design>, 2023-02-27. (2023-04-12 閲覧)
- [17] S. Turner, P. Kampanakis, J. Massimo, B. Westernbaan. Algorithm identifiers for NIST's PQC algorithms for use in the internet X.509 public key infrastructure. <https://datatracker.ietf.org/doc/html/draft-turner-lamps-nist-pqc-kem-certificates>, 2022-03-07. (2023-04-12 閲覧)
- [18] Y. Sheffer, R. Holz, P. Saint-Andre. Recommendations for secure use of transport layer security (TLS) and datagram transport layer security (DTLS). <https://www.rfc-editor.org/rfc/rfc7525>, 2015-05. (2023-04-12 閲覧)

第 3 章

格子に基づく暗号技術

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

3.1 格子に基づく暗号技術の安全性の根拠となる問題

3.1.1 LWE 問題の紹介

LWE 問題は機械学習理論から派生した求解困難な問題で, 整数剰余環 $\mathbb{Z}_q$ 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

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

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。

3.1.2 NTRU 問題の紹介

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

定義 3.1 (NTRU 問題 [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.

3.1.3 格子問題の公開チャレンジの求解状況

SVP や LWE に対する求解アルゴリズムをテストする目的で, ドイツ・ダルムシュタット大学によって「SVP チャレンジ」・「LWE チャレンジ」と呼ばれる求解コンテストがインターネット上で開催されている [101]. 2018 年に, ^{ふるい}篩をベースとした高速な格子アルゴリズム群である General Sieve Kernel (G6K)[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)

3.2 格子に基づく代表的な暗号方式

3.2.1 Hash-and-Sign に基づく署名方式の格子問題への拡張

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

- M : メッセージ
- $h = \text{hash}(M)$: 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 $f(x) = a^x \bmod p$ 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, $SA = 0 \pmod q$ 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. $tA = u \pmod q$ 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, $(\sigma - u)A \pmod q = vA \pmod q = 0$ が成り立つ. また 秘密鍵 S の特徴から, $\sigma \in D_{\mathbb{Z}^m, s}$ であることから, σ は短いベクトルとなる. 本署名方式は LWE 仮定の元で SUF-CMA (Strong Existential Unforgeability under Chosen Message Attack) 安全であることが示されている.

3.2.2 Fiat-Shamir 署名方式の格子問題への拡張

Fiat, Shamir らによって提示された Fiat-Shamir 変換 [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 $g(x) = g^x \pmod n$ 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。 $h_{\hat{a}}(\hat{z}) = \hat{a} \cdot \hat{z} = a_1 z_1 + \dots + a_m z_m \in R$ 。

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, $c = \text{hash}(h_{\hat{a}}(\hat{y}) \| M)$, $\hat{z} = \hat{y} + c\hat{s}$ 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 $c = \text{hash}(h_{\hat{a}}(\hat{z}) - Sc, M)$ が成り立つ場合に署名を受理する。

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

3.3 格子に基づく主要な暗号方式

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

格子を用いた主な公開鍵暗号の構成として, 最初期の Ajtai-Dwork 型 [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*1。

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

表 3.1: 格子に基づく暗号の分類

文献	暗号化	鍵交換	署名
CRYSTALS-Kyber [9]	○	○	
CRYSTALS-Dilithium [23]			○
FALCON [54]			○

- 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

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

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。

3.3.1 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, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé の 10 名により共同で発表され [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^{*2}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 3.3, 3.4, 3.5 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^{*3}。
- 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 (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。

- 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、

$$\hat{f}_{2i} = \sum_{j=0}^{127} f_{2j} \zeta^{(2br_7(i)+1)j} \quad \text{および} \quad \hat{f}_{2i+1} = \sum_{j=0}^{127} f_{2j+1} \zeta^{(2br_7(i)+1)j}$$

^{*2} 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。

^{*3} 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、 $\text{br}_7(i)$ 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 $\mathbf{s} = (s_0, s_1, \dots, s_{k-1}) \in R_q^k$ にも有効で、 $\text{NTT}(\mathbf{s}) = (\text{NTT}(s_0), \text{NTT}(s_1), \dots, \text{NTT}(s_{k-1}))$ 等と解釈する．

- $\text{Parse}(\text{XOF}(\rho, i, j))$: 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．
- $\text{CBD}_\eta(\text{PRF}(\sigma, i))$: 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 512 η bits 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 2 η bits ごとに切り分け $i = 0, \dots, 255$ 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．
- $\text{Encode}_\ell(\hat{\mathbf{s}}), \text{Decode}_\ell(\mathbf{b})$: Encode_ℓ 関数は 255 次の多項式 $\hat{\mathbf{s}} \in R_q$ を入力とし、各係数を ℓ bits のビット列に直したものを結合した 256 ℓ bits 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．
- $\text{Compress}_q(x, d), \text{Decompress}_q(x, d)$: $x \in \mathbb{Z}_q$ 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

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

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ことから、高速化を狙ったものである．以下の表 3.2 に用いられる関数をまとめる．なお、本節で紹介する IND-CPA 安全な方式の中では XOF, PRF および G のみが用いられ、他の 2 つは IND-CCA2 安全な方式の構成において呼び出される．

90s version の XOF 関数では、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 $\ell = 2, \dots, 11$ 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 $\ell = 1, \dots, 11$ 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^{*4} を参照した．

表 3.2: CRYSTALS-Kyber における擬似乱数生成器の実装 [9, Sect. 1.4]

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

CRYSTALS-Kyber の鍵生成関数 (表 3.3 右) を説明する．表の中で $\mathcal{B}$ は 1Byte 分の情報を表す集合 $\{0, 1, \dots, 255\}$ 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 $\mathbf{s}, \mathbf{e} \in R_q^k$ をサンプリングするためのシー

^{*4} <https://github.com/pq-crystals/kyber/blob/master/ref/symmetric-shake.c>

表 3.3: Lindner-Peikert 格子ベース暗号および CRYSTALS-Kyber における鍵生成関数の比較

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

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。

$s, e \in R_q^k$ 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 $\hat{t}$ は $\text{NTT}(As + e)$ となる。公開鍵サイズを圧縮するため、 $A, \hat{t}$ 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 $\hat{s}$ に関しても同様である。

表 3.4: Lindner-Peikert 格子ベース暗号および CRYSTALS-Kyber における暗号化関数の比較

	Lindner-Peikert [73, Sect. 3.1] $\text{Enc}(pk = (A, B), m \in \{0, 1\}^\ell) \rightarrow \text{ct}$	CRYSTALS-Kyber [9, Algorithm 5] $\text{Enc}(pk = (T \parallel \rho), m \in \mathcal{B}^{32}) \rightarrow \text{ct}$
0:		$\hat{t} \leftarrow \text{Decode}_{12}(T)$ $\hat{A}^T[i][j] \leftarrow \text{Parse}(\text{XOF}(\rho, i, j))$ // 行列 $\hat{A}$ の転置の形での復元
1:	s', e', e'' : 成分の小さいベクトル	$r[i] \leftarrow \text{CBD}_{\eta_1}(\text{PRF}(r, i))$ for $i = 0, \dots, k-1$ $e_1[i] \leftarrow \text{CBD}_{\eta_2}(\text{PRF}(r, i+k))$ for $i = 0, \dots, k-1$ $e_2 \leftarrow \text{CBD}_{\eta_2}(\text{PRF}(r, 2k))$
2:	$u = s'A + e'$	$\hat{r} \leftarrow \text{NTT}(r)$ $u \leftarrow \text{NTT}^{-1}(\hat{A}^T \circ \hat{r}) + e_1$ $v \leftarrow \text{NTT}^{-1}(\hat{t}^T \circ \hat{r}) + e_2 + \text{Decompress}_q(\text{Decode}_1(m), 1)$ $c_1 \leftarrow \text{Encode}_{d_u}(\text{Compress}_q(u, d_u))$ $c_2 \leftarrow \text{Encode}_{d_v}(\text{Compress}_q(v, d_v))$
return	$\text{ct} = (u, v)$	$\text{ct} = (c_1 \parallel c_2)$

CRYSTALS-Kyber の暗号化関数（表 3.4 右）を説明する。圧縮形で入力された公開鍵から $\hat{t}, \hat{A}$ 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 $r, e_1 \in R_q^k$ 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 $c_1 \parallel c_2$ は通常空間でのベクトル $u \in R_q^k$ と多項式 $v \in R_q$ を 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 3.5: Lindner-Peikert 格子ベース暗号および CRYSTALS-Kyber における復号関数の比較

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

CRYSTALS-Kyber の復号関数 (表 3.5 右) は圧縮されたビット列の展開, NTT 空間の利用などで表現が煩雑になっているが, Lindner-Peikert 暗号の復号処理と本質的に同様である．最後の $\text{Compress}_q(\cdot, 1)$ 関数が Lindner-Peikert 暗号における $\bar{\mathbf{m}}$ から $\mathbf{m}'$ 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 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す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 を大きくす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と、大きくす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 $(\mathbf{u}, \mathbf{v})$ をビット列で表現するための精度を指定する．小さくす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す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．

3.3.2 CRYSTALS-Dilithium

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

表 3.6: 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 3.7, 3.8, 3.10 に Lyubashevsky による Fiat-Shamir with Aborts 型の格子ベース署名, CRYSTALS-Dilithium のテンプレートアルゴリズム [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 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 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 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 w から取り出す上位ビットの長さ.

用いられるサブルーチンのうち主なものを以下に列挙する.

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

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

で計算される. ただし, $r = 1753$, $r_i = r^{\text{brv}(128+i)} \bmod q$, $\text{brv}(k)$ 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.
- ExpandA(ρ): 乱数生成のシード ρ を用いて, ランダム行列 $A \in R_q^{k \times l}$ を生成し, その NTT 表現

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

を計算し出力する.

- ExpandS(ρ'): 署名に用いる多項式 s_1, s_2 を生成するための関数で, 512bits のシードを入力とする.
- Power2Round $_q(t, d)$, HighBits $_q(t, \alpha)$, LowBits $_q(t, \alpha)$: $\mathbb{Z}_q$ の元 t で, $0 \leq t < q$ を満たすものを $t = r_1 \cdot 2^d + r_0$, $-q/2 < r \leq q/2$ と分解したときに $\text{Power2Round}_q(t, d) = (r_1, r_0)$ 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 $\mathbf{t}$ 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, $\mathbf{t} = \left(\sum_{j=1, \dots, k} t_{j,i} x^i \right)_{j=1, \dots, k}$ と書いたときに $\text{Power2Round}_q(t_{j,i}, d) \rightarrow (t_{j,i,1}, t_{j,i,0})$ とすれば, $\text{Power2Round}_q(\mathbf{t}, d) \rightarrow (\mathbf{t}_1, \mathbf{t}_0)$ は $\mathbf{t}_1 = \left(\sum_{j=1, \dots, k} q_{j,i} x^i \right)_{j=1, \dots, k}$, $\mathbf{t}_0 = \left(\sum_{j=1, \dots, k} r_{j,i} x^i \right)_{j=1, \dots, k}$, 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 $t = r_1 \cdot \alpha + r_0$, $-q/2 < r_0 \leq q/2$ の形で分解し, HighBits $_q(t, \alpha)$, LowBits $_q(t, \alpha)$ をそれぞれ r_1, r_0 で定義する.
- MakeHint $_q(z, r, \alpha)$, UseHint $_q(h, r, \alpha)$: MakeHint $_q$ 関数は HighBits $_q(r, \alpha) \neq \text{HighBits}_q(r+z, \alpha)$ であれば 1 を, そうでなければ 0 を返す関数である. UseHint $_q$ 関数は引数から 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.
- 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 $\hat{c} = \text{NTT}(c)$ が出力される.
- $\#_1 \mathbf{h}$ は $\mathbf{h} = \sum_{i=0}^{255} h_i$ の中で $h_i = 1$ となる項の数を表す.

表 3.7 の鍵生成関数について記述する. 256bits のシード ζ をハッシュ関数 H により合計 1024bits に伸長し, そのうち ρ, ρ' をそれぞれ公開鍵 A のシード, 秘密鍵 s_1, 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 s_1, 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 $S_\eta = \{\mathbf{w} \in R : \|\mathbf{w}\|_\infty \leq \eta\}$ から一様ランダムにサンプリングされる.

Fiat-Shamir 型署名における秘密鍵 $\hat{s}$ のハッシュ関数 $a(\hat{s})$ の計算が, ベクトル (s_1, s_2) と行列 A を用いた $As_1 + s_2$ の計算に対応している.

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

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

表 3.8 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 3.7: CRYSTALS-Dilithium における鍵生成関数の比較

	格子ベース署名 [77, Fig. 4] $\text{KeyGen}(1^\lambda) \rightarrow (pk, sk)$	CRYSTALS-Dilithium テンプレート [23, Fig. 1] $\text{KeyGen}(1^\lambda) \rightarrow (pk, sk)$	CRYSTALS-Dilithium 実装のための擬似コード [23, Fig. 4] $\text{KeyGen}(1^\lambda) \rightarrow (pk, sk)$
1:	$\hat{s}$: 短い多項式を 成分とするベクトル	$s_1 \leftarrow S_\eta^l, s_2 \leftarrow S_\eta^k$	$\zeta \xleftarrow{\$} \{0, 1\}^{256}$ $H(\zeta) \rightarrow (\rho, \rho', K) \in \{0, 1\}^{256} \times \{0, 1\}^{512} \times \{0, 1\}^{256}$ $\text{ExpandS}(\rho') \rightarrow (s_1, s_2) \in S_\eta^\ell \times S_\eta^k$
2:	a : ハッシュ関数	$\hat{A} \xleftarrow{\$} R_q^{k \times l}$	$\text{ExpandA}(\rho) \rightarrow \hat{A} \in R_q^{k \times \ell}$
3:	$t \leftarrow a(\hat{s})$	$t = As_1 + s_2$	$t \leftarrow \text{NTT}^{-1}(\hat{A} \cdot \text{NTT}(s_1)) = As_1 + s_2$ $\text{Power2Round}_q(t, d) \rightarrow (t_1, t_0)$ $H(\rho t_1) \rightarrow tr \in \{0, 1\}^{256}$
return	$sk = (a, \hat{s}), pk = (a, t)$	$sk = (A, t, s_1, s_2), pk = (A, t)$	$sk = (\rho, K, tr, s_1, s_2, t_0), pk = (\rho, t_1)$

シードとなる値で、 $H(K || \mu) \rightarrow \rho'$ 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 s_1, s_2, 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 (z, 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 $y \in R_q^l$ をサンプリングする。ここで、各成分は $\tilde{S}_{\gamma_1} = \left\{ \sum_{i=0}^{255} w_i x^i : -\eta < w_i \leq \eta \right\}$ から一様ランダムにサンプリングされる。このサンプリングは表 3.8 中央の $y \leftarrow D_{\gamma_1}^{l \times 1}$ 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 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、 w_1 は公開鍵 A と直前でサンプリングした y から来る要素である。計算効率のため、内積 $c \cdot s_1$ は NTT 表現で計算された後に逆変換をかけ $z = y + c \cdot s_1$ となる。

ステップ 5 では $z \notin G$ のチェックのため、 z と $w - cs_2$ の下位ビットの ℓ_∞ ノルムがそれぞれ比較される。両方が閾値よりも小さい場合には次のヒント生成関数 (*) が実行される。ヒント生成関数は表 3.9 により示され、 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 $(z, h) \leftarrow \perp$ 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 y の生成からやり直すことになる。ここで、 $-cs_2 + ct_0$ の計算は前半を r_0 で用いたものを使いまわし、後半を $\text{NTT}^{-1}(\hat{c} \cdot \hat{t}_0)$ の形で計算する。

表 3.10 の署名検証関数について記述する。公開鍵、署名に含まれる乱数のシード $\rho, \tilde{c}$ から $\hat{A}, \hat{c}$ を復元し、メッセージに対応するハッシュ値 μ を計算する。 $Az - ct_1 \cdot 2^d$ は $\hat{A} \cdot \text{NTT}(z) - \text{NTT}(\hat{c}) \cdot \text{NTT}(t_1 \cdot 2^d)$ の形で計算する。これらの値から UseHint_q を用いて w'_1 を復元し、 z のノルム、 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 3.8: 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] $\text{Sign}(sk = (a, \hat{s}), \mu \in \{0, 1\}^*) \rightarrow \sigma$	CRYSTALS-Dilithium テンプレート [23, Fig. 1] $\text{Sign}(sk = (A, t, \mathbf{s}_1, \mathbf{s}_2), \mu \in \{0, 1\}^*) \rightarrow \sigma$	CRYSTALS-Dilithium 実装のための擬似コード [23, Fig. 4] $\text{Sign}(sk = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0), M \in \{0, 1\}^*) \rightarrow \sigma$
0:			$\text{ExpandA}(\rho) \rightarrow \hat{A}$ $H(tr M) \rightarrow \mu \in \{0, 1\}^{512}$ $\kappa \leftarrow 0, (z, \mathbf{h}) \leftarrow \perp$ $H(K \mu) \rightarrow \rho' \in \{0, 1\}^{512}$ $\hat{\mathbf{s}}_1 \leftarrow \text{NTT}(\mathbf{s}_1); \hat{\mathbf{s}}_2 \leftarrow \text{NTT}(\mathbf{s}_2)$ $\hat{\mathbf{t}}_0 \leftarrow \text{NTT}(\mathbf{t}_0)$
1:	$z \leftarrow \perp$	$z \leftarrow \perp$	while $(z, \mathbf{h}) = \perp$ do $\text{ExpandMask}(\rho', \kappa) \rightarrow \mathbf{y} \in \tilde{S}_{\gamma_1}^l$
2:	while $z = \perp$ do	while $z = \perp$ do	
3:	$\hat{\mathbf{y}}$: 短い多項式を成分とするベクトル	$\mathbf{y} \leftarrow D_{\gamma_1-1}^{l \times 1}$	
4:	$c \leftarrow H(a(\hat{\mathbf{y}}) \mu)$	$\mathbf{w}_1 \leftarrow \text{HighBits}(A\mathbf{y}, 2\gamma_2)$ $c = H(\mu \mathbf{w}_1)$	$\mathbf{w} \leftarrow \text{NTT}^{-1}(\hat{A} \cdot \text{NTT}(\mathbf{y})) = A\mathbf{y}$ $\mathbf{w}_1 \leftarrow \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$ $H(\mu \mathbf{w}_1) \rightarrow \tilde{c} \in \{0, 1\}^{256}$ $\text{SampleInBall}(\tilde{c}) \rightarrow \hat{c} \in B_\tau \subset R_q$ $\mathbf{z} \leftarrow \mathbf{y} + \text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$ $\mathbf{r}_0 \leftarrow \text{LowBits}_q(\mathbf{w} - \text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2), 2\gamma_2)$ if $(\ \mathbf{z}\ _\infty \geq \gamma_1 - \beta)$ OR $(\ \mathbf{r}_0\ _\infty \geq \gamma_2 - \beta)$ then $(z, \mathbf{h}) \leftarrow \perp$ else $\mathbf{h} \leftarrow \text{MakeHint}_q(\cdot) \dots (*)$ $\kappa \leftarrow \kappa + l$
5:	$\hat{\mathbf{z}} \leftarrow \hat{\mathbf{y}} + c\hat{\mathbf{s}}$ if $\hat{\mathbf{z}} \notin G^m$ then $z \leftarrow \perp$	$\mathbf{z} \leftarrow \mathbf{y} + c\mathbf{s}_1$ $\mathbf{r}_0 \leftarrow \text{LowBits}(A\mathbf{y} - c\mathbf{s}_2, 2\gamma_2)$ if $(\ \mathbf{z}\ _\infty \geq \gamma_1 - \beta)$ OR $(\ \mathbf{r}_0\ _\infty \geq \gamma_2 - \beta)$ then $z \leftarrow \perp$	
return	$\sigma = (\hat{\mathbf{z}}, c)$	$\sigma = (\mathbf{z}, c)$	$\sigma = (\mathbf{z}, \mathbf{h}, \tilde{c})$

表 3.9: 署名生成関数におけるヒント生成時のチェック関数

$$\mathbf{h} \leftarrow \text{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$$

$$\text{if } \|c\mathbf{t}_0\|_\infty \geq \gamma_2 \text{ OR } \#_1 \mathbf{h} > \omega \text{ then } (z, \mathbf{h}) \leftarrow \perp$$

仕様書に掲載されたパラメータセットを表 3.11 に示す。セキュリティ強度を規定するパラメータのうち、問題が定義される環とモジュールのランクに関わるものが (n, k, l, q) の 4 個、ノイズに関わるものが $(\eta, \gamma_1, \gamma_2, \beta, \tau, d)$ の 6 個である。

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 3.10: 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]
	$\text{Vrfy}(pk = (a, t), \mu \in \{0, 1\}^*, \sigma = (\hat{z}, c))$	$\text{Vrfy}(pk = (A, t), \mu \in \{0, 1\}^*, \sigma = (z, c))$	$\text{Vrfy}(pk = (\rho, t_1), M \in \{0, 1\}^*, \sigma = (z, h, \tilde{c}))$
0:			$\text{ExpandA}(\rho) \rightarrow \hat{A}$ $H(H(\rho t_1) M) \rightarrow \mu \in \{0, 1\}^{512}$ $\text{SampleInBall}(\tilde{c}) \rightarrow \tilde{c}$
1:	if $\hat{z} \in G^m$ AND $c = H(a(\hat{z}) - tc, \mu)$ then accept else $\perp$	$w'_1 = \text{HighBits}(Az - ct, 2\gamma_2)$ if $\ z\ _\infty < \gamma_1 - \beta$ AND $c = H(M w'_1)$ then accept else $\perp$	$w'_1 \leftarrow \text{UseHint}_q(h, Az - ct_1 \cdot 2^d, 2\gamma_2)$ if $\ z\ _\infty < \gamma_1 - \beta$ AND $\tilde{c} = H(\mu w'_1)$ AND $\#_1 h \leq \omega$ then accept else $\perp$

表 3.11: 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。

3.3.3 FALCON

歴史: FALCON は 2017 年 11 月の NIST PQC 公募に Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang の 10 名を開発者として公表された [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 3.12, 3.13, 3.14 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。

- $\text{FFT}(f), \text{invFFT}(s)$: 多項式 $f \in \mathbb{R}[x]/\phi$ に対して、そのフーリエ変換 $\text{FFT}(f)$ 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 $\text{invFFT} : \mathbb{R}^n \rightarrow \mathbb{R}[x]/\phi$ 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 $\text{FFT}(f) \odot \text{FFT}(g)$ を成分ごとの積と定義する。FFT 表現での多項式の積 $\text{FFT}(fg)$ の計算に対応する。
- $\text{HashToPoint}(\text{str}, q, n)$: 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。
- $\text{Compress, Decompress}$: 多項式 $s \in \mathbb{Z}[x]$ を文字列に変換する関数とその逆関数とする。

表 3.12: Hash-and-Sign 型格子ベース署名および FALCON における鍵生成関数の比較

	Gentry らの格子ベース署名 [60, Sect. 7.1] $\text{KeyGen}(1^\lambda) \rightarrow (pk, sk)$	FALCON[54, Algorithm 4] $\text{KeyGen}(\phi, q) \rightarrow (pk, sk)$
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	$f, g, F, G \leftarrow \text{NTRUGen}(\phi, q)$ $B \leftarrow \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$ $\hat{B} \leftarrow \text{FFT}(B)$ $G \leftarrow \hat{B} \times \hat{B}^*$ $T \leftarrow \text{ffLDL}^*(G)$ for each leaf leaf of T do $\text{leaf.value} \leftarrow \sigma / \sqrt{\text{leaf.value}}$ $h \leftarrow gf^{-1} \pmod{q}$
return	$pk = A, sk = B$	$pk = h, sk = (\hat{B}, T)$

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

$$fG - gF = q \pmod{\phi} \quad (3.1)$$

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

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

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

また、署名の生成には $sA \equiv H(m)$ を満たす短いベクトル s を生成する必要がある。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

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

およびそれを元にした LDL 木と呼ばれる木構造 T である。木の中には $\hat{B}$ のグラム行列 $G = \hat{B} \times \hat{B}^*$ の^{*5} 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 $\sigma/\sqrt{\text{leaf.value}}$ に書き換えることで鍵生成が完了する。

表 3.13: Hash-and-Sign 型格子ベース署名および FALCON における署名生成関数の比較

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

表 3.13 の署名生成関数の説明を記述する。平文にランダムビット r を結合した後、`HashToPoint` 関数で多項式 $c \in \mathbb{Z}_q/\phi$ を出力する。関係式 (3.1), (3.2) より、ベクトル $\hat{t}$ は $(\text{FFT}(c), \text{FFT}(0))\hat{B}^{-1}$ 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 `ffSamplingn` 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 $z \in \mathbb{Z}^{2n}$ を、 $t = [c, 0]B^{-1}$ を中心として $\exp(-\|(z - t)B\|^2/2\sigma^2)$ 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、 $(t - z)B$ は原点を中心とした集合

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

上の離散ガウス分布となるため、 s は短く、かつ

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

が成り立つ。このとき、 $sA = c$ の関係から $s_1 + s_2h = c$ が成り立つ。この関係式が署名の検証時に用いられる。

サンプリングされた $\hat{s}$ が $\|\hat{s}\|^2 \leq \lfloor \beta^2 \rfloor$ 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 3.14 の署名検証関数の説明を記述する。平文、ハッシュ関数のシード値、署名文字列から各要素を復元し、 $s_1 = c - s_2h$ を計算する。署名が正しく生成されていれば $sA = c$ 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。

^{*5} B^* は体 $\mathbb{Q}[x]/\phi$ におけるエルミート共役。詳細は [54, p.23]

表 3.14: Hash-and-Sign 型格子ベース署名および FALCON における署名検証関数の比較

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

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、大きく取ることによって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、大きくとることによって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、大きくとることによって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、大きくとることによって署名生成時のやり直し回数がかかるが、安全性が低下する。

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

表 3.15: 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 * ⁶	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。

*⁶ 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。

3.4 格子に基づく暗号技術に関するまとめ

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

この米国 NIST PQC プロジェクトを通じて 2022 年 7 月に CRYSTALS-Kyber が標準的な暗号方式として, CRYSTALS-Dilithium および FALCON が標準的な署名方式として選定された. また, CRYSTALS-Kyber と CRYSTALS-Dilithium は 2022 年 9 月に米国国家安全保障局の Commercial National Security Algorithm Suite 2.0 (CNSA2.0) にも選定されている [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 CECQP1[31] および CECQP2[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 (Bounded Distance Decoding) 問題, SIS (Small Integer Solution) 問題他, 多くのバリエーションが存在している. 一般的な格子問題を解く手法としては, LLL アルゴリズム, BKZ アルゴリズムがよく知られており, LWE 問題については更に SIS 問題や BDD 問題に還元する解析手法が知られている.

格子問題の困難性をベースとした暗号方式で最初のものは, Ajtai[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]^{*7} 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deal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

^{*7} 文献上は 1998 年の国際会議 ANTS だが, 初出は CRYPTO1996 の Rump Session である.

要な三つの性質が示された。一つは問題の average-case to worst case reduction, つまりパラメータを固定した際, 問題の (秘密ベクトル $\mathbf{s}$ に関する) 平均的な計算量が, 最悪計算量 (難しいインスタンスを生成するような $\mathbf{s}$ の集合に対する計算量) と高々多項式倍の違いしか無いことであり, 残りの二つは判定 LWE と探索 LWE の等価性, および量子アルゴリズムによる困難な格子問題への還元である。これらの定理を組み合わせることにより, Regev 自身により提案された公開鍵暗号を解読することが平均的に難しいことが示され, その後の様々な LWE ベース暗号の構成の基礎となった。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 3.1 節に, 2022 年 8 月現在の状況について記載した。特に 3.3 節で示された各暗号方式のパラメータから見ると, 解が得られている値からは, 大きな隔たりがみられる。格子に基づく暗号技術は, 各方式毎にパラメータ設定手法に対する制約が異なっていることから, 解読コンテストのサイズに基づく解読到達レベルを, 具体的な暗号方式の安全性の根拠とすることは, 難しいところではあるものの, 古典計算機での解読困難性を測る上での検討の一つに値すると考えられる。

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

第 3 章の参考文献

- [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 $\geq$ 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 閱覽)
- [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 閱覽)
- [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 閱覽)
- [34] The Chromium Projects. CECpq2. <https://www.chromium.org/cecpq2/>. (2023-02-21 閱覽)
- [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 閱覽)
- [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 閱覽)
- [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 閱覽)
- [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 閱覽)

- [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 閱覽)
- [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 閱覽)
- [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 閱覽)
- [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 閱覽)
- [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 閱覽)
- [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 閱覽)
- [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 4 章

符号に基づく暗号技術

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

■準備: 本章で使用する記号・用語を以下にまとめる．以下では, q を素数 p のべきとする．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 $\mathbf{v} = (v_1, v_2, \dots, v_n) \in V_n$ のハミング重みとは, 非ゼロの係数の数である．すなわち, $\text{HW}(\mathbf{v}) = \#\{i \mid v_i \neq 0\}$ である．
- ハミング距離を $d_H(\mathbf{x}, \mathbf{y}) = \text{HW}(\mathbf{x} - \mathbf{y})$ 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 $\mathbf{c}$ は, 一次独立な k 個の符号語 $\mathbf{c}_1, \dots, \mathbf{c}_k$ の $\mathbb{F}_q$ 係数の一次結合で表すことができる．このとき $[\mathbf{c}_1, \dots, \mathbf{c}_k]$ 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 $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ 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 $\mathbf{H} \in \mathbb{F}_q^{r \times n}$ で, $\mathbf{c} \in \mathbb{F}_q^n$ に対して, $\mathbf{c} \in \mathcal{C}$ ならばかつその時に限り $\mathbf{c} \cdot \mathbf{H}^\top = \mathbf{0}$ となるものである． $\mathbf{H}$ 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, $d = \min_{\mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}} \text{HW}(\mathbf{c})$ である．

$[n, k]_q$ -線形符号 $\mathcal{C}$ の生成行列 $\mathbf{G}$ が決まっている場合, メッセージ $\mathbf{s} \in \mathbb{F}_q^k$ を 符号語 $\mathbf{sG}$ 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 $\mathbf{c}$ とし, 通信路上で乗った誤りを $\mathbf{e}$ とする．受信者側は受信語として, $\mathbf{y} = \mathbf{c} + \mathbf{e}$ 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, $\mathbf{y}$ から $\mathbf{c}$ を得る．受信者がハミング重み t までの誤り $\mathbf{e}$ 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, $\mathbf{t} = \mathbf{yH}^\top$ 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.

4.1 符号に基づく暗号技術の安全性の根拠となる問題

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

4.1.1 LPN 問題とは

LPN 問題とは誤差付きの線形方程式を解けるかどうかという問題である. 1993 年に, Blum, Furst, Kearns, Lipton [8] が困難と思われる問題として挙げ, 定式化を行った. 第 3 章において, この問題を一般化した LWE 問題を既に扱っている.

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

■ **LPN 問題:** $\mathbb{F}_2$ 上の分布 χ および $\mathbf{s} \in \mathbb{F}_2^k$ について, オラクル $\mathcal{O}_{\mathbf{s}, \chi}$ を以下で定義する. (1) $\mathbf{a}$ 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) $b = \mathbf{s} \cdot \mathbf{a}^\top + e$ と計算し, (4) $(\mathbf{a}, b)$ を出力する.

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

定義 4.1 (探索版 LPN 問題) 探索版 LPN 問題とは, オラクル $\mathcal{O}_{\mathbf{s}, \chi}$ へのアクセスが可能なときに, $\mathbf{s}$ を出力する問題である.

特に $\chi = \text{Ber}_\tau$ のとき, $\text{LPN}_{k, \tau}$ 問題と呼ぶ. また $\text{LPN}_{k, \tau}$ 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, $\text{LPN}_{k, n, \tau}$ 問題と呼ぶ.

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

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

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

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

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

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 $\mathbf{s} \in \mathbb{F}_2^k$ について探索版 LPN 問題を解けるならば, 任意の $\mathbf{s} \in \mathbb{F}_2^k$ 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 4.5 ([32]) 判定版 $\text{LPN}_{k,\tau}$ 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 $\text{LPN}_{k,\tau}$ 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,

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

■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 を取る確率が大いことが求められる. たとえば, ベルヌーイ分布の一般化として, 確率 τ で 0 を確率 $1 - \tau$ で $\mathbb{F}_q \setminus \{0\}$ のランダムな値を取る分布が用いられる. これは格子問題と符号問題のアナロジーとして考えることができる.

4.1.2 LPN 問題の拡張

4.1.2.1 復号問題

オラクルからのサンプル数を固定し $n = n(k)$ とする. $\text{LPN}_{k,n,\tau}$ 問題での m 個のサンプル $(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)$ を行列・ベクトル表示して,

$$\mathbf{A} = [a_1^\top a_2^\top \dots a_n^\top] \in \mathbb{F}_2^{k \times n}, \quad \mathbf{b} = \mathbf{s} \cdot \mathbf{A} + \mathbf{e}$$

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

4.1.2.2 シンドローム復号問題

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

$$\mathbf{H} = [h_1^\top h_2^\top \dots h_n^\top] \in \mathbb{F}_2^{(n-k) \times n}, \quad \mathbf{u} \in \mathbb{F}_2^k$$

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

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

4.1.2.3 Module-LPN 問題

Module-LWE 問題 (3.1.1 節) と同様に, Module-LPN 問題を定義することができる.

定義 4.6 (探索版 Module-LPN 問題) 適当な k_0 次の $\mathbb{F}_q$ 係数多項式 $f(x)$ を考え, 環 $R_q = \mathbb{F}_q[x]/(f(x))$ を固定する. R_q 上の確率分布 χ を固定する.

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

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

4.1.3 LPN 問題に対する評価

サンプル数を固定した場合, $\mathbf{A}$ および $\mathbf{b}$ 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.

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

4.1.3.1 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 $(\mathbf{a}, b)$ が常に $\mathbf{a} = (1, 0, \dots, 0)$ 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 $\mathbf{u}_j$ を j 番目の単位ベクトルとして, $(\mathbf{u}_j, b)$ 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 $\mathcal{O}_{\mathbf{s}, \tau}$ 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.

Blum らの見積もりによれば, うまくパラメータを設定することで, 時間計算量・空間計算量ともに $2^{O(k/\log k)}$ を得る. その後, Levieil と Fouque [36], Kirchner [30], Guo, Johansson, Löndahl [25], Zhang, Jiao, Wang [47], Bogos と Vaudenay [16], Esser, Kübler, May [21], Esser, Heuer, Kübler, May, Sohler [20] などで改良やメモリと時間のトレードオフの議論がおこなわれている.

■**サンプル数が少ない場合:** これまでに挙げてきた BKW アルゴリズムおよびその改良では, サンプルが $O(2^{k/\log k})$ 個必要であった. Lyubashevsky [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, Kübler, 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.

4.1.3.2 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 LPN $_{k,n,\tau}$ に用いた場合, $w = \tau n$ として, $\text{poly}(k^w)$ 時間で解くことができる.

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

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

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

$\text{poly}(k^w) = 2^{O(\tau n \log k)}$ であるから, $\tau = o(k/(n \log^2 k))$ のようにエラーが疎であれば, BKW アルゴリズムよりも効率が良い. 実際の符号暗号のパラメータ設定では, エラーをこのように疎に設定することはないため, 暗号の攻撃アルゴリズムとして用いるには重要度が低い.

4.1.3.3 SD 問題を經由するアルゴリズム

$\text{LPN}_{k,n,\tau}$ に対応するシンδροーム復号問題を考える. ハミング重みを $w \approx \tau n$ とし, $\mathbf{H} \in \mathbb{F}_2^{(n-k) \times n}$ および $\mathbf{u} \in \mathbb{F}_2^{n-k}$ が与えられ, $\mathbf{e} \cdot \mathbf{H}^\top = \mathbf{u}$ となるような, ハミング重みが w 以下の $\mathbf{e} \in \mathbb{F}_2^n$ 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, Gilbert-Varshamov 限界により, $k/n \approx 1 - H(d/n)$ である^{*4}.) $w \approx d$ の場合を Full Distance Decoding の場合と呼び, $w \approx d/2$ の場合を Half Distance Decoding と呼ぶ.

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

$\mathbf{u}'$ のハミング重みが w 以下であるため, $\mathbf{e}$ のハミング重みも w 以下である. また, $\mathbf{e} \cdot \mathbf{H}^\top = (\mathbf{u}', \mathbf{0}_k) \cdot \mathbf{P}^\top \mathbf{H}^\top = (\mathbf{u}', \mathbf{0}_k) \cdot \tilde{\mathbf{H}}^\top = (\mathbf{u}, \mathbf{0}_k) \mathbf{S}^\top \tilde{\mathbf{H}}^\top = (\mathbf{u}, \mathbf{0}_k) \cdot [\mathbf{I}_{n-k} \mid \mathbf{Z}]^\top = \mathbf{u}$ が成立する. よって, ステップ 4 のチェックを通るならば, $\mathbf{e}$ はシンδρο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 $\text{poly}(n, k) \cdot O\left(\binom{n}{w} / \binom{n-k}{w}\right)$ 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 4.1 および 4.2 に示す. この表は, 時間計算量を最小化した場合の $R = k/n$ の最悪時 ($1/2$ の少し下) についてまとめられている. したがって, 問題のパラメータによっては, 表の数値よりも速く解くことが可能となる.

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

4.1.3.4 量子アルゴリズムへの耐性

現在のところ多項式時間で LPN 問題を解く量子アルゴリズムは提案されていない. しかし量子アルゴリズムを利用した攻撃の高速化方法を Kachigar と Tillich [33] が提案している.^{*5} 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.

^{*4} ここで $H(p) = -p \log(p) - (1-p) \log(1-p)$.

^{*5} 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.

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

	$\log(\text{Time})/n$	$\log(\text{Space})/n$	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???
BM17	0.0953	0.0910	[13]; MO15 を最適化したもの
BM18	0.0885	0.0736	[14]

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

	$\log(\text{Time})/n$	$\log(\text{Space})/n$	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???
BM18	0.0465	0.0294	[14]

4.1.3.5 現状の進展

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

1. $\mathbb{F}_2$ 係数の一様ランダムな線形符号に対するシンドローム復号問題
2. $\mathbb{F}_2$ 係数の一様ランダムな線形符号に対するハミング重みが小さい符号語を探索する問題
3. $\mathbb{F}_3$ 係数の一様ランダム線形符号に対するシンドローム復号問題
4. Goppa 符号を用いた Niederreiter 暗号の場合のシンドローム復号問題 (Classic McEliece の一方向性に対応 4.3.1)
5. QC-MDPC 符号に基づくシンドローム復号問題のチャレンジ (BIKE や HQC の一方向性に対応 4.3.2 4.3.3)

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

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

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

4.2 符号に基づく代表的な暗号方式

本節では, 符号に基づく代表的な暗号方式と署名方式の説明を行う。以下では, $\text{GL}_k(\mathbb{F}_q)$ 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 $\text{GL}_n(\mathbb{F}_q)$ 中の置換行列と同一視することとする。

4.2.1 McEliece 暗号

McEliece [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 G を生成する。 $S \leftarrow \text{GL}_k(\mathbb{F}_2)$ を一様ランダムに選ぶ。 $P \leftarrow S_n$ を一様ランダムに選ぶ。 $\tilde{G} = SGP$ とする。

公開鍵を $\tilde{G}$ とし, 秘密鍵を (S, G, P) とする。

暗号化: 平文を $m \in \mathbb{F}_2^k$ とする。乱数 $e \leftarrow \text{Ber}_\tau^n$ を選び, 暗号文 $c = m\tilde{G} + e$ を計算する。

復号: $\hat{v} = cP^{-1}$ を計算する。 $\hat{v}$ を誤り訂正符号で訂正し $m' \in \mathbb{F}_2^k$ を得る。 $m = m'S^{-1}$ を出力する。

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

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

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

平文 m および $\tilde{G}$ が一様ランダムであれば, 暗号文 c は LPN 仮定の下で擬似ランダムである。 $\tilde{G}$ 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 4.7 (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

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

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つ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。

4.2.2 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 $H \in \mathbb{F}_2^{(n-k) \times n}$ を生成する. $T \leftarrow \text{GL}_{n-k}(\mathbb{F}_2)$ を一様ランダムに選ぶ. $Q \leftarrow S_n$ を一様ランダムに選ぶ. $\tilde{H} = THQ$ とする.
公開鍵を $\tilde{H}$ とし, 秘密鍵を (T, H, Q) とする.

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

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

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

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

を得る. $eQ^\top$ 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, $e' = eQ^\top$ を得る. よって, 高い確率で復号に成功する.

平文 e および $\tilde{H}$ が一様ランダムであれば, 暗号文 d は 判定版 LPN 仮定の下で擬似ランダムである. $\tilde{H}$ 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 4.8 (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

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

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, 暗号文の一方方向性は, 探索版 LPN 仮定および Niederreiter 仮定から言える.

4.3 符号に基づく主要な暗号方式

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 4.3: 符号に基づく暗号の分類

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

4.3.1 Classic McEliece

- 提案者: Albrecht, Bernstein, Chou, Cid, Gilcher, Lange, Maram, von Maurich, Misoczki, Niederhagen, Paterson, Persichetti, Peters, Schwabe, Sendrier, Szefer, Tjhai, Tomlinson, Wang.
- 基本方式の説明: Niederreiter 暗号方式に基づいている. 基本符号方式として $\mathbb{F}_2$ 上の Goppa 符号を利用している. (具体的な Goppa 符号の生成方法や符号化および復号の方法については提案方式の仕様書を参照のこと.)
 $q = 2^m$ とし, $n \leq q$ を用いる. 2 以上の t を $mt < n$ となるように取り, $k = n - mt$ とする.

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

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

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

1. $\mathbf{c}$ に k 個ゼロを加え, $\mathbf{v} = (\mathbf{c}, \mathbf{0}_k) \in \mathbb{F}_2^n$ を考える
 2. Goppa 符号の復号アルゴリズムを用いて, $\mathbf{v}$ と距離 t 以下にある符号語 $\mathbf{d}$ を計算する. (なければ $\perp$ を出力する)
 3. $\mathbf{e} = \mathbf{v} + \mathbf{d}$ とする.
 4. $\text{HW}(\mathbf{e}) = t$ かつ $\mathbf{c} = \tilde{\mathbf{H}}\mathbf{e}$ ならば $\mathbf{e}$ 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, $\mathbf{U}_m^k$ 変換をかけたものとみなせる. (第 3 ラウンドまでは $\mathbf{HU}_m^k$ を用いていた. 以下ではハッシュ関数 $\mathbf{H}: \{0, 1\}^* \rightarrow \{0, 1\}^{256}$ 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 $\mathbf{T}$ 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. $\mathbf{e} \leftarrow \mathcal{S}_H(n, t)$ をあるアルゴリズムに従ってランダム生成する.

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

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

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

表 4.4: 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

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 無しのもので変わらない). 表 4.4 に鍵カプセル化方式の鍵長および暗号文長, 想定セキュリティレベルをまとめた.

4.3.2 BIKE

- 提案者: Aragon, Barreto, Bettaieb, Bidoux, Blazy, Deneuville, Gaborit, Gueron, Güneysu, Aguilar Melchor, Misoczki, Persichetti, Sendrier, Tillich, Zémor, Vasseur, Ghosh, Richter-Brokmann.
- 基本方式の説明: Niederreiter 暗号方式に基づいている. 基本となる符号に QC-MDPC 符号を採用し, 公開鍵サイズを圧縮している. そのため, 鍵や暗号化は格子暗号の一種の NTRU 暗号と非常に近い形をしている点特徴である. 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.

- ch_0 を計算する.
 - 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 $L: \{0, 1\}^* \rightarrow \{0, 1\}^{256}$ を用いて, 平文 $m \in \{0, 1\}^{256}$ と乱数 (e_0, e_1) に対して $(c_0 = E(pk, (e_0, e_1)), c_1 = m \oplus L(e_0, e_1))$ 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 $\text{FO}_m^\mathcal{L}$ 変換を適用したものとみなせる. 以下ではハッシュ関数 $H, L: \{0, 1\}^* \rightarrow \{0, 1\}^{256}$ と $G: \{0, 1\}^* \rightarrow \mathcal{S}_H(2n, t)$ 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.

- $(e_0, e_1) = G(m)$ を計算する.
- $c_0 = e_0 + e_1 h$ と, $c_1 = m \oplus L(e_0, e_1)$ を計算する.
- $K = H(m, (c_0, c_1))$ を計算する.

表 4.5: 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

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. $m' \leftarrow c_1 \oplus L(e'_0, e'_1)$ を計算する.

4. $(e'_0, e'_1) = G(m')$ ならば, $K = H(m', (c_0, c_1))$ を出力して停止

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

表 4.5 に鍵カプセル化方式の鍵長および暗号文長をまとめた. 3 つのパラメータセットがそれぞれレベル 1, 3, 5 相当として提案された.

4.3.3 暗号方式 3: HQC

- 提案者: Aguilar Melchor, Aragon, Bettaieb, Bidoux, Blazy, Deneuville, Gaborit, Persichetti, Zémor, Bos, Dion, Lacan, Robert, Veron.
- 基本方式の説明: 符号版の LPR/LP 暗号に基づき, 公開鍵暗号を構成している. $\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1)$ とする. $n' = n_1 n_2$ とし, $[n', k]$ 線形符号 $\mathcal{C}$ を採用する. 線形符号 $\mathcal{C}$ の符号化・復号アルゴリズムを $\text{encode}, \text{decode}$ 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 $E(pk, m; s, e_1, e_2) : c = (u, v) = (sa + e_1, sb + e_2 + \text{encode}(m))$ 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, $\text{decode}(v - ux)$ 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, $\text{HFO}^\perp$ 変換を適用したものとみなせる. 以下ではハッシュ関数 $H, H' : \{0, 1\}^* \rightarrow \{0, 1\}^{256}$ を用いる. また, XOF^{*6} として $H_G : \{0, 1\}^* \rightarrow \{0, 1\}^*$ 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, $pk = (\text{seed}, b)$ を公開する. また, 秘密鍵にもシード seed を加える.

鍵カプセル化: 1. $m \leftarrow \mathbb{F}_2^k$ を一様ランダムにとる.

2. $\text{salt} \leftarrow \mathbb{F}_2^{128}$ を一様ランダムにとる.

3. $\theta \leftarrow H_G(m, \text{seed}, \text{salt})$ を計算する. θ から s, e_1, e_2 を生成する.

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

表 4.6: HQC のパラメータ. 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

4. $c = (u, v) = E(pk, m; s, e_1, e_2)$ を計算する. $d = H'(m)$ とする. $K = H(m, c)$ とする.

5. 暗号文 $C = (c, d, \text{salt})$, セッション鍵 K を出力する.

デカプセル: 1. $m' \leftarrow D(sk, c)$ を計算する.

2. $\theta' = H_G(m', \text{seed}, \text{salt})$ を計算する. θ' から s', e'_1, e'_2 を生成する.

3. $c \neq E(pk, m'; s', e'_1, e'_2)$ or $d \neq d'$ ならば $\perp$ を出力して停止する.

4. $K = H(m, c)$ を出力する.

3つのパラメータセットがそれぞれレベル 1, 3, 5 相当として提案された. 表 4.6 に鍵カプセル化方式の鍵長および暗号文長をまとめた. 表中では, 秘密鍵はシードだけ記憶していることにされており, 40 バイトしかない. また公開鍵の a の部分もシードから再生成されることと定義されている点に注意されたい.

4.4 まとめ

基本となる McEliece 暗号方式は, McEliece により 40 年以上前に提案されており, パラメータは改訂されているものの, いまだに破られていない. Classic McEliece などのように, 公開鍵や秘密鍵は長いものの, 暗号文は短い方式が多い. LPN 問題は学習理論や符号理論から派生した問題であり, 誤り確率 τ が十分大きい場合の LPN 問題を確率的多項式時間で効率的に解くことは困難であると予想されている.

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

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

が挙げられる. 一方, 欠点として,

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

が挙げられる.

暗号方式のパラメータ設定の際には, 4.1 節で挙げたさまざまなアルゴリズムを考慮する必要がある. アルゴリズムの高速化について盛んに研究されており, 動向を注視する必要がある. また, 攻撃に用いられるアルゴリズムの研究は理論的なものが多く, 攻撃実験報告は小さいパラメータに対して行ったものが多い. そのため, 攻撃実験に関する研究もこれから非常に重要である.

公開鍵や秘密鍵を圧縮しようと特殊な符号を採用したり, 距離の定義を変える提案も多くある. これらは解読攻撃を

受けることも多く, 評価が確定していない暗号・署名方式については注視が必要である.

第 4 章の参考文献

- [1] C. Aguilar Melchor, N. Aragon, S. Bettaleb, 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. Bettaleb, 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 security. *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. Zveydinger. 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. $HB^\#$: 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*, volume 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. Leveil, 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 5 章

多変数多項式に基づく暗号技術

多変数公開鍵暗号^{*1} (Multivariate Public Key Cryptosystems) における暗号方式の特徴として、有限体上の多変数多項式を用いた連立方程式

$$\begin{cases} p_1(x_1, x_2, \dots, x_n) = 0, \\ p_2(x_1, x_2, \dots, x_n) = 0, \\ \vdots \\ p_m(x_1, x_2, \dots, x_n) = 0 \end{cases}$$

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 以上に限定できる。本報告書では、多変数公開鍵暗号の多くの方式で採用されている双極型システムを中心に解説する。

5.1 多変数多項式に基づく暗号技術の安全性の根拠となる問題

$\mathbb{F}_q$ で位数 q の有限体を表し、 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 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 $p_i(\mathbf{x})$ ($i = 1, \dots, m$) により、 $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_m(\mathbf{x}))$ 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 $P: \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ と表すことにする。

5.1.1 MP 問題 (MQ 問題)

MP 問題は次のように述べられる。

MP 問題 多変数多項式系 $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_m(\mathbf{x}))$ と $\mathbf{d} = (d_1, d_2, \dots, d_m) \in \mathbb{F}_q^m$ に対して、変数 $\mathbf{x}$ に関する連立方程式

$$\begin{cases} p_1(x_1, x_2, \dots, x_n) = d_1, \\ p_2(x_1, x_2, \dots, x_n) = d_2, \\ \vdots \\ p_m(x_1, x_2, \dots, x_n) = d_m \end{cases} \quad (5.1)$$

の解 (が存在するなら) 少なくとも 1 つ求めよ。

^{*1} かつては「多次多変数公開鍵暗号」と呼ばれていた。

連立方程式 (5.1) の右辺の各 d_i を左辺に移行して $p_i(\mathbf{x})$ に吸収させることができるので、右辺を 0 として MP 問題を表現する場合もある。MP 問題において、 $P(\mathbf{x})$ の全ての成分 $p_i(\mathbf{x})$ が 1 次以下となる場合、MP 問題は単に線形方程式を解く問題となり、ガウスの消去法などで m, n に関し多項式時間で求解することが可能である。よって、MP 問題を考える場合は通常、各 $p_i(\mathbf{x})$ の次数は 2 以上であると仮定する。特に、 $p_i(\mathbf{x})$ の次数が全て 2 となるとき、MP 問題は MQ 問題と呼ばれる。 $\mathbb{F}_q = \mathbb{F}_2$ の場合、MQ 問題は NP 完全であることが知られている [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 5.1 のようになっている。

表 5.1: Fukuoka MQ challenge で解かれた MQ 問題のパラメータの最大値 (2022/9/30 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)

5.1.2 MinRank 問題

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

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

となるものを求めよ。(Rank M は行列 M のランクを表す。)

MinRank 問題は HFEv- や Rainbow など様々な方式の安全性に関わっている。また、MinRank 問題を解く計算の困難性をベースとした署名方式などがいくつか提案されている [9, 4, 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。 $x_1, \dots, x_k$ 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

$$SC = \sum_{i=1}^k x_i M_i \quad (5.2)$$

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、(5.2) より $\mathbf{r}_j$ は C の行ベクトルが張る空間に属する。

よって、行列 $C'_j \in \mathbb{F}_q^{(r+1) \times n}$ を

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

で定めると、各 $j = 1, \dots, m$ に対して、 $\text{Rank } C'_j \leq r$ 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 $m \binom{n}{r+1}$ 個存在

する. $\#T = r$ となる $T \subset \{1, 2, \dots, n\}$ 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 $x_1, \dots, x_k$ と c_T ($T \subset \{1, \dots, m\}, \#T = r$) を用いて多項式で表すことができる. これらの変数の個数は $k + \binom{n}{r}$ である. つまり, MinRank 問題は $k + \binom{n}{r}$ 個の変数の $m \binom{n}{r+1}$ 個の方程式からなる MP 問題に帰着される.

5.1.3 IP 問題, EIP 問題

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

IP 問題 S, T をそれぞれ, $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像とする. 多変数多項式系 $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_m(\mathbf{x}))$ 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 $\tilde{P}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$ 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. 5.2 節で見るように, EIP 問題は双極型システムで構成される暗号方式, 署名方式の鍵復元攻撃に対する安全性に関わる. EIP 問題を解く方法はクラス $\mathcal{C}$ の取り方 (あるいは方式) に依存する.

5.2 多変数多項式に基づく代表的な暗号方式

5.2.1 双極型システム

IP 問題ベース [25] や MinRank 問題ベース [9, 1, 4, 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, $P(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_m(\mathbf{x}))$ が三

角型多変数多項式系である、すなわち、

$$\begin{aligned} p_1(\mathbf{x}) &= x_1, \\ p_2(\mathbf{x}) &= x_2 + g_2(x_1) \quad (g_2(x_1) \in \mathbb{F}_q[x_1]), \\ p_3(\mathbf{x}) &= x_3 + g_3(x_1, x_2) \quad (g_3(x_1, x_2) \in \mathbb{F}_q[x_1, x_2]), \\ &\vdots \\ p_m(\mathbf{x}) &= x_m + g_m(x_1, \dots, x_{m-1}) \quad (g_m(x_1, \dots, x_{m-1}) \in \mathbb{F}_q[x_1, \dots, x_{m-1}]) \end{aligned}$$

の形で表されるとき、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 $\mathcal{C}_{\text{cent}}$ を見つけ固定する。 $G(\mathbf{x}) \in \mathcal{C}_{\text{cent}}$ と $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T をそれぞれ任意にとり、これらを合成した多変数多項式系 $F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ をトラップドア付き一方向関数として利用するのが、双極型システムのアイデアである。ただし、 $F(\mathbf{x})$ が実際にトラップドア付き一方向関数となるかどうかは $\mathcal{C}_{\text{cent}}$ のとり方に依存する。

双極型システムの鍵生成は次のように行う。

鍵生成

1. $G(\mathbf{x}) \in \mathcal{C}_{\text{cent}}$ をランダムに選ぶ。
2. $\mathbb{F}_q^n, \mathbb{F}_q^m$ 上のアフィン同型写像 S, T をランダムに選ぶ。
3. $F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ とする。

このとき、公開鍵は $F(\mathbf{x})$ 、秘密鍵は $G(\mathbf{x}), S, T$ となる。 $F(\mathbf{x})$ はその係数集合が公開鍵として保管される。また、 $G(\mathbf{x})$ を (この方式の) **中心写像**とよぶ。中心写像のクラス $\mathcal{C}_{\text{cent}}$ は公開鍵長 (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 $\mathcal{C}_{\text{cent}}$ の取り方を変えることで幅広い方式の構成が可能である。例えば、

$\mathcal{C}_{\text{cent}} = \{ \text{三角型多変数多項式系} \}$ とすると暗号方式が得られる．双極型システムにおいて， $\mathcal{C}_{\text{cent}}$ に関する EIP 問題がその安全性に大きく関わってくる．実際，EIP 問題を解けた場合， $F(\mathbf{x}) = T \circ G(\mathbf{x}) \circ S$ の代わりに分解 $F(\mathbf{x}) = T' \circ G'(\mathbf{x}) \circ S'$ を用いても，暗号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， $\mathcal{C}_{\text{cent}} = \{ \text{三角型多変数多項式系} \}$ 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．

5.2.2 HFE 方式，HFEv-方式

5.2.2.1 暗号方式 HFE

$K = \mathbb{F}_{q^n}$ を $\mathbb{F}_q$ の n 次拡大体とし， $\mathbb{F}_q$ -線形同型写像 $\phi: \mathbb{F}_q^n \xrightarrow{\sim} K$ を 1 つ固定する． D を正の整数として， K 上の 1 変数多項式 $\mathcal{G}(X)$ を次のようにとる．

$$\mathcal{G}(X) = \sum_{0 \leq i \leq j}^{q^i + q^j \leq D} \alpha_{i,j} X^{q^i + q^j} + \sum_{0 \leq i}^{q^i \leq D} \beta_i X^{q^i} + \gamma \quad (\alpha_{i,j}, \beta_i, \gamma \in K)$$

$\mathcal{G}(X)$ 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 $G: \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ 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 $\mathcal{C}_{\text{HFE}}$ に対し， $\mathcal{C}_{\text{cent}} = \mathcal{C}_{\text{HFE}}$ 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．

5.2.2.2 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 $\phi: \mathbb{F}_q^n \rightarrow K$ 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．

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

ここで， $\beta_i(x_{n+1}, \dots, x_{n+v}), \gamma(x_{n+1}, \dots, x_{n+v})$ は共に $\mathbb{F}_q^v$ から K への多項式写像であり， $\beta_i(x_{n+1}, \dots, x_{n+v})$ は線形関数， $\gamma(x_{n+1}, \dots, x_{n+v})$ 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 $G = \phi^{-1} \circ \mathcal{G} \circ (\phi \times id_v): \mathbb{F}_q^{n+v} \rightarrow \mathbb{F}_q^n$ により定める． $\alpha_{i,j} \in K$ と $\beta_i(x_{n+1}, \dots, x_{n+v}), \gamma(x_{n+1}, \dots, x_{n+v})$ 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．公開鍵は通常の変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 $\sigma = F^{-1}(M)$ は以下のように計算される．

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] で効率的な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．

5.2.3 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，5.3.1 節で説明する．

正の整数 $t, v_1, o_1, \dots, o_t$ に対し， $v_{i+1} = v_i + o_i$ により， $v_2, \dots, v_{t+1}$ を順次定める．また， $i = 1, \dots, t$ に対し， $S_i = \{1, \dots, v_i\}$ ， $O_i = \{v_i + 1, \dots, v_{i+1}\}$ 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 $G(\mathbf{x}) = (g_{v_1+1}(\mathbf{x}), \dots, g_n(\mathbf{x}))$ を次の形で与える：

$$g_k(x_1, \dots, x_n) = \sum_{i \in O_h, j \in S_h} \alpha_{i,j}^{(k)} x_i x_j + \sum_{i,j \in S_h, i \leq j} \beta_{i,j}^{(k)} x_i x_j + \sum_{i \in S_{h+1}} \gamma_i^{(k)} x_i + \eta^{(k)} \quad (k = v_1+1, \dots, n).$$

但し， h は k が属する層番号，すなわち，“ $k \in O_h$ ” で定まる整数 $1 \leq h \leq t$ である． $\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in \mathbb{F}_q$ を動かしてできる $G(\mathbf{x})$ のなすクラスを $\mathcal{C}_{\text{Rainbow}}$ と定め，これを Rainbow の中心写像のクラスとする．

署名生成

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

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

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

の解を計算し，それを $b_{v_h+1}, \dots, b_{v_{h+1}}$ と置く．もし解がなければ 2 に戻る．

4. $\mathbf{b} \leftarrow (b_1, \dots, b_n)$ ．
5. $\sigma \leftarrow S^{-1}(\mathbf{b})$ ．(σ 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。

5.3 多変数多項式に基づく主要な暗号方式

MPKC で標準化が期待されるのは署名方式の UOV である。5.2.3 節で述べたように Rainbow は UOV を多層化したものであるが、その多層化が結果として暗号の強度を弱めることになった [5, 7]。しかし、多層化していない UOV は高い安全性を維持している。また、NIST の PQC の新たな署名方式の公募 [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 5.2: 多変数多項式に基づく暗号の分類

文献	暗号化	鍵交換	署名
UOV [21]			○

5.3.1 署名方式 UOV

5.3.1.1 UOV の概要

UOV [21] は、双極型システムを用いた署名方式であり、署名長が短く、検証が速いという長所を持つ。署名生成では、線形連立方程式の求解を利用しており、2 次以上の連立方程式の求解は必要としない。UOV の変種方式として、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 $G(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_m(\mathbf{x}))$ を次の形で与える。

$$g_k(\mathbf{x}) = \sum_{\substack{1 \leq i \leq v \\ v+1 \leq j \leq n}} \alpha_{i,j}^{(k)} x_i x_j + \sum_{1 \leq i \leq j \leq v} \beta_{i,j}^{(k)} x_i x_j + \sum_{1 \leq i \leq n} \gamma_i^{(k)} x_i + \eta^{(k)} \quad (k = 1, \dots, m).$$

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 $\mathbf{c} = (c_1, \dots, c_m) \in \mathbb{F}_q^m$ に対し、 $\mathbf{b} = G^{-1}(\mathbf{c})$ (の一つ) が以下のように計算できる。

1. $b_1, \dots, b_v \in \mathbb{F}_q$ をランダムにとる。
2. $g_1(\mathbf{x}), \dots, g_m(\mathbf{x})$ に $x_1 = b_1, \dots, x_v = b_v$ を代入し、 $x_{v+1}, \dots, x_n$ に関する 1 次の多項式系 $\bar{g}_1(x_{v+1}, \dots, x_n), \dots, \bar{g}_m(x_{v+1}, \dots, x_n)$ を得る。1 次方程式

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

の解を計算し、それを $b_{v+1}, \dots, b_n$ と置く。もし解がなければ 1 に戻る。

3. $\mathbf{b} = (b_1, \dots, b_n)$ 。

$\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in \mathbb{F}_q$ を動かしてできる $G(\mathbf{x})$ の集合を $\mathcal{C}_{\text{UOV}}$ としたとき, $\mathcal{C}_{\text{cent}} = \mathcal{C}_{\text{UOV}}$ 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. $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{F}_q^m$ を暗号学的ハッシュ関数とする.

鍵生成

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

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

の解を計算し, それを $b_{v+1}, \dots, b_n$ と置く. もし解がなければ **3** に戻る.

4. $\mathbf{b} \leftarrow (b_1, \dots, b_n)$.
5. $\sigma \leftarrow S^{-1}(\mathbf{b})$.

σ が署名となる. 最後に検証である.

検証

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

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

5.3.1.2 UOV のパラメータ選択

UOV の設計に必要なパラメータは, 有限体の位数 q , 方程式数 m , 変数の個数 n である. PQC Forum [28] では, 表 5.3 のように UOV のパラメータ見積もりが公開されている.

5.4 多変数多項式に基づく暗号技術に関するまとめ

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

表 5.3: 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

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ではないが、定期的に big field 方式を用いた暗号方式は提案されているので、今後の新たな方式の出現が期待できる。本稿では MQ 問題ベースの双極型システムのみを説明したが、まだ数は少ないが IP 問題ベースや MinRank 問題ベースの方式も存在する。近年、MinRank 問題の解析が進歩しているため、MinRank 問題ベースの新たな方式の出現も今後、期待できる。

第 5 章の参考文献

- [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] 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*.
- [4] E. Bellini, A. Esser, C. Sanna, J. Verbel. MR-DSS – Smaller MinRank-based (Ring-)signatures. *IACR Cryptology ePrint Archive*, 2022/973.
- [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. Solving 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 cryptosystem. *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 Vinegar 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/km50w_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 6 章

同種写像に基づく暗号技術

本章では同種写像に基づく暗号技術についてまとめる。同種写像に基づく暗号技術の安全性は、同種写像問題を解く計算の困難性及び（それと同値な）自己準同型環計算問題の困難性に依存している。

6.1 節では、安全性の根拠となる問題として、同種写像問題の一般形を述べた後、最近発見された SIDH (Supersingular Isogeny Diffie–Hellman) 同種写像問題 [13] に対する解法 [3, 27, 30] が適用できない計算問題として、自己準同型環計算問題及び SQISign (Short Quaternion and Isogeny Signature) 署名方式 [14] の安全性に関する計算問題の概要を記述していく。SIDH 同種写像問題に対する解法に関しては調査報告書を参照のこと。6.2 節では、代表的な暗号方式として、自己準同型環計算問題に基づく GPS (Galbraith–Petit–Silva) 署名 [18] を取り上げる。6.3 節では、主要な暗号方式として、GPS 署名を改良した SQISign 署名方式を解説する。

本章では、超特異楕円曲線を用いた暗号技術しか扱わない。しかし、通常楕円曲線に基づく CRS (Couveignes–Rostovtsev–Stolbunov) 鍵共有法 [9, 31] を改良した De Feo ら [15] の方式は、それ自体は実用的な性能にはまだ遠いが、調査報告書に記載されている CSIDH 鍵共有の原型を与えているという点で重要である。

同種写像の数学的詳細については、De Feo の概説記事 [12] や Washington の楕円曲線の教科書 [35] を、四元数環については Voight の教科書 [34] を参照のこと。また、Galbraith–Vercauteren による同種写像関連問題のサーベイ [19] も参照する。

■記法 $x \leftarrow_R X$ は、 x を集合 X から一様ランダムにサンプリングすることを表す。以下では、有限体上に定義された楕円曲線のみを扱い、同種写像暗号では、多くの場合、モンゴメリ型の楕円曲線定義式 $E_{a,b} : by^2 = x^3 + ax^2 + x$ が用いられる。標数 p の有限体 $\mathbb{F}$ 上定義された楕円曲線 E に対し、 O_E は E の無限遠点であり、 $\mathbb{F}$ の拡大体 $\mathbb{K}$ に対して、 $\mathbb{K}$ -有理点群は $E(\mathbb{K}) := \{(x, y) \in \mathbb{K}^2 \mid (x, y) \text{ は } E \text{ の定義式を満たす}\} \cup \{O_E\}$ で与えられる。また、 E の r -ねじれ部分群は $E[r] := \{P \in E(\overline{\mathbb{F}}_p) \mid rP = O_E\}$ で与えられる。

6.1 同種写像に基づく暗号技術の安全性の根拠となる問題

同種写像問題の一般形を述べた後、自己準同型環計算問題と SQISign 署名方式の安全性に関する計算問題の概要及びそれら問題に対する解析状況について記述していく。

6.1.1 同種写像問題の一般形

同種写像とは、2 つの楕円曲線 E, E' の間の写像 φ であり、 E の座標 (x, y) の有理式で与えられると共に、楕円曲線の加法構造に関する準同型性、即ち $\varphi(P + Q) = \varphi(P) + \varphi(Q)$ 、を有する非零写像である。（その正確な定義は、前掲の

各文献を参照のこと。) また, E, E' の間に, 同種写像 φ が存在する時に, E と E' は同種であるという.

同種写像 φ は, その核 $C = \ker(\varphi)$ によって決まるので, φ の定義域曲線 (始点曲線) E に対して φ の値域となる楕円曲線を E/C と書き表す, すなわち, $\varphi: E \rightarrow E/C$. 核 $C = \ker(\varphi)$ の位数がセキュリティパラメータ λ の多項式サイズであれば, $C = \ker(\varphi)$ となる φ を効率的に計算するアルゴリズムが Vélu によって与えられている [33]. (モンゴメリ型楕円曲線に対する Vélu の公式に関しては, [29] を参照のこと.) 特に核の位数 $\#C$ が小素数になる同種写像を同種写像基本演算として, それらの合成が同種写像暗号での基本的な暗号演算を与えることになる. そして, その合成における基本演算の組み合わせ方法が, 秘密鍵情報を与える.

つまり, 同種な楕円曲線の間の同種写像を計算することを要求する次の同種写像問題が, 具体的な暗号方式の安全性を根拠づける次節以降の諸問題の基本形となる. (超特異同種写像問題と自己準同型環計算問題との計算量的同値性に関しては 6.1.2 節で触れる.)

定義 6.1 (一般形同種写像問題 [19]) 2つの同種な楕円曲線 E, E' に対して, 同種写像 φ を計算せよ. (φ のコンパクトな表現を与えよ.)

ここで, 「 φ のコンパクトな表現」とは, 様々な表現方法が考えられる. 例えば, $\deg(\varphi)$ が小素数 ℓ_i によって $\prod_i \ell_i^{e_i}$ となっている場合には, この分解に沿って φ を分解した各 ℓ_i 次同種写像の像に現れる値域楕円曲線 (又は j 不変量) の列挙で与えることができる.

定義 6.1 において, φ の次数が多項式サイズであれば, 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 [19] 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 [22] や Galbraith による同種写像問題に関する研究 [17] 及び Couveignes と Rostovtsev–Stolbunov による初期の暗号応用への提案 [9, 31] がある. その後, Charles らによる同種写像に基づいたハッシュ関数の提案 [5] 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キスパンダーグラフであることに着目して暗号に応用した意義は大きい.

■超特異同種写像問題と通常同種写像問題 標数 p の有限体上の楕円曲線 E の p -ねじれ部分群 $E[p]$ が, $E[p] = \{O_E\}$ の時, E を超特異楕円曲線といい, そうでない時, E を通常楕円曲線という. 超特異楕円曲線の j 不変量は, $\mathbb{F}_{p^2}$ の要素である. つまり, 超特異 j 不変量の個数は, 有限個であり, 具体的に $p/12 + \epsilon$ (但し $\epsilon \in \{0, 1, 2\}$) で与えられる. 超特異, 通常という楕円曲線の性質は, 同種写像によって保存されるため, 同種写像問題も, この2つの性質によって, 超特異同種写像問題と通常同種写像問題という2つの問題に分類される.

■超特異同種写像問題の計算困難性 超特異同種写像問題の計算困難性を評価することは重要である. また, 自己準同型環計算問題との関係性については 6.1.2 節を参照のこと.

超特異同種写像問題の古典計算機による解読時間は $\tilde{O}(\sqrt{p})$, 量子計算機による解読時間は $\tilde{O}(\sqrt[3]{p})$ と見積もられている. 古典解読アルゴリズムは Galbraith [17] による中間一致攻撃で, 解読時間は $\tilde{O}(\sqrt{p})$ であり, 量子解読アルゴリズムは Biasse ら [1] によって時間計算量が $\tilde{O}(\sqrt[3]{p})$ 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 [4] と Grover アルゴリズムに基づく $\tilde{O}(\sqrt[3]{p})$ の道探索アルゴリズムを結合したものである.

また, Costello ら [6], Longa ら [26] による報告, Udovenko–Vitto [32] による \$IKEp182 Challenge [8] 解読報告,

Jaques-Schanck [21] による同種写像問題に対する（量子）安全性評価報告は、いずれも SIDH 鍵共有（及び SIKE 暗号化 [20]）法への攻撃として提案されているが、多くの部分は一般的な超特異同種写像問題に関する知見としても有効であることに注意する。

6.1.2 自己準同型環計算問題と SQISign 署名方式の安全性に関する計算問題

6.1.2.1 自己準同型環計算問題

同種写像暗号は、Kohel [22], Galbraith [17], Couveignes [9] らの先駆的研究にその起源をもつが、特に、Kohel は有限体上の楕円曲線の自己準同型環を計算するアルゴリズムを探索しており、そのために楕円曲線の同種写像からなる「同種写像グラフ」の性質を見極めることから始めて、目的とする自己準同型環計算を同種写像グラフ上のアルゴリズム構成に帰着していく。その後、Kohel-Lauter-Petit-Tignol [23] は、この「同種写像計算」と「自己準同型環計算」を並置しながら考察する視点を、「構成的 Deuring 対応」として計算論的観点から捉え直した（表 6.1 参照）。そこでは、四元数環側での ℓ -同種写像探索問題を解く 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 [10, 11, 36]、現在、自己準同型環計算問題の困難性に基づいた暗号構成の研究が進められている [18, 14, 16]。

■自己準同型環計算問題とその超特異同種写像計算問題との同値性 以下の記述に関しては、例えば [24, 25] を参照する。有理数体 $\mathbb{Q}$ 上 $\{1, i, j, k\}$ を基底とするベクトル空間でありかつ $a, b \in \mathbb{Q}$ により $i^2 = a, j^2 = b, k = ij = -ji$ という積構造が入った $\mathbb{Q}$ 上の代数（環）を四元数環 $\mathcal{B}$ と呼ぶ。各素点 ν （素数または ∞ ）における $\mathbb{Q}$ の完備化 $\mathbb{Q}_\nu$ による $\mathcal{B} \otimes \mathbb{Q}_\nu$ が $\nu = p, \infty$ の時にのみ斜体（可除環）になる四元数環 $\mathcal{B} = \mathcal{B}_{p, \infty}$ を扱う。これを、 $\mathcal{B}_{p, \infty}$ は p, ∞ の 2 点のみで分岐する四元数環であるといい、 $\mathcal{B}_{p, \infty}$ は同型を除いて一意に決まる。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 $\text{End}(E)$ は E の自己準同型環と呼ばれて、 $\text{End}(E)$ は $\mathcal{B}_{p, \infty}$ の極大整環 $\mathcal{O}$ になっている^{*1}。ここで、（四元数環の）整環とは $\mathbb{Z}$ 上階数 4 の加群でありかつ環であるものであり、極大整環とは、そのような整環の中で包含関係に関して極大になっているものを指す。この自己準同型環 $\text{End}(E)$ を計算する以下の問題が基本である。

定義 6.2 (自己準同型環計算問題 [22]) 超特異楕円曲線 E が与えられて、 E の自己準同型環 $\text{End}(E)$ を計算せよ。

Eisenträger らの研究 [10, 11] により、超特異同種写像計算問題と（超特異）自己準同型環計算問題の間に多項式時間帰着による計算問題としての同値性が示された。そこではヒューリスティックな仮定が使われていたが、Wesolowski [36] は、一般化されたリーマン予想に基づいて、その同値性に対して厳密な証明を与えた。

6.1.1 節で、超特異同種写像問題の古典計算機による現在最速の解読時間は $\tilde{O}(\sqrt{p})$ と見積もられていたもので、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、[11] において、 $\tilde{O}(\sqrt{p})$ 時間の自己準同型環計算（古典）アルゴリズムが報告されている。

■Deuring 対応 自己準同型環計算問題で与えられる楕円曲線 E から極大整環 $\mathcal{O}$ への対応は、表 6.1 に掲げたように、楕円曲線に関する様々な概念から四元数環に関する概念への対応に拡張される。その詳細に関しては、例えば [25, 第 2 章] を参照していただきたいが、特に基本的な対応としては、同種写像 $\varphi: E \rightarrow E_1$ が、極大整環の間の同型 $\mathcal{O} \cong \text{End}(E), \mathcal{O}_1 \cong \text{End}(E_1)$ 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 $\varphi: E \rightarrow E_1$ 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

^{*1} 自己準同型写像は英語で endomorphism であるので、その全体を $\text{End}(E)$ で表す。

り、超特異 j 不変量 ($\in \mathbb{F}_{p^2}$) の集合がイデアル類集合 $\text{cl}(\mathcal{O})$ と一対一に対応していることもわかる。

一般に表 6.1 に示されるように、幾何的な情報から成る楕円曲線側のデータと代数的な情報から成る四元数環側のデータの間に対応関係が存在しており、Deuring 対応と呼ばれる。自己準同型環計算問題 (定義 6.2) は Deuring 対応に基づいた問題であり、楕円曲線側の超特異 j 不変量 $j(E)$ から対応する四元数環側の極大整環 $\mathcal{O} = \text{End}(E)$ を計算する問題となっている。そして、この Deuring 対応は、6.2.1 節及び 6.3.1 節での暗号構成を理解する際にも重要な鍵となっている。

表 6.1: Deuring 対応

楕円曲線側	四元数環側
超特異 j 不変量 $j(E) \in \mathbb{F}_{p^2}$ (の $\mathbb{F}_{p^2}/\mathbb{F}_p$ -Galois 共役類)	$\mathcal{B}_{p,\infty}$ 内の極大整環 $\mathcal{O} = \text{End}(E)$ の自己同型類 (タイプ)
同種写像 $\varphi: E \rightarrow E_1$ 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 $\theta \in \text{End}(E)$	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 $\varphi: E \rightarrow E_1, \psi: E \rightarrow E_1$	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 $\text{cl}(\mathcal{O})$
同種写像の合成 $\tau \circ \rho: E \rightarrow E_1 \rightarrow E_2$	イデアル積 $I_{\tau \circ \rho} = I_\rho \cdot I_\tau$
N -同種写像の同型類	レベル N の Eichler 整環の類集合

6.1.2.2 SQISign 署名の安全性に関する計算問題

次に、SQISign 署名の安全性を示すために必要な計算問題を述べる。

■SQISign 署名の健全性に関する計算問題 まずは、SQISign 署名の健全性 (偽造不可能性) を示すための計算問題である超特異平滑自己準同型写像計算問題 (SEP: Smooth Endomorphism Problem) を定義する。以下では、核が巡回群となる自己準同型写像を巡回自己準同型写像と呼ぶ。

定義 6.3 (超特異平滑自己準同型写像計算問題 [14, 25]) 超特異楕円曲線 E が与えられて、平滑な整数を次数にもつ E 上の (非自明な) 巡回自己準同型写像を見つけよ。

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、[11] で見るように、自己準同型環 $\text{End}(E)$ 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 [14]。よって、 $\tilde{O}(\sqrt{p})$ 時間での古典アルゴリズム [11] 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 $\mathcal{O}_0 = \text{End}(E_0)$ 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 $\text{End}(E_0) = \mathbb{Z} + \mathbb{Z}\iota + \mathbb{Z}\frac{\iota+\pi}{2} + \mathbb{Z}\frac{1+\iota\pi}{2}$ 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 $\mathfrak{D} \subset \mathcal{O}_0 \cap \mathbb{Q}[i]$ による $\mathfrak{D} + j\mathfrak{D} \subset \mathcal{O}_0$ が部分整環であり $\mathfrak{D} \subset (j\mathfrak{D})^\perp$ と直交分解しているとき^{*2}, 特殊極值的 (special extremal) であるという. 詳細は [14, 25] を参照. $p = 7 \bmod 12$ の時, 上述の E_0 に対して $\text{End}(E_0)$ は特殊極值的であり, この時, E_0 は特殊極值的の曲線と呼ばれる. 特殊極值的の曲線 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_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}. [14, 16] において定義された E_0 を始点とする同種写像から成るある集合 $\mathcal{P}_{N_\tau}$ を τ によって E_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_A を始点とした 2 べき次数 $D (= 2^e)$ の巡回同種写像全体 $\text{Iso}_{D,j(E_A)}$ 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

$$\begin{array}{ccc} E_0 & \xrightarrow{\psi} & E_1 \\ \tau \downarrow & & \downarrow \varphi \\ E_A & \xrightarrow{\sigma} & E_2 \end{array}$$

CSI-FiSh, GPS 図式と同様に可換図式ではない.

定義 6.4 (SQISign 署名のランダム識別問題 [14, 25]) $\tau : E_0 \rightarrow E_A$ 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_{sqisign} (詳しくは 6.3.1 節参照) と公開鍵 E_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 $\mathcal{O}_\tau$ への多項式回のアクセスが許される時に, E_A を始点とする同種写像 σ が与えられて σ が $\text{Iso}_{D,j(E_A)}$ 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_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 [14, 25]. つまり, $\tilde{O}(\sqrt{p})$ 時間を必要とすると見積もられている.

また, 上述の 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, 調査報告書に記載されている最近の SIDH 同種写像問題に対する攻撃法が適用できないことに注意する.

6.2 同種写像に基づく代表的な暗号方式

6.2.1 GPS 署名

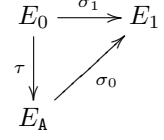
Galbraith–Petit–Silva (GPS) [18] によって始めて自己準同型環の知識証明に基づく署名方式が提案された. GPS 署名は実際に利用するのは困難であろうと思われるが, 現在, GPS 署名は, SQISign 署名の原型を与えているという点で重要である. 6.1.2 節で述べた Deuring 対応と KLPT アルゴリズム [23] が GPS 署名の理論的基礎を与える.

^{*2} $\mathcal{B}_{p,\infty}$ における内積は $\alpha, \beta \in \mathcal{B}_{p,\infty}$ に対して $\frac{1}{2} \text{tr}(\alpha\bar{\beta})$ 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 [25] を参照のこと).

^{*5} $\hat{\tau}$ は τ の双対同種写像である. 表 6.1 も参照のこと.

右図において E_0 は 6.1.2.2 節で与えた $j(E_0) = 1728$ なる楕円曲線（特殊極値の楕円曲線）であり，そこで見たようにその E_0 に関しては $\text{End}(E_0)$ 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 $\tau: E_0 \rightarrow E_A$ を知っている証明者（署名生成者）は， E_A から別の楕円曲線 E_1 への同種写像 $\sigma_0: E_A \rightarrow E_1$ と τ との合成 $\sigma_0 \circ \tau: E_0 \rightarrow E_1$ 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 $\sigma_1: E_0 \rightarrow E_1$ 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 $\text{End}(E_A)$ を計算する問題の困難性に基づけば，このようなランダム化ができるのは， τ を知っている証明者に限られるので，チャレンジ bit $c \in \{0, 1\}$ を送って証明者に同種写像 σ_c を答えさせることにより， τ に関する知識の有無を検査することができて，認証・署名方式が構成できる．それが GPS 認証方式，そしてその Fiat-Shamir 変換署名が GPS 署名である．ここでは [18, 第 4 章] と [25, 5.1.2 節] に基づいて GPS 署名を記述する．また，[18, 第 4 章] では，通常の Fiat-Shamir 変換を施した署名方式と Unruh 変換を施した署名方式の 2 方式が記述されているが，ここでは記述の簡便さを考慮して前者の記述を基にして以下に署名方式を与える．

鍵生成： 既知の特殊極値的自己準同型環 $\mathcal{O}_0$ をもつ超特異楕円曲線 E_0 ，互いに素な B -べき平滑数 S_1, S_2 ^{*6} を， S_1, S_2 次の同種写像グラフ上ランダムウォークが急攪拌性定理により一様分布を導く程度に十分大きくとる．セキュリティパラメータ λ に対して $t := \lambda$ （または $t := 2\lambda$ ）として， t bits 出力のハッシュ関数 H を選ぶ． $pp_{\text{gps}} := (E_0, S_1, S_2, H)$ 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 $\tau: E_0 \rightarrow E_A$ を計算して， pp_{gps} と E_A を公開鍵として， τ を秘密鍵とする．

署名生成： 各 $i = 1, \dots, t$ に関して E_A を始点とする S_2 次のランダムな同種写像 $\sigma_{0,i}: E_A \rightarrow E_{1,i}$ を計算する．署名対象メッセージ msg に対してチャレンジ bit 列 $h := b_1 \parallel \dots \parallel b_t := H(j(E_{1,1}), \dots, j(E_{1,t}), \text{msg}) \in \{0, 1\}^t$ をハッシュ関数 H で計算する．各 $i = 1, \dots, t$ 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 $\sigma_{1,i}: E_0 \rightarrow E_{1,i}$ を計算する．署名を $\sigma := (h, \sigma_{b_1,1}, \dots, \sigma_{b_t,t})$ とする．

署名検証： 公開鍵 (pp_{gps}, E_A) ，メッセージ msg と署名 $\sigma = (h, \sigma_1, \dots, \sigma_t)$ を入力として，各 $i = 1, \dots, t$ 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 $H(j(E_{1,1}), \dots, j(E_{1,t}), \text{msg})$ 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 $i = 1, \dots, t$ に対して検証が成功すれば受理を出力して，そうでなければ，不受理を出力する．

GPS 署名は，超特異楕円曲線同種写像計算問題またはそれと同値な自己準同型環計算問題（定義 6.2）の困難性を仮定すればランダムオラクルモデルの下で EUF-CMA 安全であることが示されている [18, 定理 10]．GPS 署名では，1 bit のチャレンジを用いた Σ -プロトコルに基づいているため，署名サイズが大きくなるのが欠点である．また，署名生成で使われた KLPT アルゴリズムの計算時間改善も課題であった [25, 5.1.2 節]．以上，GPS 署名には (1) 署名サイズ 及び (2) KLPT アルゴリズム計算時間 に関する 2 つの課題が存在する．

6.3 同種写像に基づく主要な暗号方式

本節では，公開鍵と署名サイズが小さいことを特長にもつ SQISign 署名について述べる（表 6.2 参照）．

^{*6} S_k ($k = 1, 2$) が B -べき平滑数 (powersmooth number) とは， S_k が $\ell_{k,i}^{e_{k,i}} < B$ なる $\ell_{k,i}^{e_{k,i}}$ の積で表される (i.e., $S_k = \prod_i \ell_{k,i}^{e_{k,i}}$) ことである．

表 6.2: 同種写像に基づく暗号の分類

文献	暗号化	鍵交換	署名
SQISign [14]			○

6.3.1 SQISign 署名

以下、自己準同型環計算問題（定義 6.2）の困難性に安全性の根拠を置く SQISign 署名を概説する。SQISign 署名は公開鍵と署名を合わせたサイズが小さい方式として注目されている。6.2.1 節で述べた GPS 署名を基にして改良を加えた署名方式が SQISign 署名であり、ASIACRYPT 2020 で De Feo–Kohel–Leroux–Petit–Wesolowski [14] により提案された。6.2.1 節末尾に付した GPS 署名の 2 つの課題を克服している。チャレンジ空間に同種写像の空間を用いることで、そのサイズをセキュリティパラメータ λ まで大きくして、 Σ -プロトコルを 1 度適用するだけで十分な Fiat–Shamir 署名構成とした。これで署名サイズが格段に小さくなった。また、GPS 署名生成においては、表 6.1 の Deuring 対応に基づいて、同種写像のイデアル表現（表 6.1 の四元数環側）をねじれ点を使った表現（表 6.1 の楕円曲線側）に変換する部分で時間が費やされていたが、SQISign 署名ではその処理を速度改善したサブルーチン（IdealTolsogeny）に置き換えるのに成功して演算効率も大きく改善した [14, 16]。

また、安全性に関しては、健全性は超特異平滑自己準同型写像計算問題（定義 6.3）の困難性に基づき、零知識性は定義 6.4 で述べた SQISign 署名のランダム識別問題の困難性に基づいている。初期提案 [14] では、ノルム方程式を解くサブルーチンに不備があり、生成される署名同種写像 σ に偏りが生じていたことが [16] において指摘された。そして、更に [16] でその不備を除去したアルゴリズム提案が行われた。

具体的なパラメータ、特に適切な SQISign 素数（SQISign-friendly prime） p を生成する問題は非自明であり、初期提案 [14] から始まり、[7, 16, 2] と現在も進行中の研究テーマである。その現状報告を後ほど行う。また、小貫 [28] により、SQISign 鍵生成で得られる鍵の分布の理論・実験による解析がなされており、[14] 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_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

$$\begin{array}{ccc} E_0 & \xrightarrow{\psi} & E_1 \\ \tau \downarrow & & \downarrow \varphi \\ E_A & \xrightarrow{\sigma} & E_2 \end{array}$$

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, $pp_{\text{sqisign}} := (E_0, D_c, D)$ 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 $\tau: E_0 \rightarrow E_A$ を計算して, pp_{sqisign} と E_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 $\psi: E_0 \rightarrow E_1$ を計算。署名対象メッセージ msg に対してハッシュ

関数 H で計算した $H(j(E_1), \text{msg})$ から決まる D_c 次の巡回同種写像 $\varphi : E_1 \rightarrow E_2$ を計算. 同種写像の合成 $\varphi \circ \psi \circ \hat{\tau} : E_A \rightarrow E_2$ 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 $\sigma : E_A \rightarrow E_2$ を計算. (E_1, E_2, σ) を msg の署名として出力.

署名検証: 公開鍵 $(pp_{\text{sqisign}}, E_A)$, メッセージ 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 $\varphi := H(j(E_1), \text{msg})$ を計算する. σ が E_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_A から E_1 への巡回同種写像であることを検証して, 共に成立すれば受理を出力して, そうでなければ, 不受理を出力する.

既に述べたように, SQISign 署名の安全性は, 超特異平滑自己準同型写像計算問題 (定義 6.3) の困難性と, 定義 6.4 で述べた SQISign 署名 σ のランダム識別問題の困難性にに基づいている.

■**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 $2^f \cdot T \mid p^2 - 1$ が満たされる素数 p (SQISign 素数) を生成することが必要である. 具体的には, ある B に対して B -平滑な T , $T \approx p^{5/4+\epsilon}$ ([2] では例えば $0.02 < \epsilon < 0.1$ とする) に対して $2^f \cdot T \mid p^2 - 1$ となる素数 p を探索する必要がある. SQISign 素数の選択基準として, 署名検証の効率化には f をできるだけ大きくして, 署名生成の効率性にとっては $\sqrt{B}/f$ をできるだけ小さくするのが望ましい [16].

NIST 安全性レベル 1: SQISign 素数は, NIST 安全性レベル 1 については, [16] において, XGCD アルゴリズムに基づいて, $B = 3923$ に関して B -平滑な T を持つ以下の 254 bits 素数 p が生成された. $f = 66$ であり, T は式 (6.1) で灰色でない B 以下の奇素因数の積で与えられる. T に関しては他の SQISign 素数についても同様である. また, [14, 25] によればレベル 1 パラメータでは署名 σ の次数 $D = 2^e$ の指数 e が $e = 1000$ で与えられている.

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

同じく NIST 安全性レベル 1 で最良の SQISign 素数として, [2] において, $B = 523$ に関して B -平滑な T を持つ以下の 254 bits 素数 $p = 2r^3 - 1$ ($r = 20461449125500374748856320$) が生成されている. $f = 47$ である.

$$\begin{aligned} p + 1 &= 2^{46} \cdot 5^3 \cdot 13^3 \cdot 31^3 \cdot 73^3 \cdot 83^3 \cdot 103^3 \cdot 107^3 \cdot 137^3 \cdot 239^3 \cdot 271^3 \cdot 523^3, \\ p - 1 &= 2 \cdot 3^3 \cdot 7 \cdot 11^2 \cdot 17^2 \cdot 19 \cdot 101 \cdot 127 \cdot 149 \cdot 157 \cdot 167 \cdot 173 \cdot 199 \cdot 229 \cdot 337 \cdot 457 \cdot 479 \cdot \\ &\quad 141067 \cdot 3428098456843 \cdot 4840475945318614791658621. \end{aligned} \quad (6.2)$$

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

$$\begin{aligned} p + 1 &= 2^{79} \cdot 3^6 \cdot 23^{12} \cdot 107^6 \cdot 127^6 \cdot 307^6 \cdot 401^6 \cdot 547^6, \\ p - 1 &= 2 \cdot 5^2 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 47 \cdot 71 \cdot 79 \cdot 109 \cdot 149 \cdot 229 \cdot 269 \cdot 283 \cdot 349 \cdot 449 \cdot 463 \cdot 1019 \cdot 1033 \cdot 1657 \cdot 2179 \cdot \\ &\quad 2293 \cdot 4099 \cdot 5119 \cdot 10243 \cdot 381343 \cdot 19115518067 \cdot 740881808972441233 \cdot 83232143791482135163921. \end{aligned} \quad (6.3)$$

NIST 安全性レベル 5: また, レベル 5 についても, [2] において最良の SQISign 素数として, $B = 150151$ に関して B -平滑な T を持つ以下の 508 bits 素数 $p = 2r^6 - 1$ ($r = 26697973900446483680608256$) が生成されている. $f = 86$

である。

$$\begin{aligned} p+1 &= 2^{85} \cdot 17^{12} \cdot 37^6 \cdot 59^6 \cdot 97^6 \cdot 233^6 \cdot 311^{12} \cdot 911^6 \cdot 1297^6, \\ p-1 &= 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11^2 \cdot 23^2 \cdot 29 \cdot 127 \cdot 163 \cdot 173 \cdot 191 \cdot 193 \cdot 211 \cdot 277 \cdot 347 \cdot 617 \cdot 661 \cdot 761 \cdot 1039 \cdot 4637 \cdot \\ &\quad 5821 \cdot 15649 \cdot 19139 \cdot 143443 \cdot 150151 \cdot 3813769 \cdot 358244059 \cdot 992456937347 \cdot \\ &\quad 353240481781965369823897507 \cdot 8601020069514574401371658891403021. \end{aligned} \quad (6.4)$$

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

データサイズ: 式 (6.1), (6.2) の NIST レベル 1 パラメータに関して, 秘密鍵サイズ 16 Bytes, 公開鍵サイズ 64 Bytes, 署名サイズ 204 Bytes が報告されている [14, 16]。

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

6.4 同種写像に基づく暗号技術に関するまとめ

本章では, 同種写像に基づいた暗号技術をまとめた。特に, SQISign 署名に関して, 関連する GPS 署名も含めた方式記述と安全性研究についてまとめた。

[9] によると, Couveignes は, 1997 年の École Normale Supérieure でのセミナーで既に同種写像に基づく暗号技術を提案しており, ほぼ同時期に Kohel [22] や Galbraith [17] も, 同種写像問題に関する研究を始めていた。つまり, 同種写像暗号技術の研究は既に 25 年の歴史をもつ。そして, 最近になり, 耐量子計算機暗号の必要性が高まることで, 同種写像暗号技術は注目されて研究が進み, NIST 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 [3] を始めとする一連の攻撃法 [27, 30] は SIDH 鍵共有に対して決定的な結果をもたらした。

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

現在, 特に, 公開鍵と署名を合わせたサイズが短い SQISign 署名が注目されていると共に, 調査報告書に記載されている CSIDH ベースの一方性群作用に関する研究も注目されており, 種々の暗号プロトコルへの応用も視野に入れた研究も進んでいる。それらも含めて, 今後, 特に注意すべきこと数点について以下にまとめておく。

- SQISign 署名は, 公開鍵と署名のサイズの小ささ, 補助点なしの署名構成, そして短署名に対する強い社会的ニーズなどを踏まえると, 現在有望な同種写像暗号技術と思われる。その一方, 零知識性に関する計算問題 (定義 6.4) の安全性検討などに関して, まだ安全性評価が不十分であり, その安全性評価は今後の重要な課題の一つである。さらに, 今後, 標準化などを考慮するのであれば, 実装研究を進める必要があり, 特にさまざまなプラットフォームでの実装結果を蓄えていく必要もある。
- 調査報告書に記載されている CSIDH ベース署名である SeaSign 署名と CSI-FiSh 署名についても, 今後の耐量子計算機署名として研究が進められているが, 安全性評価が定まった実用的な署名を得るためには, まだ今後の安全性・実装研究が必要である。そして, リング署名・グループ署名などの高機能暗号系への応用研究も重要であり, 今後の研究動向に注目する必要がある。

- (一般的な) 超特異同種写像問題及びそれと同値な自己準同型環計算問題に関しては、これまで主に、SIKE 暗号化パラメータに関して、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 6 章の参考文献

- [1] 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.
- [2] 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.
- [3] W. Castryck, T. Decru. An efficient key recovery attack on SIDH (preliminary version). *IACR Cryptology ePrint Archive*, 2022/975.
- [4] 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.
- [5] D. Charles, K. Lauter, E. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, volume 22, number 1, pp. 93–113, Springer, 2009.
- [6] 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.
- [7] 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.
- [8] C. Costello. The case for SIKE: A decade of the supersingular isogeny problem. *IACR Cryptology ePrint Archive*, 2021/543.
- [9] J. Couveignes. Hard homogeneous spaces. *IACR Cryptology ePrint Archive*, 2006/291.
- [10] 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.
- [11] K. Eisenträger, S. Hallgren, C. Leonardi, T. Morrison, J. Park. Computing endomorphism rings of supersingular elliptic curves and connections to pathfinding in isogeny graphs. *ANTS-XIV*, volume 4 of *The Open Book Series*, pp. 215–232, Mathematical Sciences Publishers, 2020.
- [12] L. De Feo. Mathematics of isogeny based cryptography. *arXiv*: 1711.04062, 2017.
- [13] L. De Feo, D. Jao, J. Plüt. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, volume 8, number 3, pp. 209–247, De Gruyter, 2014.
- [14] 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.
- [15] 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.
- [16] L. D. Feo, A. Leroux, B. Wesolowski. New algorithms for the Deuring correspondence: SQISign twice as

fast. *IACR Cryptology ePrint Archive*, 2022/234.

- [17] S. Galbraith. Constructing isogenies between elliptic curves over finite fields. *Journal of Computational Mathematics*, volume 2, pp. 118–138, Global Science Press, 1999.
- [18] 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.
- [19] S. D. Galbraith, F. Vercauteren. Computational problems in supersingular elliptic curve isogenies. *Quantum Information Processing*, volume 17, number 10, article 265, Springer, 2018.
- [20] 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 閱覽)
- [21] 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.
- [22] D. Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California at Berkeley, 1996.
- [23] D. Kohel, K. Lauter, C. Petit, 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.
- [24] A. Leroux. A new isogeny representation and applications to cryptography. *ASIACRYPT 2022*, Part II, volume 13792 of LNCS, pp. 3–35, Springer, 2022.
- [25] A. Leroux. *Quaternion algebras and isogeny-based cryptography*. PhD thesis, Ecole Polytechnique, 2022.
- [26] 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.
- [27] L. Maino, C. Martindale. An attack on SIDH with arbitrary starting curve (draft). *IACR Cryptology ePrint Archive*, 2022/1026.
- [28] H. Onuki. On the key generation in SQISign. *Number-Theoretic Methods in Cryptology (NutMic 2021)*.
- [29] J. Renes. Computing isogenies between Montgomery curves using the action of $(0, 0)$. *PQCrypto 2018*, volume 10786 of LNCS, pp. 229–247, Springer, 2018.
- [30] D. Robert. Breaking SIDH in polynomial time. *IACR Cryptology ePrint Archive*, 2022/1038.
- [31] A. Rostovtsev, A. Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptology ePrint Archive*, 2006/1450.
- [32] A. Udovenko, G. Vitto. Breaking the \$SIKEp182 challenge. *IACR Cryptology ePrint Archive*, 2021/1421.
- [33] J. Vélú. Isogénies entre courbes elliptiques. *C.R. Acad. Sc. Paris, Séries A.*, volume 273, pp. 238–241, 1971.
- [34] J. Voight. *Quaternion algebras*, Springer, June 2022.
- [35] L. Washington. *Elliptic Curves: Number Theory and Cryptography*, CRC Press, 2nd edition, 2008.
- [36] B. Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. *FOCS 2021*, pp. 1100–1111, IEEE, 2022.

第 7 章

ハッシュ関数に基づく署名技術

本章ではハッシュ関数に基づく署名技術についてまとめる。ハッシュ関数に基づく署名技術の安全性はハッシュ関数の第二原像攻撃に対する安全性に依存している。

ハッシュ関数に基づく署名技術は、最初に Lamport により one-time signature として提案された [10, 22]。また、この方式を改良した Winternitz one-time signature が Merkle [27] 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 [26, 27] により述べられている。

7.1 ハッシュ関数に基づく署名技術の安全性の根拠となる問題

ハッシュ関数は任意長あるいは実用上十分な長さ以下の入力 $\{0, 1\}$ 系列に対して固定長の $\{0, 1\}$ 系列を出力する関数である。ハッシュ関数を $H : \mathcal{D} \rightarrow \mathcal{R}$ 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 [14] 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 [30], SHA-3 [31], Haraka [21] のうちのいくつかのハッシュ関数を用いることが想定されている。これらのうち、SHA-2, SHA-3 は米国 NIST の指定する標準ハッシュ関数族である。

SHA-2 は Merkle-Damgård 構造 [9, 28] を有するハッシュ関数の族であり、Secure Hash Standard [30] のうち、SHA-1 を除く SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 からなる。SHA-2 の各ハッシュ関数の名称の末尾の数値は出力の bit 長を表す。SHA-3 は置換を用いたスポンジ構造 [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。SHAKE128, SHAKE256 については、出力長は任意に設定できる。

Haraka はブロック暗号 AES [12, 29] に基づく置換を用いた Davies-Meyer 構造 [33] に基づくハッシュ関数の族であり、Haraka-256, Haraka-512 からなる。Haraka-256, Haraka-512 の末尾の数値は入力の bit 長を表す。出力長はいずれも 256 bits である。Haraka は、ハッシュ関数に基づく署名技術での使用を想定し、短い入力に対して高い処理性

能を達成するよう設計されている。

本章で使用する記号・用語を以下にまとめる。

- $\{0, 1\}$ 系列 α, β の接続を $\alpha\|\beta$ と表記する。
- $\ll$ は左論理シフトを表す。
- 整数 ν について $[\nu]_l$ は ν の長さ l bits の 2 進数表記を表す。
- $\mathbb{B} := \{0, 1\}^8$ とする。

7.2 ハッシュ関数に基づく代表的な署名方式

7.2.1 Winternitz One-Time Signature

Winternitz one-time signature [27] 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 $N_{\ell_m-1}N_{\ell_m-2}\cdots N_1N_0$ で表記されたとする。このとき、 $0 \leq k \leq \ell_m - 1$ について $N_k \in \{0, 1, \dots, b-1\}$ であり、 $N = \sum_{k=0}^{\ell_m-1} N_k 2^k$ である。さらに、 N のチェックサムを $C := \sum_{k=0}^{\ell_m-1} (b-1-N_k)$ と定義する。 C が ℓ_c 桁の b 進数 $N_{\ell_m+\ell_c-1}N_{\ell_m+\ell_c-2}\cdots N_{\ell_m+1}N_{\ell_m}$ で表記されたとする。 $\ell := \ell_m + \ell_c$ とする。

■鍵生成アルゴリズム 秘密鍵 $(x_0, x_1, \dots, x_{\ell-1})$ 、公開鍵 $(pub_0, pub_1, \dots, pub_{\ell-1})$ は以下のように生成される。

1. $x_0, x_1, \dots, x_{\ell-1} \in \mathcal{D}$ を無作為に選択する。
2. $0 \leq k \leq \ell - 1$ について $pub_k := H^{b-1}(x_k) := \underbrace{H(H(\cdots(H(x_k))\cdots))}_{b-1 \text{ times}}$ 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 $(s_0, s_1, \dots, s_{\ell-1})$ は以下のように生成される。

1. $0 \leq k \leq \ell - 1$ について $s_k := H^{N_k}(x_k)$ とする。

■検証アルゴリズム メッセージのハッシュ値 N とその署名 $(s_0, s_1, \dots, s_{\ell-1})$ の検証は以下のように行われる。

1. $0 \leq k \leq \ell - 1$ について $pub_k = H^{b-1-N_k}(s_k)$ かつそのときに限り、 $(s_0, s_1, \dots, s_{\ell-1})$ 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 $s_0, s_1, \dots, s_{\ell_m-1}$ が得られたとき、 $0 \leq k \leq \ell_m - 1$ について $N'_k \geq N_k$ を満たす N' について、 $s'_k := H^{N'_k-N_k}(s_k)$ によって、署名 $(s'_0, s'_1, \dots, s'_{\ell_m-1})$ 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 [24] によりまとめられている。

7.2.2 マークル木を用いた署名方式

1 回署名方式を用いて複数のメッセージに署名を行う場合、メッセージの個数と同じ個数の公開鍵と秘密鍵の組が必要となる。マークル木を用いることにより、このような複数回署名方式の公開鍵の大きさを削減できる [26]。

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 $pk_0, pk_1, \dots, pk_{2^h-1}$ とする。このとき、高さが h 、すな

わち、葉の個数が 2^h のマークル木は以下のように構成される．高さ $j (\geq 0)$ の左から $i (\geq 0)$ 番目の節点を $v_{i,j}$ と表記する． $v_{i,j}$ は以下のように計算される．

1. $0 \leq i \leq 2^h - 1$ について、 $v_{i,0} := H(pk_i)$ とする．
2. $1 \leq j \leq h$ に対し、 $0 \leq i \leq 2^{h-j} - 1$ について、 $v_{i,j} := H(v_{2i,j-1} \| v_{2i+1,j-1})$ とする．

この署名方式の公開鍵は $v_{0,h}$ である．秘密鍵は 1 回署名の公開鍵 $pk_0, pk_1, \dots, pk_{2^h-1}$ に対応するすべての秘密鍵である． i 個目のメッセージの署名を検証するためには、 $v_{0,h}$ を用いて pk_i が正しいことを検証する必要がある．このために、 i 個目のメッセージの署名には、マークル木の $v_{i,0}$ から $v_{0,h}$ に至る経路上の各節点の、経路上にない子節点が含まれる．これらの節点の列は認証パスと呼ばれる．

7.2.3 マークル木の階層構造による署名方式

前節で述べた一つのマークル木を用いた署名方式では、鍵生成時にすべての 1 回署名の公開鍵と秘密鍵を生成する必要がある．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 [16]．

この署名方式のマークル木の階層構造の階層数を L とする．この署名方式では、 $0 \leq i \leq L-1$ について、第 i 層のマークル木の高さはすべて等しく h_i であると仮定する．このとき、この署名方式は $2^{\sum_{i=0}^{L-1} h_i}$ 個のメッセージに署名できる．

この署名方式で、 $0 \leq i \leq L-1$ について、第 i 層のマークル木は $2^{\sum_{j=0}^{i-1} h_j}$ 個存在する．第 0 層（最上層）のマークル木は 1 個であり、その根がこの署名方式の公開鍵となる．したがって、この公開鍵を生成する際には、1 回署名の公開鍵と秘密鍵の組を 2^{h_0} 個だけ生成すれば良い． $0 \leq i < L-1$ について、第 i 層の各マークル木は第 $(i+1)$ 層の 2^{h_i} 個のマークル木の根を署名するために使用される．第 $(L-1)$ 層（最下層）のマークル木は、それぞれ $2^{h_{L-1}}$ 個のメッセージの署名に使用される．

この署名方式では、一つのメッセージの署名の際に、各層についてそれぞれ一つのマークル木を生成しておけば十分である．各メッセージの署名は、そのメッセージに対する最下層のマークル木による署名と、 $0 \leq i < L-1$ について、そのメッセージの署名の際に使用された第 i 層のマークル木による第 $(i+1)$ 層のマークル木の根の署名からなる．この署名方式について、階層数 $L = 3$ 、各階層のマークル木の高さ $h_0 = h_1 = h_2 = 3$ の模式図を図 7.1 に示す．灰色の節点は認証パスをなす節点である．

7.2.4 プレフィクスとビットマスク

プレフィクスは、ハッシュ関数に基づく署名方式の処理において、すべてのハッシュ関数の計算がそれぞれ異なる入力に対して行われるよう入力に付加される系列である．プレフィクスは、Lighton と Micali [23] 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．

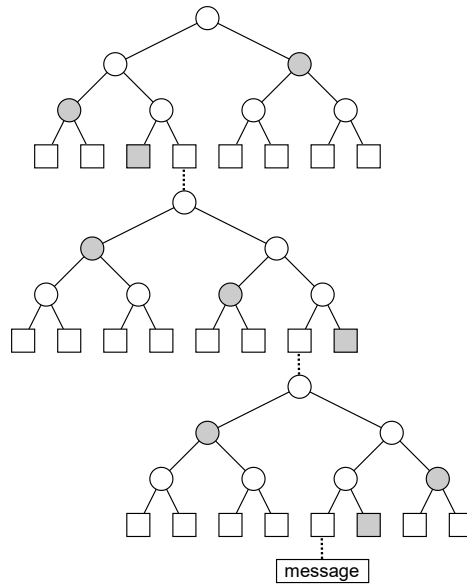


図 7.1: マークル木の階層構造による署名方式

7.3 ハッシュ関数に基づく主要な署名方式

本章で取り上げるハッシュ関数に基づく署名方式を表 7.1 に示す.

表 7.1: ハッシュ関数に基づく署名方式

文献	暗号化	鍵交換	署名
eXtended Merkle Signature Scheme (XMSS) [15, 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) [25]
- eXtended Merkle Signature Scheme (XMSS), multi-tree XMSS (XMSS^{MT}) [15]

LMS は Lighton と Micali による署名方式 [23] に基づく. HSS, XMSS^{MT} はそれぞれ, 7.2.3 節で述べられたような, LMS, XMSS の階層構造による署名方式である. ハッシュ関数に基づく stateful な署名方式では, 同一の秘密鍵が複数のメッセージの署名に使用されることがないように秘密鍵を管理することが必須である. LMS, HSS については, 調査報告書で取り上げられている.

SPHINCS⁺ [1] は 2022 年 7 月に NIST Post Quantum Cryptography Standardization Process で標準化の候補アルゴリズムの一つに選出された. SPHINCS⁺ はハッシュ関数に基づく stateless な署名方式であり, stateful な方式に求められるような秘密鍵の管理が不要な方式である. SPHINCS⁺ は SPHINCS [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. SPHINCS⁺ は XMSS の設計で得られた知見に基づいて設計されており, XMSS^{MT} とよく似た構造を有することから, 以下の SPHINCS⁺ の

記述では適宜 XMSS を参照する。

7.3.1 XMSS: eXtended Merkle Signature Scheme

XMSS は [3, 16] で提案された方式の改良版 [17] 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 [19] を用いる。

XMSS では三つの鍵付きハッシュ関数 F, H, H_{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。 H_{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 [30] または SHA-3 [31] 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。

$$\begin{aligned} F(k, x) &:= \text{SHA-256}([0]_{256} \| k \| x) \\ H(k, x) &:= \text{SHA-256}([1]_{256} \| k \| x) \\ H_{\text{msg}}(k, x) &:= \text{SHA-256}([2]_{256} \| k \| x) \\ R(k, x) &:= \text{SHA-256}([3]_{256} \| k \| x) \end{aligned}$$

XMSS では, ハッシュ関数の呼び出しをランダム化するために, それぞれのハッシュ関数の呼び出しで, 鍵とビットマスクが用いられる。これらは擬似ランダム関数を用いて生成され, 入力として Byte 系列の seed と長さ 32 Bytes のアドレス ADRS が与えられる。アドレスは 3 種あり, それぞれ OTS ハッシュアドレス, L 木アドレス, ハッシュ木アドレスと呼ばれる。それらの構造を図 7.2 に示す。

layer address	(32 bits)
tree address	(64 bits)
type = 0	(32 bits)
OTS address	(32 bits)
chain address	(32 bits)
hash address	(32 bits)
keyAndMask	(32 bits)

(a) OTS ハッシュアドレス

layer address	(32 bits)
tree address	(64 bits)
type = 1	(32 bits)
L-tree address	(32 bits)
tree height	(32 bits)
tree index	(32 bits)
keyAndMask	(32 bits)

(b) L 木アドレス

layer address	(32 bits)
tree address	(64 bits)
type = 2	(32 bits)
Padding = 0	(32 bits)
tree height	(32 bits)
tree index	(32 bits)
keyAndMask	(32 bits)

(c) ハッシュ木アドレス

図 7.2: アドレスの構造

7.3.1.1 WOTS⁺

$w \in \{4, 16\}$ は Winternitz パラメータと呼ばれる。 $\ell := \ell_1 + \ell_2$ は公開鍵, 秘密鍵, 署名を構成する Byte 長 n の要素の個数を表す。ここで,

$$\ell_1 := \lceil 8n / \log_2 w \rceil, \quad \ell_2 := \lfloor \log_2(\ell_1(w-1)) / \log_2 w \rfloor + 1$$

である。

■**チェイニング関数** チェイニング関数 chain の入力は、長さ 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．

$$\text{chain}(X, i, s, \text{seed}, \text{ADRS}) := \begin{cases} X & s = 0 \text{ のとき} \\ \text{NULL} & i + s \geq w \text{ のとき} \\ F(\text{Key}, \text{chain}(X, i, s - 1, \text{seed}, \text{ADRS}) \oplus \text{BM}) & \text{それ以外のとき} \end{cases}$$

ここで，

$$\text{Key} := R(\text{seed}, \text{ADRS}' \parallel [i + s - 1]_{32} \parallel [0]_{32}), \quad \text{BM} := R(\text{seed}, \text{ADRS}' \parallel [i + s - 1]_{32} \parallel [1]_{32})$$

である．なお， ADRS' は ADRS の上位 24 Bytes であり，例えば， $\text{ADRS}' \parallel [i + s - 1]_{32} \parallel [0]_{32}$ は図 7.2a の ADRS の hash address, keyAndMask の値をそれぞれ， $[i + s - 1]_{32}, [0]_{32}$ とすることを表している．

■**鍵生成アルゴリズム** 入力は ADRS, seed である．

1. $0 \leq i \leq \ell - 1$ について， $sk_i \in \{0, 1\}^{8n}$ を無作為に選択する．
2. $0 \leq i \leq \ell - 1$ について， ADRS の chain address の値を $[i]_{32}$ とし，

$$pk_i := \text{chain}(sk_i, 0, w - 1, \text{seed}, \text{ADRS})$$

とする．この計算を図 7.3 に示す．この図で

$$\text{Key}_j := R(\text{seed}, \text{ADRS}' \parallel [j]_{32} \parallel [0]_{32}), \quad \text{BM}_j := R(\text{seed}, \text{ADRS}' \parallel [j]_{32} \parallel [1]_{32})$$

である．

公開鍵は $pk := (pk_0, pk_1, \dots, pk_{\ell-1})$ である．秘密鍵は $sk := (sk_0, sk_1, \dots, sk_{\ell-1})$ である．

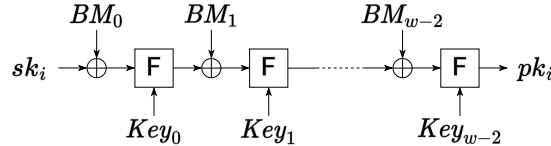


図 7.3: pk_i の計算

■**署名アルゴリズム** 入力は Byte 長 n のメッセージ M ，秘密鍵 sk ，アドレス ADRS ，シード seed である．

1. M をそれぞれ長さ $\log_2 w$ bits の ℓ_1 個のブロックに分割し，先頭から順に $M_0, M_1, \dots, M_{\ell_1-1}$ 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， $M_i \in \{0, 1, \dots, w - 1\}$ 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 $M_{\ell_1}, M_{\ell_1+1}, \dots, M_{\ell-1}$ とする．
4. $0 \leq i \leq \ell - 1$ について， ADRS の chain address の値を i とし，

$$sig_i := \text{chain}(sk_i, 0, M_i, \text{seed}, \text{ADRS})$$

とする．

メッセージ M に対する署名は $sig_0, sig_1, \dots, sig_{\ell-1}$ 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 [15] を参照のこと。

7.3.1.2 XMSS

XMSS はマー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。

$$RH(LEFT, RIGHT, seed, ADRS) := H(Key, (LEFT \oplus BM_0) \parallel (RIGHT \oplus BM_1))$$

ここで、

$$Key := R(seed, ADRS' \parallel [0]_{32}), \quad BM_0 := R(seed, ADRS' \parallel [1]_{32}), \quad BM_1 := R(seed, ADRS' \parallel [2]_{32})$$

である。なお、 $ADRS'$ は $ADRS$ の上位 28 Bytes であり、例えば、 $ADRS' \parallel [0]_{32}$ は $ADRS$ の図 7.2 の $keyAndMask$ の値を $[0]_{32}$ とすることを表している。

秘密鍵の生成には [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 $Node_{i,j}$ 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 $pk := (pk_0, pk_1, \dots, pk_{\ell-1})$, 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、 $Node_{i,0} := pk_i$ 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 $Node_{i,j+1}$ を計算する。なお、値の定義された $Node_{i,j}$ の個数を ℓ' とする。
 - (a) $0 \leq i < \lfloor \ell'/2 \rfloor$ について、 $Node_{i,j+1} := RH(Node_{2i,j}, Node_{2i+1,j}, seed, ADRS)$ 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、 $Node_{\lfloor \ell'/2 \rfloor, j+1} := Node_{\ell'-1, j}$ 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ル木の高さを h とすると、このマール木には 2^h 個の葉が存在する。このマール木に対応する 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 木の節点と同じ方で計算される。なお、このマー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} , マール木の根 $root$, $seed$ である。公開鍵は、マール木の根, $seed$ 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 $M' := H_{\text{msg}}(r \| \text{root} \| [idx]_{8n}, M)$ とする. ここで, $r := R(SK_{\text{PRF}}, [idx]_{32})$ 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 [15] を参照のこと.

7.3.1.3 XMSS^{MT}

XMSS^{MT} は, 7.2.3 節のマークル木の階層構造による署名方式に相当する. XMSS^{MT} 木はハイパー木と呼ばれ, d 層の XMSS 木からなる. ここで, XMSS 木は 7.3.1.2 節の鍵生成アルゴリズムで生成される L 木とマークル木からなる木を表す. 第 $(d-1)$ 層と第 0 層はそれぞれ, XMSS^{MT} 木の根と葉に相当する*1. すべての XMSS 木の高さは等しく, Winternitz パラメータもすべて同じ値が用いられる. 第 x 層の左から y 番目の XMSS 木の構成で使用される OTS ハッシュアドレス, L 木アドレス, ハッシュ木アドレスの layer address と tree address は, それぞれ $[x]_{32}$, $[y]_{32}$ である.

XMSS^{MT} の鍵生成, 署名, 検証の各アルゴリズムについての詳細は [15] を参照のこと.

7.3.1.4 パラメータの設定と安全性

Kampanakis と Fluhrer [20] により, LMS と XMSS の比較が論じられている.

Hülsing [17] 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 F, H, H_{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
- H_{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, F, H, 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 [15] 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 [15] 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 [15] では, 各層の番号付けが 7.2.3 節の番号付けとは逆順であり, 本稿でもそれに従って記述する.

RFC 8391 [15] の両方に掲載されている SHA-256 を用いる場合の WOTS⁺, XMSS, XMSS^{MT} のパラメータセットの値の一覧をそれぞれ表 7.2, 7.3, 7.4 に示す.

表 7.2: WOTS⁺ のパラメータセット

名称	n	w	ℓ
WOTSP-SHA2_256	32	16	67

表 7.3: XMSS のパラメータセットと署名長 (単位は Byte)

名称	n	w	ℓ	h	署名長
XMSS-SHA2_10_256	32	16	67	10	2,500
XMSS-SHA2_16_256	32	16	67	16	2,692
XMSS-SHA2_20_256	32	16	67	20	2,820

表 7.4: XMSS^{MT} のパラメータセットと署名長 (単位は Byte)

名称	n	w	ℓ	h	d	署名長
XMSSMT-SHA2_20/2_256	32	16	67	20	2	4,963
XMSSMT-SHA2_20/4_256	32	16	67	20	4	9,251
XMSSMT-SHA2_40/2_256	32	16	67	40	2	5,605
XMSSMT-SHA2_40/4_256	32	16	67	40	4	9,893
XMSSMT-SHA2_40/8_256	32	16	67	40	8	18,469
XMSSMT-SHA2_60/3_256	32	16	67	60	3	8,392
XMSSMT-SHA2_60/6_256	32	16	67	60	6	14,824
XMSSMT-SHA2_60/12_256	32	16	67	60	12	27,688

7.3.2 SPHINCS⁺

SPHINCS⁺[1] は 7.2.3 節のマークル木の階層構造による署名方式に基づく方式である. ただし, XMSS^{MT} とは異なり, stateless な署名方式である.

SPHINCS⁺ では, 以下のような, いくつかの tweakable ハッシュ関数 $\mathbf{T}_\ell$, 二つの擬似ランダム関数 $\mathbf{PRF}, \mathbf{PRF}_{\text{msg}}$, 一つの鍵付きハッシュ関数 $\mathbf{H}_{\text{msg}}$ が用いられる.

$$\begin{aligned} \mathbf{T}_\ell : \mathbb{B}^n \times \mathbb{B}^{32} \times \mathbb{B}^{\ell n} &\rightarrow \mathbb{B}^n & \mathbf{PRF} : \mathbb{B}^n \times \mathbb{B}^{32} &\rightarrow \mathbb{B}^n & \mathbf{H}_{\text{msg}} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^* &\rightarrow \mathbb{B}^m \\ \mathbf{PRF}_{\text{msg}} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B}^* &\rightarrow \mathbb{B}^n \end{aligned}$$

以下では, $\mathbf{T}_1, \mathbf{T}_2$ について, $\mathbf{F} := \mathbf{T}_1, \mathbf{H} := \mathbf{T}_2$ の表記が用いられる.

SPHINCS⁺ では, 図 7.4 に示す 7 種のアドレスが用いられる. どのアドレスも長さは 32 Bytes である.

7.3.2.1 WOTS⁺

SPHINCS⁺ でも WOTS⁺ が用いられているが, XMSS の WOTS⁺ と以下の点で異なる.

- Winternitz パラメータは $w \in \{4, 16, 256\}$ である.

layer address	(32 bits)
tree address	(96 bits)
type = 0	(32 bits)
key pair address	(32 bits)
chain address	(32 bits)
hash address	(32 bits)

(a) WOTS⁺ ハッシュアドレス

layer address	(32 bits)
tree address	(96 bits)
type = 1	(32 bits)
key pair address	(32 bits)
00...0	(32 bits)
00...0	(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...0	(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)
tree 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...0	(32 bits)
00...0	(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...0	(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...0	(32 bits)
tree index	(32 bits)

(g) FORS 鍵生成アドレス

図 7.4: アドレスの構造

- チェイニング関数は以下のように定義される.

$$\text{chain}(X, i, s, \mathbf{PK.seed}, \mathbf{ADRS}) :=$$

$$\begin{cases} X & s = 0 \text{ のとき} \\ \text{NULL} & i + s \geq w \text{ のとき} \\ \mathbf{F}(\mathbf{PK.seed}, \mathbf{ADRS}' \parallel [i + s - 1]_{32}, \text{chain}(X, i, s - 1, \mathbf{PK.seed}, \mathbf{ADRS})) & \text{それ以外のとき} \end{cases}$$

なお, $\mathbf{ADRS}$ は WOTS⁺ ハッシュアドレスであり, $\mathbf{ADRS}' \parallel [i + s - 1]_{32}$ は $\mathbf{ADRS}$ 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 $\mathbf{SK.seed}$, $\mathbf{PK.seed}$, WOTS⁺ ハッシュアドレス $\mathbf{ADRS}$ である.

1. $0 \leq i \leq \ell - 1$ について, $\mathbf{ADRS}$ の chain address の値を $[i]_{32}$, hash address の値を $[0]_{32}$ とし,

$$sk_i := \text{PRF}(\mathbf{SK.seed}, \mathbf{skADRS}) \quad pk_i := \text{chain}(sk_i, 0, w - 1, \mathbf{PK.seed}, \mathbf{ADRS})$$

とする. なお, $\mathbf{skADRS}$ 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. $pk := \mathbf{T}_\ell(\mathbf{PK.seed}, \mathbf{wotspkADRS}, pk_0 \parallel \dots \parallel pk_{\ell-1})$ 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 $sk := (sk_0, sk_1, \dots, sk_{\ell-1})$ である。

■署名アルゴリズム 詳細は [1] を参照のこと。

■検証アルゴリズム 詳細は [1] を参照のこと。

7.3.2.2 HT

SPHINCS⁺ では, ハイパー木 HT と呼ばれる XMSS^{MT} と同様の XMSS 木の階層構造が用いられる。HT が d 層の XMSS 木からなるとき, 第 $(d-1)$ 層と第 0 層はそれぞれ, HT の根と葉に相当する。すべての XMSS 木の高さは等しく, Winternitz パラメータもすべて同じ値が用いられる。HT の各 XMSS 木の各葉は, 7.3.2.1 節の WOTS⁺ の公開鍵である。各 XMSS 木の葉以外の節点の計算にはハッシュ関数 **H** が用いられる。第 x 層の左から y 番目の XMSS 木の構成で使用される WOTS⁺ ハッシュアドレス, WOTS⁺ 公開鍵圧縮アドレス, WOTS⁺ 鍵生成アドレス, ハッシュ木アドレスの layer address と tree address はそれぞれ $[x]_{32}$, $[y]_{96}$ である。

HT の鍵生成, 署名, 検証の各アルゴリズムについての詳細は [1] を参照のこと。

7.3.2.3 FORS (Forest of Random Subsets)

SPHINCS⁺ では, メッセージの署名に WOTS⁺ ではなく, FORS と呼ばれる方式が用いられる。FORS では一組の公開鍵と秘密鍵を用いて複数個のメッセージに署名できる。FORS は, 数回 (few-time) 署名方式 HORS [34] に基づく HORST [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 **SK.seed**, **PK.seed**, FORS 木アドレス **ADRS** である。

1. $0 \leq i < kt$ について,

$$sk_i := \mathbf{PRF}(\mathbf{SK.seed}, \mathbf{skADRS}) \quad \text{Node}_{i,0} := \mathbf{F}(\mathbf{PK.seed}, \mathbf{ADRS}, sk_i)$$

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. $1 \leq j \leq a$ について, それぞれ, $0 \leq i < kt/2^j$ について,

$$\text{Node}_{i,j} := \mathbf{H}(\mathbf{PK.seed}, \mathbf{ADRS}, \text{Node}_{2i,j-1} \parallel \text{Node}_{2i+1,j-1})$$

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. $pk := \mathbf{T}_k(\mathbf{PK.seed}, \mathbf{forspkADRS}, \text{Node}_{0,a} \parallel \dots \parallel \text{Node}_{k-1,a})$ 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, $\text{Node}_{0,a}, \text{Node}_{1,a}, \dots, \text{Node}_{k-1,a}$ 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 $sk_0, sk_1, \dots, sk_{kt-1}$ 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 $M_0, M_1, \dots, M_{k-1}$ に分割する。すなわち、 $M = M_0 \| M_1 \| \dots \| M_{k-1}$ 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 $sk_{0 \cdot t + M_0}, sk_{1 \cdot t + M_1}, \dots, sk_{(k-1)t + M_{k-1}}$ と、 $0 \leq i < k$ について、 $\text{Node}_{i,a}$ を根とするマークル木の $\text{Node}_{it+M_i,0}$ の認証パスである。

■検証アルゴリズム 詳細は [1] を参照のこと。

7.3.2.4 SPHINCS⁺

前節までの構成要素を用いて SPHINCS⁺ の署名が構成される。SPHINCS⁺ のパラメータは以下のとおりである。

- セキュリティパラメータ n (単位は Byte)
- Winternitz パラメータ w
- ハイパー木の高さ h と階層数 d
- FORS の木の個数 k と各木の葉の個数 t

メッセージダイジェストの Byte 長は $m := \lfloor (k \log_2 t + 7)/8 \rfloor + \lfloor (h - h/d + 7)/8 \rfloor + \lfloor (h/d + 7)/8 \rfloor$ となる。

■鍵生成アルゴリズム $\text{SK.seed}, \text{SK.prf} \in \mathbb{B}^n$ はいずれも無作為に選択される。 $\text{PK.seed} \in \mathbb{B}^n$ は無作為に選択される。 $\text{PK.root} \in \mathbb{B}^n$ は HT の第 $(d-1)$ 層の XMSS 木の根である。秘密鍵は $\text{SK.seed}, \text{SK.prf}, \text{PK.seed}, \text{PK.root}$ である。公開鍵は $\text{PK.seed}, \text{PK.root}$ 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. $\mathbf{R} := \text{PRF}_{\text{msg}}(\text{SK.prf}, \text{opt}, M)$ とする。 $\text{opt} = \text{PK.seed}$ 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. $\text{digest} := \mathbf{H}_{\text{msg}}(\mathbf{R}, \text{PK.seed}, \text{PK.root}, M)$ 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 $\text{tmp}_0, \text{tmp}_1, \text{tmp}_2$ とする。さらに、 tmp_0 の先頭 ka bits を md , tmp_1 の先頭 $(h - h/d)$ bits を idx_{tree} , tmp_2 の先頭 h/d bits を idx_{leaf} とする。
3. HT の第 0 層の左から idx_{tree} 番目の XMSS 木の左から idx_{leaf} 番目の葉に対応する FORS の鍵を用いて md の署名を生成する。このとき、FORS の **ADRS** の layer address は $[0]_{32}$, tree address は idx_{tree} , key pair address は idx_{leaf} である。さらに、tree height, tree index はともに $[0]_{32}$ である。
4. 上の署名で用いられた FORS の公開鍵への HT による署名を生成する。

M の署名は $\mathbf{R}$, md への FORS による署名, md への署名の検証に用いられる FORS の公開鍵への HT による署名からなる。

■検証アルゴリズム 詳細については [1] を参照のこと。

SPHINCS⁺ の秘密鍵, 公開鍵, 署名のサイズはそれぞれ, $4n$ Bytes, $2n$ Bytes, $(h + k(\log_2 t + 1) + d\ell + 1)n$ Bytes である。ここで、 $\ell := \ell_1 + \ell_2$ であり、 $\ell_1 := \lceil 8n/\log_2 w \rceil$, $\ell_2 := \lfloor (\log_2(\ell_1(w-1)))/\log_2 w \rfloor + 1$ である。

7.3.2.5 パラメータの設定と安全性

Hülsing と Kudinov [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 $\mathbf{H}_{\text{msg}}$, 擬似ランダム関数 $\text{PRF}, \text{PRF}_{\text{msg}}$ 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)
- $\mathbf{H}_{\text{msg}}$ が interleaved target subset resilience (ITSR) を満たすこと
- $\text{PRF}, \text{PRF}_{\text{msg}}$ が擬似ランダム関数 (PRF) であること

ここで, SM-TCR, SM-DSPR, ITSR は, $\mathbf{T}_\ell, \mathbf{H}_{\text{msg}}$ の構成に用いられるハッシュ関数の第二原像攻撃に対する安全性に基づく性質であり, SM-PRE は原像攻撃に対する安全性に基づく性質である. 一方, SM-UD, PRF は, 秘密鍵入力を有するハッシュ関数が擬似ランダム関数であることを要求する. さらに, ビットマスクの生成については, ハッシュ関数がランダムオラクルであることが仮定される.

SPHINCS⁺ については, 表 7.5 のパラメータセットが示されている. この表の最左欄のラベルの s と f はそれぞれ, 署名長, 計算時間について最適化されたパラメータセットであることを示している. ただし, 一方の最適化で他方が非実用的にならないよう配慮されている. また, 安全性レベルは NIST Post Quantum Cryptography Standardization Process の Call for Proposals に記された安全性強度のカテゴリである. なお, Haraka を用いる場合については, 表 7.5 とは異なり, $n = 24, 32$ のときの安全性はレベル 2 とされており, したがって, おおよそ 128-bit の安全性が確保される. Haraka の安全性について, Haraka-512 が (第二) 原像攻撃に対して 256-bit 安全性を有しないことが示されている [2] が, これは Haraka を用いた SPHINCS⁺ の安全性がレベル 2 であることを否定するものではない.

表 7.5: SPHINCS⁺ のパラメータセットの例. 署名長の単位は Byte である.

名称	n	h	d	$\log_2 t$	k	w	bit 安全性	安全性レベル	署名長
SPHINCS ⁺ -128s	16	63	7	12	14	16	133	レベル 1	7,856
SPHINCS ⁺ -128f	16	66	22	6	33	16	128	レベル 1	17,088
SPHINCS ⁺ -192s	24	63	7	14	17	16	193	レベル 3	16,224
SPHINCS ⁺ -192f	24	66	22	8	33	16	194	レベル 3	35,664
SPHINCS ⁺ -256s	32	64	8	14	22	16	255	レベル 5	29,792
SPHINCS ⁺ -256f	32	68	17	9	35	16	255	レベル 5	49,856

7.3.2.6 ハッシュ関数の実現法

SPHINCS⁺ のハッシュ関数はすべて, SHAKE256, SHA-2, Haraka のうちのいずれかを用いて定義される. なお, tweakable ハッシュ関数については robust と simple の二つの実現が示されている. robust な実現では 7.2.4 節で述べられたビットマスクが用いられるが, simple な実現では用いられない.

SHA-2 を用いた実現では, 当初は SHA-256 のみが用いられていたが, SHA-256 を用いた実現では安全性のレベル 5 が達成できないことを示す攻撃 [32] が示されたため, $n = 24, 32$ については, 一部の関数が SHA-512 を用いて実現されることとなった.

SHAKE256 を用いた構成は以下のとおりである.

$$\begin{aligned} \mathbf{H}_{\text{msg}}(\mathbf{R}, \mathbf{PK}.\text{seed}, \mathbf{PK}.\text{root}, M) &:= \text{SHAKE256}(\mathbf{R} \parallel \mathbf{PK}.\text{seed} \parallel \mathbf{PK}.\text{root} \parallel M, 8m) \\ \mathbf{PRF}(\mathbf{PK}.\text{seed}, \mathbf{SK}.\text{seed}, \mathbf{ADRS}) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \parallel \mathbf{ADRS} \parallel \mathbf{SK}.\text{seed}, 8n) \\ \mathbf{PRF}_{\text{msg}}(\mathbf{SK}.\text{prf}, \text{OptRand}, M) &:= \text{SHAKE256}(\mathbf{SK}.\text{prf} \parallel \text{OptRand} \parallel M, 8n) \end{aligned}$$

robust な実現では

$$\begin{aligned} \mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_1) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \parallel \mathbf{ADRS} \parallel M_1^\oplus, 8n) \\ \mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_1 \parallel M_2) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \parallel \mathbf{ADRS} \parallel M_1^\oplus \parallel M_2^\oplus, 8n) \\ \mathbf{T}_\ell(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \parallel \mathbf{ADRS} \parallel M^\oplus, 8n) \end{aligned}$$

simple な実現では

$$\begin{aligned} \mathbf{F}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_1) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \parallel \mathbf{ADRS} \parallel M_1, 8n) \\ \mathbf{H}(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M_1 \parallel M_2) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \parallel \mathbf{ADRS} \parallel M_1 \parallel M_2, 8n) \\ \mathbf{T}_\ell(\mathbf{PK}.\text{seed}, \mathbf{ADRS}, M) &:= \text{SHAKE256}(\mathbf{PK}.\text{seed} \parallel \mathbf{ADRS} \parallel M, 8n) \end{aligned}$$

である. ここで, $M \in \{0,1\}^l$ のとき, $M^\oplus := M \oplus \text{SHAKE256}(\mathbf{PK}.\text{seed} \parallel \mathbf{ADRS}, l)$ である.

7.4 ハッシュ関数に基づく署名技術に関するまとめ

本章では, ハッシュ関数に基づく署名技術として, XMSS, SPHINCS⁺ を取り上げた. これらはいずれも 7.2 節で述べた代表的なハッシュ関数に基づく署名方式に基づく構造を有する. XMSS [15] 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. XMSS は stateful な署名方式であるが, それを推奨アルゴリズムとする NIST SP 800-208 [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, XMSS^{MT} と同様の構造を有するが, 各メッセージの署名に一つの秘密鍵で数回署名可能な FORS を用いることによって署名可能な回数を増加させることにより, stateless であることを達成している.

第 7 章の参考文献

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

- [15] 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 閱覽)
- [16] 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.
- [17] 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.
- [18] A. Hülsing, M. Kudinov. Recovering the tight security proof of SPHINCS⁺. *IACR Cryptology ePrint Archive*, 2022/346.
- [19] A. Hülsing. W-OTS⁺ – shorter signatures for hash-based signature schemes. *AFRICACRYPT 2013*, volume 7918 of LNCS, pp. 173–188, Springer, 2013.
- [20] P. Kampanakis, S. Fluhrer. LMS vs XMSS: Comparison of two hash-based signature standards. *IACR Cryptology ePrint Archive*, 2017/349.
- [21] 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.
- [22] L. Lamport. Constructing digital signatures from a one-way function. Technical Report, CSL-98, SRI International, 1979.
- [23] F. T. Leighton, S. Micali. Large provably fast and secure digital signature schemes based on secure hash functions. US Patent 5,432,852, 1995.
- [24] 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.
- [25] 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 閱覽)
- [26] R. C. Merkle. *Secrecy, Authentication, and Public Key Systems*. PhD thesis, Stanford University, 1979. <https://www.merkle.com/papers/Thesis1979.pdf>.
- [27] R. C. Merkle. A certified digital signature. *CRYPTO 1989*, volume 435 of LNCS, pp. 218–238, Springer, 1989.
- [28] R. C. Merkle. One way hash functions and DES. *CRYPTO 1989*, volume 435 of LNCS, pp. 428–446, Springer, 1989.
- [29] NIST. Advanced encryption standard (AES). *FIPS PUB 197*, <https://csrc.nist.gov/publications/detail/fips/197/final>, 2001-11. (2023-04-10 閱覽)
- [30] NIST. Secure hash standard (SHS). *FIPS PUB 180-4*, <https://csrc.nist.gov/publications/detail/fips/180/4/final>, 2015-08. (2023-04-10 閱覽)
- [31] 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 閱覽)
- [32] 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.
- [33] J. Quisquater, M. Girault. $2n$ -bit hash-functions using n -bit symmetric block cipher algorithms. *EURO-CRYPT 1989*, volume 434 of LNCS, pp. 102–109, Springer, 1989.
- [34] 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.

CRYPTREC 暗号技術ガイドライン（耐量子計算機暗号）

[CRYPTREC GL-2004-2022]

不許複製 禁無断転載

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 2 8 番 8 号

独立行政法人情報処理推進機構

（セキュリティセンター セキュリティ技術評価部 暗号グループ）

INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN

2-28-8 HONKOMAGOME, BUNKYO-KU

TOKYO, 113-6591 JAPAN

