

2024年10月18日 全21頁

# 耐量子計算機暗号とは何か

## 移行のために知っておきたいこと

デジタルソリューション研究開発部 松井 直己

### [要約]

- 現代の安全な通信は公開鍵暗号技術によって支えられているが、量子コンピュータによってその安全性が脅かされる可能性が指摘されている。そのため、量子コンピュータに対しても安全であると期待される耐量子計算機暗号（post-quantum cryptography, PQC）が求められている。
- 現時点では、現在利用されている公開鍵暗号を破れる程度の計算能力を持つ量子コンピュータは存在していない。しかし、ハーベスト攻撃（暗号化通信や署名等を今から収集し、実用化されたタイミングで解読を試みる）の存在が指摘されている。
- NISTは2016年から、既存の電子署名アルゴリズムと鍵交換アルゴリズムを置き換えるべく耐量子計算機暗号の候補を公募し、長い時間を掛けて安全性や性能を精査した。2024年8月13日には耐量子計算機暗号として初めての標準の最終版が承認された。
- 耐量子計算機暗号はその安全性を十分に精査されているが、攻撃アルゴリズムの発見により、最新鋭とは言えない現在のコンピュータによって破られる可能性がある。そのため、安全性について過信せず、必要に応じて既存の暗号と併用するハイブリッド方式を選択することでリスクを抑制できる。
- 通信や記憶領域等の観点で、耐量子計算機暗号はこれまでの暗号よりコストがかかる可能性もある。耐量子計算機暗号へ移行する場合、アルゴリズムの選定は用途に応じて慎重に行う必要がある。
- 暗号は多くのシステムの様々な場面で利用されていることから、移行のためのコストが膨大となりうる。今回だけではなく将来も含めた暗号を移行するコストを抑えるためにクリプトアジリティを確保することが重要である。
- 移行のために今からできることとして、使用している暗号のインベントリやCBOM（cryptography bill of materials）の作成、ハードウェアの調達条件に耐量子計算機暗号への対応の有無・可否やクリプトアジリティの確保が含まれているかの確認が挙げられる。

## 目次

1. はじめに.....	3
(1) 耐量子計算機暗号とは.....	3
(2) ハーベスト攻撃.....	3
2. 耐量子計算機暗号標準化の経緯.....	4
(1) NISTIR 8105.....	4
(2) NIST による耐量子計算機暗号の公募から標準化まで .....	5
(3) NIST の選定の基準 .....	5
3. 耐量子計算機暗号へ移行するにあたっての注意点.....	6
(1) 耐量子計算機暗号は現代暗号より安全であるとは限らない.....	6
(2) 耐量子計算機暗号のデメリット.....	7
(3) ハイブリッド方式の利用.....	9
(4) クリプトアジリティの確保.....	9
(5) 今からすべきこと.....	10
Appendix A. 現代暗号と量子コンピュータ.....	11
(1) 現代暗号技術総論.....	11
(2) ビットセキュリティ.....	12
(3) RSA 暗号・楕円曲線暗号・DH 鍵共有と量子コンピュータ .....	12
(4) 共通鍵暗号・ハッシュ関数と量子コンピュータ .....	13
Appendix B. 選定された耐量子計算機暗号の概要.....	14
(1) CRYSTALS-KYBER .....	14
(2) CRYSTALS-Dilithium.....	14
(3) FALCON.....	15
(4) SPHINCS <sup>+</sup> .....	15
(5) BIKE.....	16
(6) Classic McEliece.....	16
(7) HQC.....	17
(8) SIKE.....	17
引用文献.....	19

## 1. はじめに

### (1) 耐量子計算機暗号とは

量子コンピュータとは、量子力学的な性質を利用することで現在主流の計算機（以下、古典コンピュータ）では効率よく解けなかった問題を効率よく解けると期待されている次世代の計算機である。その性能や実現の可能性・時期については様々な見立てがあるが、ある程度の計算能力を持つ量子コンピュータがいずれ実現すると仮定するならば、現在主流の公開鍵暗号が Shor のアルゴリズムによって脆弱になることが指摘されている<sup>1</sup>。

公開鍵暗号は現代の安全な通信のために必要不可欠な技術である。たとえば、以下のような場面で使用されている：

- アクセス先のウェブサイトや端末が正当であることを保証するための電子証明書
- ファイルが改竄されていないことを証明するための電子署名
- 通信相手になりすましでないことを確認するための認証
- 通信の内容を暗号化するための鍵を共有する鍵交換

つまり、公開鍵暗号の安全性を脅かす程度の計算能力を持つ量子コンピュータが実現したとき、通信や保管データの秘匿が守られない、署名が偽造される、といったことが発生するだろう。そこで、実用的な量子コンピュータに対しても安全性を保つことができると期待される暗号が求められている。そのような暗号を耐量子計算機暗号（post-quantum cryptography, PQC）と呼ぶ。

量子が関係する情報技術として、前述の量子コンピュータ、耐量子計算機暗号の他に量子通信が存在する。量子通信には更に大容量化を志向する量子符号化と安全性を志向する量子暗号の二つのテーマがある。量子暗号とは量子を利用した暗号技術の総称であり、その代表的なものとしては量子鍵配送がある。但し、現在研究中の技術であり、量子鍵配送を利用するためには専用のハードウェアが必要となる。

耐量子計算機暗号は量子暗号とは異なり、量子を利用した暗号ではない。そのため、専用のハードウェアを必要とせず、古典コンピュータで利用可能な暗号方式である。

### (2) ハーベスト攻撃

そもそも公開鍵暗号を始めとした現代暗号<sup>2</sup>を破れる程度に実用的な量子コンピュータは現時点では存在していない。だからといって、今は何もしなくて良いと考えるのは早計である。将来的に実用化されると期待して今から通常の通信によって署名等を、盗聴によって暗号化され

<sup>1</sup> 実際にどの程度脆弱になるかについては本レポート Appendix A. (3)を参照のこと。

<sup>2</sup> 現在使われている暗号技術の総称。詳細は Appendix A. (1)を参照のこと。

た通信やデータ等を集め、実用化されたタイミングで解読を試みるハーベスト攻撃<sup>3</sup>の存在が指摘されている。この攻撃によって長期間安全に保管すべきデータが解読されたり、過去の署名付き文書の信頼性が失われたりするといったリスクが未来に生じることになる。そのため、ハーベスト攻撃のリスクを抑えるために、早い段階で耐量子計算機暗号へ移行しておくことが望ましい。

## 2. 耐量子計算機暗号標準化の経緯

### (1) NISTIR 8105

前述の通り、十分な計算能力を持つ量子コンピュータの実現によって公開鍵暗号の安全性が損なわれることが分かっている。NIST<sup>4</sup>はこの潜在的な脅威への対策として 2015 年 4 月 2 日から 3 日にかけてワークショップを開催し、2016 年 4 月には耐量子計算機暗号に関する調査報告書 NISTIR 8105 [1]を公開した。これらに対するフィードバックを受け、NIST は量子コンピュータに対して耐性のある公開鍵暗号アルゴリズムの追加開発を決定した [2]。

耐量子計算機暗号への移行が早い段階、特に今であるべき理由が NISTIR 8105 に記載されている。その NISTIR 8105 の Introduction の要約は以下の通りである。なお、和訳及び要約は筆者による。

公開鍵暗号は我々の安全な通信のためには不可欠なものとなっている。そして多くの重要な通信プロトコルは公開鍵暗号、デジタル署名、鍵交換の三つに依存している。これらは通常、DH 鍵共有、RSA 暗号、楕円曲線暗号を用いて実装され、それらの安全性は離散対数問題の困難性や素因数分解問題の困難性と関連している。

1994 年、ベル研究所の Peter Shor は、量子コンピュータによって離散対数や素因数分解を効率良く計算できることを示した。つまり、十分な計算能力のある量子コンピュータは現在使われている鍵交換や暗号化、デジタル認証といった多くの通信技術を危険に曝す。

「量子コンピュータがいつ実現されるのか」は複雑な問題である。ただ、かつては物理的に可能かということ自体が明確ではなかったが、現在では単なる重要な工学的挑戦として考えられている。また、ある専門家は 2031 年までに RSA-2048 を破る確率は 1/2 と述べている [3]。現在の公開鍵暗号インフラを展開するにはほぼ 20 年掛かったことから、現在広く使われている暗号システムから量子コンピュータに耐性のある暗号へのスムーズで安全な移行を確保するためには大きな努力が必要となる。従って、量子コンピュータの時代が到来する正確な時期を予測できるかどうかに関係なく、今から情報セキュリティシステムを準備し、量子コンピュータに対抗できるようにする必要がある。

<sup>3</sup> Harvest now, decrypt later 攻撃や store now, decrypt later 攻撃等とも呼ばれる。

<sup>4</sup> アメリカ国立標準技術研究所 (National Institute of Standards and Technology)。

## (2) NIST による耐量子計算機暗号の公募から標準化まで

NIST は 2016 年 12 月 20 日から 2017 年 11 月 30 日にかけて耐量子計算機暗号アルゴリズムの候補を公募した [4]。この公募ではデジタル署名アルゴリズム、離散対数暗号を利用した鍵確立アルゴリズム、素因数分解を利用した鍵確立アルゴリズムを置き換えることを目指している。提出されたアルゴリズムは 82 個で、その内の 69 個が第 1 ラウンドの候補として受け入れられた。

その後、2019 年 1 月 30 日には第 2 ラウンドに進出した 17 個の公開鍵暗号方式・鍵確立アルゴリズムと 9 個のデジタル署名アルゴリズムが発表された [5] [6]。更に 1 年半程経過し、2020 年 7 月 22 日には 4 個の公開鍵暗号方式・鍵確立アルゴリズムと 5 個の代替アルゴリズム、3 個のデジタル署名アルゴリズムと 3 個の代替アルゴリズムが発表された [7]。

2022 年 7 月 5 日、第 3 ラウンドに進出したアルゴリズムから、公開鍵暗号方式・鍵確立アルゴリズムとして CRYSTALS-KYBER が、デジタル署名アルゴリズムとして CRYSTALS-Dilithium、FALCON、SPHINCS+が標準化対象として選ばれた<sup>5</sup> [8]。同時に、BIKE、Classic McEliece、HQC、SIKE が第 4 ラウンドに進出した。なお、2023 年には SIKE が破られた<sup>6</sup>。

2023 年 8 月 24 日、標準化対象として選ばれた CRYSTALS-KYBER、CRYSTALS-Dilithium、SPHINCS+を基にしたアルゴリズム標準の草案が公開された [9]。同年 11 月 22 日までフィードバックを募集し、2024 年 8 月 13 日には米国商務長官によって標準の最終版が承認された [10]。

## (3) NIST の選定の基準

NIST が提出された耐量子計算機暗号候補から標準化対象を選定するにあたって、「セキュリティ [11]」「コストとパフォーマンス [12]」に加えて、「アルゴリズムと実装における特徴 [13]」を評価基準としている。

セキュリティ要件では用途ごとの定義に加え、量子セキュリティレベルを 1 から 5 で設定している（次ページ図表 1）。併せて、AES に対する鍵回復攻撃と SHA-3 に対する衝突攻撃に必要な古典的ゲート数と量子ゲート数の推定値も公開している。

コストとパフォーマンスでは、公開鍵等のサイズや計算効率、復号の失敗率を挙げている。アルゴリズムと実装における特徴では、スキームの柔軟性や設計のシンプルさ、ライセンス等の有無を挙げている。

また、NIST はアルゴリズムが特定の方式に偏重しないことも重視している。たとえば、SIKE は第 2 ラウンド候補唯一の同種写像を利用したアルゴリズムであり、NISTIR 8309 [14]では同種写像ベースの暗号の研究の更なる奨励を述べている。また、Picnic は非対話型ゼロ知識証明

<sup>5</sup> 標準化対象となった 4 個のアルゴリズムと第 4 ラウンドに進出した 4 個のアルゴリズムの概要と NIST による評価については Appendix B. を参照のこと。

<sup>6</sup> 本レポート 3. (1)を参照のこと。

を利用しており、署名スキームの設計手法の新しさや安全性の仮定が保守的であることから、パフォーマンスの悪さを指摘されながらも代替候補として選ばれている<sup>7</sup>。一方で、Round5は優れたパフォーマンスを持つと評されたが、構造化格子ベースのスキームであり、少数のアルゴリズムへの集中を避けるために選択されなかった。他にも、第3ラウンドを勝ち抜けた公開鍵暗号・鍵確立アルゴリズムは Classic McEliece、CRYSTALS-KYBER、NTRU、Saber で、このうち Classic McEliece 以外は構造化格子スキームであるため最大で一つを選択する意向であることが明言されていた。

図表1 量子セキュリティレベル

レベル	カテゴリの定義
1	破るためには 128 ビットの鍵を持つブロック暗号の鍵探索に必要な計算資源と同等以上の計算資源を必要とする。
2	破るためには 256 ビットのハッシュ関数の衝突探索に必要な計算資源と同等以上の計算資源を必要とする。
3	破るためには 192 ビットの鍵を持つブロック暗号の鍵探索に必要な計算資源と同等以上の計算資源を必要とする。
4	破るためには 384 ビットのハッシュ関数の衝突探索に必要な計算資源と同等以上の計算資源を必要とする。
5	破るためには 256 ビットの鍵を持つブロック暗号の鍵探索に必要な計算資源と同等以上の計算資源を必要とする。

(出所) [11]より大和総研作成

### 3. 耐量子計算機暗号へ移行するにあたっての注意点

#### (1) 耐量子計算機暗号は現代暗号より安全であるとは限らない

耐量子計算機暗号として標準化されたアルゴリズムはその安全性が十分検討され、古典コンピュータだけではなく量子コンピュータに対しても安全であると考えられている。しかし、耐量子計算機暗号は現在使われている公開鍵暗号より常に安全であるとは限らないことに留意しなければならない。

SIKE は第2ラウンド候補の中で唯一の同種写像を用いたアルゴリズムであり、パフォーマンスは他の候補と比べて1桁程度悪いながらも公開鍵サイズ・暗号文サイズが小さい点が非常に魅力的な候補であるとして第4ラウンド候補に選出された。しかし、2022年に Castryck and Decru [15]でその安全性が破られた。著者らは Magma<sup>8</sup> V2.27-5 で作成されたコードを 2.60GHz

<sup>7</sup> なお、セキュリティを定量化できなかったことや、multi-party-computation-in-the-head paradigm から派生する暗号システムの設計がより優れたものになる可能性があるとして第3ラウンドでは選択されなかった。

<sup>8</sup> 代数学分野の計算に特化したソフトウェアパッケージ。

の Intel Xeon CPU E5-2630v2<sup>9</sup>上で実行し、その攻撃時間は、量子セキュリティレベル 1 を満たすと考えられていた SIKEp434 で約 10 分と報告している。また、SIKEp503 (レベル 2) では約 20 分、SIKEp610 (レベル 3) では約 55 分、SIKEp751 (レベル 5) では約 3 時間 15 分であった。この攻撃の基礎となったのは 1997 年の Kani [16]での結果であった。この攻撃が致命的でありながら 25 年以上発見されなかった理由として、攻撃に非常に高度な数学を適用していることが指摘されている [17]。

また、署名アルゴリズムとして提出された Rainbow も Beullens [18]で 2.5GHz の Intel i9-10885H<sup>10</sup>を 8 コア使用して約 53 時間で破れると指摘されている。

このように、他の有力視されているアルゴリズムに関しても最新鋭とは言えないスペックの古典コンピュータで破られてしまう可能性は否定できない。つまり、コストを掛けて移行した後に採用したアルゴリズムが危殆化してしまった場合、再び同じだけのコストを掛けて再移行する必要があるかもしれない。再移行の可能性に備えて後述のハイブリッド方式の利用やクリプトアジリティの確保をしておくことで、別の暗号方式へ移行するための時間の確保やコストの抑制ができる。

## (2) 耐量子計算機暗号のデメリット

耐量子計算機暗号のデメリットとして、鍵交換にかかる時間や署名のサイズが増大する可能性が挙げられる。採用するアルゴリズムによって差はあるものの、通信や記憶のためのコストが数倍程度となり得る<sup>11</sup>。そのため、頻繁に鍵交換や署名の送信を行う場合や通信容量・記憶容量に制約があるような場合は採用するアルゴリズムの鍵サイズや署名サイズなどに気を付ける必要がある。

NISTIR 8413 [19]では、各アルゴリズムの鍵・暗号文・署名サイズを公開している<sup>12</sup>。鍵カプセル化メカニズムについては図表 2、署名アルゴリズムについては図表 3 の通りである。

**図表 2 鍵カプセル化メカニズムのファイナリストの鍵サイズと署名サイズ**

アルゴリズム名	主張する安全性	公開鍵サイズ	秘密鍵サイズ	暗号文サイズ
Classic McEliece348864	1	261, 120	6, 492	128
Classic McEliece460896	3	524, 160	13, 608	188
Classic McEliece6688128	5	1, 044, 992	13, 932	240

<sup>9</sup> 2013 年に発売されたサーバ向けの CPU。

<sup>10</sup> 2020 年に発売されたノートパソコン向けの CPU。

<sup>11</sup> 参考までに RSA 暗号・RSA 署名のサイズを紹介する。規格によって微妙に異なるが、基本的に暗号文・署名サイズは RSA-2048 なら 256 バイト、RSA-3072 なら 384 バイトである。公開鍵サイズはそれらに公開指数のサイズを足したものとなる。

<sup>12</sup> アルゴリズムによってはホームページでもサイズや鍵生成にかかる時間等のベンチマークが掲載されている。また、<https://bench.cr.yp.to/>でも有志による測定結果が公開されている。

Classic McEliece6960119	5	1, 047, 319	13, 948	226
Classic McEliece8192128	5	1, 357, 824	14, 120	240
KYBER512	1	800	1, 632	768
KYBER768	3	1, 184	2, 400	1, 088
KYBER1024	5	1, 568	3, 168	1, 569
BIKE	1	1, 540	280	1, 752
BIKE	3	3, 082	418	3, 114
BIKE	5	5, 122	580	5, 154
HQC-128	1	2, 249	40	4, 481
HQC-192	3	4, 522	40	9, 026
HQC-256	5	7, 245	40	14, 469

(出所) [19], Tables 6 and 7 より大和総研作成

(注) 安全性は量子セキュリティレベル、サイズの単位はバイト。

図表3 署名アルゴリズムのファイナリストの鍵サイズと署名サイズ

アルゴリズム名	主張する安全性	公開鍵サイズ	秘密鍵サイズ	署名サイズ
Dilithium	2	1, 312	2, 528	2, 420
Dilithium	3	1, 952	4, 000	3, 293
Dilithium	5	2, 592	4, 864	4, 595
FALCON-512	1	897	7, 553	666
FALCON-1024	5	1, 793	13, 953	1, 280
SPHINCS <sup>+</sup> -128s	1	32	64	7, 856
SPHINCS <sup>+</sup> -128f	1	32	64	17, 088
SPHINCS <sup>+</sup> -192s	3	48	96	16, 224
SPHINCS <sup>+</sup> -192f	3	48	96	35, 664
SPHINCS <sup>+</sup> -256s	5	64	128	29, 792
SPHINCS <sup>+</sup> -256f	5	64	128	49, 856

(出所) [19], Tables 8 and 9 より大和総研作成

(注) 安全性は量子セキュリティレベル、サイズの単位はバイト。

### (3) ハイブリッド方式の利用

ハイブリッド方式とは、鍵交換やデジタル署名において、現在使われているアルゴリズムとポスト量子アルゴリズムを組み合わせる方式のことである。現行の暗号と耐量子計算機暗号の両方のアルゴリズムを適切に組み合わせることにより、一方が危殆化しても、もう一方のアルゴリズムによって一定程度は暗号としての機能を果たすことができる。これは単に保険となるだけでなく、別の暗号方式へ移行するための時間稼ぎにもなる。

一方で、ハイブリッド方式は単一のアルゴリズムを使用するよりも実装や保守の点でより複雑になり、コストや効率の面では悪くなる。そのため、相互運用性やセキュリティ、プロトコル等によって課せられる制約に応じてハイブリッド方式の採用の有無を選択することも必要である。

なお、ハイブリッドは対称アルゴリズムと非対称アルゴリズムを組み合わせているという意味で既に使われており [20]、IETF<sup>13</sup>では伝統的アルゴリズムとポスト量子アルゴリズムを組み合わせた構造のための用語として PQ/T ハイブリッドが提案されている [21]。

### (4) クリプトレジリティの確保

暗号を移行するにあたって、日本銀行金融研究所がディスカッションペーパーを出している [22]。以下、その概要を説明する。

まず、システム移行は通常、「システム移行の準備」「新システムへの切替え」「旧システムの暗号化データの保護」という三つのフェーズで構成される。量子コンピュータによって既存の暗号を現実的な時間で解読可能となる時期をそれぞれのフェーズで迎える三つのケースが想定される。

まずは、システム移行の準備のフェーズで迎えた場合である。このとき、旧システムで生成されたまたは保管されている暗号化データが量子コンピュータを使用する攻撃者によって盗聴・解読・改変・偽造される可能性が生じる。

次に、新システムへの切替えのフェーズで迎えた場合である。このとき、旧システムはまだ稼働していることから、システム移行の準備のフェーズと同様の可能性が想定される。加えて、新システムには移管された旧システムの暗号化データがあることから、こちらも攻撃の対象となり得る。

最後に、旧システムの暗号化データの保護のフェーズで迎えた場合である。このとき、旧システムは既に稼働を停止していることから、こちらは問題にはならない。一方、新システムに移管された旧システムの暗号化データは依然として残っているため、引き続きこちらは攻撃対象となり得る。

<sup>13</sup> Internet Engineering Task Force

従って、上記のシステム移行の各フェーズの期間を短くし、量子コンピュータによって既存の暗号を解読されるようになる前に旧システムの暗号化データの保護期間を満了することが望ましい。そのためには、データ保護期間を最小限に設定した上で、情報システムの設計の時点で暗号移行に伴うシステム移行の準備期間や切替え期間を短縮することが求められる。このような設計思想をクリプトアジリティ (crypto agility) という。

SIKE や Rainbow のように耐量子計算機暗号の有力候補が突然破られる可能性も十分にあることから、クリプトアジリティは既存の暗号から耐量子計算機暗号への移行だけではなく、耐量子計算機暗号が危殆化した場合にも有用である。

## (5) 今からすべきこと

耐量子計算機暗号として初となる標準の最終版は 2024 年 8 月 13 日に決定されたが、それ以前から既に Google Chrome は対応しており、2024 年 8 月時点では Cloudflare ネットワークの 15%程度が X25519 と KYBER のハイブリッドで保護されている [23]。標準の最終版の決定により、各ベンダーはこれまでよりも本格的に耐量子計算機暗号へ対応し始めるだろう。ユーザー企業はそれに合わせてシステムを入れ替える等によって対応することになる。しかし、一度に全てのシステムで耐量子計算機暗号へ対応することは現実的ではないため、優先度をつけて対応していくことになる。その対応の可否や優先順位をつけるためにはシステムのどこに何の暗号を利用しているのかを把握している必要がある。従って、耐量子計算機暗号への移行のための第一歩は現在使用している暗号のインベントリの作成である。

また、近年では脆弱性管理やライセンスに関する予防法務の一環として SBOM (software bill of materials) の作成が経済産業省等によって推奨されている。ソフトウェアと同様に暗号に関しても、資産の暗号アルゴリズムやライブラリ、鍵管理情報、ビットセキュリティや量子セキュリティレベルなどを管理する CBOM (cryptography bill of materials) が存在する。CBOM の作成は耐量子計算機暗号への移行をやすくするだけではなく、脆弱性管理という観点でも重要である。CBOM に関するツールとしては、たとえば IBM 社が CycloneDX を拡張した CBOM のオブジェクトモデル [24] や CBOM を生成する SonarQube プラグイン [25] を公開している。

暗号は多くのシステムの様々な場面で利用していることから、移行対応のための作業量が膨大であることも考えられる。まずは一つのシステムに対してインベントリや CBOM を作成することで、システム全体で使用している暗号を耐量子計算機暗号へ移行するために必要な時間や人員の見通しも立てやすくなる。

加えて、ハードウェアは一度導入するとしばらく使うものである。そのため、現時点で耐量子計算機暗号への対応が可能であったり、クリプトアジリティが十分確保されていたりすることが望ましい。これからの更新や新規導入時にはそれらの条件が調達条件に含まれているかの確認も重要である。

## Appendix A. 現代暗号と量子コンピュータ

### (1) 現代暗号技術総論

現在使われている暗号技術は、それ以前に利用されていた暗号技術と比較して、「アルゴリズムが公開されている<sup>14</sup>」「電子計算機を利用する」「安全性は数学的理論に依拠している」といった特徴を有する。具体的な現代暗号のアルゴリズムとしては、AES、RSA 暗号、楕円曲線暗号、DH 鍵共有といったものがよく利用されている。

一般に暗号で想定されるものは、復号のための鍵を知らない場合には平文を知ることができないという意味で、機密性を担保する技術である。現代暗号ではこの機密性に加え、後述の鍵交換や完全性<sup>15</sup>、真正性<sup>16</sup>、否認防止といった機能も取り扱っている。

機密性を実現するという最も基本的な目的で用いられる暗号は、アルゴリズムと事前に生成した鍵、必要に応じて乱数を利用して、平文の暗号化・暗号文の復号を行う。この観点では、現代暗号は共通鍵暗号と（狭義の）公開鍵暗号に大別される。共通鍵暗号は暗号化・復号で同じ鍵を使用する暗号方式で、公開鍵暗号は異なる鍵を使用する暗号方式である。共通鍵暗号の代表例としては AES やワンタイムパッドが、公開鍵暗号の代表例としては RSA 暗号や楕円曲線暗号が挙げられる。

共通鍵暗号の問題は、暗号化・復号のための鍵を第三者に知られずに共有する必要があることである（鍵配送問題）。これを解決する方法が鍵交換であり、具体的な技術として DH 鍵共有が挙げられる。また、前述の公開鍵暗号は鍵交換に転用することもできる。

次に、現代暗号を語る上で欠かせない技術としてデジタル署名がある。これは秘密鍵<sup>17</sup>によって生成されたデータ（署名）を公開鍵によって検証することで真正性や完全性、否認防止の機能を実現する技術である。RSA 暗号や楕円曲線暗号等の一部の公開鍵暗号はこのデジタル署名に転用されているが、必ずしもデジタル署名に対応する公開鍵暗号が存在するわけではなく、また、必ずしも公開鍵暗号をデジタル署名に転用できるというわけでもない。

上記の（狭義の）公開鍵暗号・鍵交換・デジタル署名を総称して（広義の）公開鍵暗号と呼ぶことも多い。そのため、「公開鍵暗号」やそれに関する用語の使用において混乱が生じることもある。

他にも、完全性を担保する機能を持つハッシュ関数、真正性を担保する機能を持つ認証といった技術がある。

<sup>14</sup> オランダの言語学者・暗号研究者である Auguste Kerckhoffs が提案した軍用暗号についての原則の一つとして「暗号システムに秘密は必要なく、敵に盗まれても問題が起きるべきではない」というものがある。現在はこれを Kerckhoffs の原理と言い、現代暗号のアルゴリズムを公開する慣行はこの原理を体現したものである。

<sup>15</sup> データの改竄や破壊等がなく、正確な状態にあるということ。

<sup>16</sup> 人やデータ等が主張通りであるということ。

<sup>17</sup> デジタル署名の場合は、秘密鍵を署名鍵、公開鍵を検証鍵ということもある。

## (2) ビットセキュリティ

通常、暗号に使用する鍵の長さはセキュリティ強度を表す。そのため、同一のアルゴリズムについては鍵が長ければ長いほど安全である。しかし、異なるアルゴリズム間では、鍵の長さが同一であったとしてもそのセキュリティ強度が等しいわけではないため、単純に鍵の長さだけで安全性を比較することはできない。そこで、ビットセキュリティと呼ばれる指標を用いてセキュリティ強度を比較する。

ある暗号が $n$ ビットセキュリティを持つとは、その暗号への攻撃アルゴリズムの計算時間が $t$ で成功確率が $\varepsilon$ のとき、 $t/\varepsilon \geq 2^n$ であることをいう<sup>18</sup>。これは、 $n$ ビットセキュリティを持つ暗号を破るためには少なくとも $2^n$ 回の計算が平均的には必要であることを意味している。

NIST は様々なアルゴリズムについてそのセキュリティ強度を公表している [26]。日本では CRYPTREC が『暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準』で公開鍵暗号・共通鍵暗号・ハッシュ関数のセキュリティ強度やその考え方について公表している [27]。一例を挙げると、CRYPTREC は RSA-2048 のビットセキュリティは 112 で 2031 年以降は原則として使用不可とし、2040 年まではビットセキュリティが 128 以上のものを、2041 年以降は 192 以上のものを選択すべきとしている。

## (3) RSA 暗号・楕円曲線暗号・DH 鍵共有と量子コンピュータ

RSA 暗号が安全であることの必要条件として素因数分解問題の困難性がある。特に、素因数を効率よく<sup>19</sup>求めることができる場合、RSA 暗号の完全解読や RSA 署名の偽造が可能となる。同値性は証明されていないことから、素因数を効率よく求められなかったとしても RSA 暗号が破られる可能性は否定できない。しかし、脆弱な実装である場合を除くと、そのような攻撃方法は見つかっていない。そのため、セキュリティ強度を考えるにあたって、今のところは素因数分解をいかに効率良く行えるかが問題となる。

CRYPTREC は過去に素因数分解を行うアルゴリズムについての調査をしている [28]。RSA 暗号に対する攻撃という観点で最も効率がよいアルゴリズムは一般数体篩法<sup>ふるい</sup>で、 $n$ ビット整数に対するその計算量は $\exp O(n^{1/3}(\log n)^{2/3})$ と準指数オーダーである。一方、量子コンピュータでは Shor のアルゴリズムにより、多項式オーダーの $O(n^3)$ で素因数分解が可能である<sup>20</sup>。

同様に、DH 鍵共有や楕円曲線暗号が安全であることの必要条件として離散対数問題の困難性がある。現在、離散対数を求めるアルゴリズムで最も効率がよいのは数体篩法で、素因数分解問題と同じく計算量は $\exp O(n^{1/3}(\log n)^{2/3})$ である。Shor のアルゴリズムを利用した場合も同様に $O(n^3)$ で離散対数を求めることができる。

<sup>18</sup> 成功確率が $\varepsilon$ の試行について、初めて成功させるまでの試行回数の期待値は成功確率の逆数となる。

<sup>19</sup> 「効率よく」とは多項式時間で実行できることをいう。

<sup>20</sup> たとえば、Nielsen-Chuang, Quantum Computation and Quantum Information, Section 5.3 等を参照。なお、乗算を工夫することで更に高速化することが可能である。

#### (4) 共通鍵暗号・ハッシュ関数と量子コンピュータ

共通鍵暗号に対する最良の攻撃方法は基本的に全数探索である。そのため、共通鍵暗号のビットセキュリティは、全数探索より効率的な攻撃方法が見つかった場合を除き、鍵長と一致する。

ハッシュ関数の場合は衝突困難性<sup>21</sup>を必要とするかどうかでセキュリティ強度が異なる。衝突困難性を必要としない場合は原像計算困難性<sup>22</sup>があれば十分で、この場合の最良の攻撃方法は共通鍵暗号と同じく全数探索となるから、基本的にハッシュ長と一致する。一方で、衝突困難性を必要とする場合<sup>23</sup>は誕生日攻撃によりハッシュ長の半分となる。

共通鍵暗号やハッシュ関数に対してはGroverのアルゴリズムを使うことで少し効率的に攻撃できる。Groverのアルゴリズムは未整列のデータベースから特定のデータを探索するためのアルゴリズムである。 $n$ 個のデータからなる未整列のデータベースから特定のデータを見つける場合、古典アルゴリズムの計算量は $O(n)$ である。これに対し、Groverのアルゴリズムでは $O(\sqrt{n})$ の計算量で十分である。一方で、 $\Omega(\sqrt{n})$ の計算量であることも示されている [29]ため、Groverのアルゴリズムは未整列のデータベースに対する探索アルゴリズムとしては定数倍の違いを除いて最速である。

Groverのアルゴリズムを共通鍵暗号やハッシュ関数への攻撃に適用した場合、セキュリティ強度が概ね半減することになる。一方で、他に効率の良い攻撃が無い場合は半減程度で済む。つまり、量子コンピュータへの対応としては鍵長やハッシュ長を倍にすることでこれまでと同じ安全性を確保することができる。そのため、RSA暗号や楕円曲線暗号等と比べてもまだ安全性を保っていると考えられる。但し、個別の共通鍵暗号やハッシュ関数については、他の古典アルゴリズムや量子アルゴリズムによって危殆化するリスクは依然として存在することには留意しなければならない。

<sup>21</sup> 同じハッシュ値となる異なるデータを見つけることが困難であること。

<sup>22</sup> 与えられたハッシュ値を出力するデータを見つけることが困難であること。

<sup>23</sup> たとえば、メッセージダイジェスト。同じメッセージダイジェストを生成する二つの文書の一方に署名した場合、その署名はもう一方の文書で使用しても有効となる。

## Appendix B. 選定された耐量子計算機暗号の概要

本項では、標準化対象となった CRYSTALS-KYBER、CRYSTALS-Dilithium、FALCON、SPHINCS<sup>+</sup>と第4ラウンドに進出した BIKE、Classic McEliece、HQC、SIKE の概要と NIST による定性的な評価 [19]を紹介する。

### (1) CRYSTALS-KYBER

CRYSTALS-KYBER は加群 (module) 格子を利用した鍵カプセル化メカニズムで、[30]で提案された。

最初に格子ベースの暗号を提案したのは Ajtai and Dwork [31]である。その後、Regev [32], [33]で改良されており、中でも特筆すべきは LWE (learning with errors) 問題の導入である<sup>24</sup>。LWE 問題は整数<sup>25</sup>上で考える問題であったが、多項式環上での一般化が考案され、Ring-LWE 問題が定式化された。多項式環の次元が大きいくほど送信可能なビットが増えることから、Ring-LWE 問題の困難性に基づく暗号は LWE 問題の困難性に基づく暗号より効率的である。その後、Brakerski, Gentry, and Vaikuntanathan [34]<sup>26</sup>, Langlois and Stehlé [35]での更なる一般化によって Module-LWE 問題が定式化された。

CRYSTALS-KYBER は Module-LWE 問題の困難性に基づいている。効率という面では Ring-LWE 暗号と同程度である。一方、Ring-LWE 暗号と異なり、セキュリティパラメータの変更に再実装コストがかかる環の次元を伴わないことから柔軟性があるとしている。

NIST はパフォーマンスに関して、サイズは殆どのアプリケーションでは許容範囲とし、速度は鍵生成・カプセル化・カプセル化解除が高速であると評している。セキュリティに関しては、格子暗号自体が良く研究されていることや提出されたパラメータセットがセキュリティレベルを満たす可能性が高いと評している。加えて Saber や NTRU が依存している Module-LWR (learning with rounding) 問題や NTRU 問題と比較して Module-LWE 問題が良く研究されていることも評価している。

### (2) CRYSTALS-Dilithium

CRYSTALS-Dilithium は Fiat-Shamir パラダイムに基づく加群格子ベースの署名アルゴリズムで、その安全性は Module-LWE 問題の困難性と Module-SIS (small integer solution) 問題の困難性に基づいている。

NIST はパフォーマンスに関して FALCON と比較している。まず、x86-64 や類似のプロセッサでは Dilithium は FALCON と比べて署名生成がやや速い。しかし、コストは署名の生成や検証だけ

<sup>24</sup> Oded Regev はこの功績により 2018 年のゲーデル賞を受賞している。

<sup>25</sup> 厳密には剰余群  $\mathbb{Z}/p\mathbb{Z}$ 。

<sup>26</sup> Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan も 2022 年にゲーデル賞を受賞している。

ではなく、公開鍵と署名の送信にも留意しなければならない。公開鍵や署名のサイズは FALCON の方が小さいため、送信も考慮した総コストという点では FALCON の方が小さくて済む。特に、大きな署名サイズに対応することが困難な場合は FALCON が好まれるかもしれないとしている。

一方、FALCON は浮動小数点演算があり、Dilithium にはないという違いがある。そのため、浮動小数点演算をサポートしていない ARM Cortex-M4 では FALCON の署名生成速度がかなり遅くなり、結果として送信を考慮しても Dilithium の方がコストを抑えられる。また、Dilithium はいくつかのハードウェア実装が開発されているが FALCON はそうではなく、FALCON は Dilithium より多くの RAM を必要とすることから、スマートカード等の制約がある場合には FALCON の実装が困難であると指摘されている。

以上のことから、以前は Dilithium と FALCON のいずれか一方を標準する意向を示していたが、両方を標準化することとした。

### (3) FALCON

FALCON (Fast Fourier Lattice-based Compact Signatures over NTRU) は Hash-and-Sign パラダイムを利用した NTRU 格子ベースの署名スキームで、その安全性は SIS 問題の困難性に基いている。

素因数分解問題等のよく知られた困難な計算問題は最悪時困難性であるが、暗号で利用する場合は平均的に困難である方が望ましい。また、一般に最悪ケースのインスタンスを作成することは難しい。Ajtai [36]は格子問題に関してこの平均時困難性と最悪時困難性を関連付けた。これによりいくつかの最悪時困難性に基づく格子ベースの暗号が考案されることとなったが、そのような暗号プリミティブは一方向き及び衝突耐性があるハッシュ関数か公開鍵暗号に限られており、安全で効率的で単純なデジタル署名を構築することは未解決であった。そこで、Gentry, Peikert, and Vaikuntanathan [37]は格子が自然に持つトラップドアを利用してデジタル署名等を構築する方法を示した。FALCON はこの構築を NTRU 格子上で行い、トラップドアのサンプリングに高速 Fourier サンプリングを使用している<sup>27</sup>。

NIST による評価は Dilithium を参照のこと。

### (4) SPHINCS<sup>+</sup>

SPHINCS<sup>+</sup>はステートレスなハッシュ関数ベースの署名アルゴリズムである。

最初にハッシュ関数ベースの署名アルゴリズムを提案したのは Lamport [38]である。この Lamport 署名には署名鍵をたった一度しか使えないという制約があった。その後、多くのメッセージで署名するために Merkle [39]で Merkle tree が考案された。それに続いて、様々なハッシ

<sup>27</sup> <https://falcon-sign.info/>, accessed August 2024.

署名が考案されたが、基本的にステートフルな署名方式であり、状態を管理する必要があるため不便であった。その中で Goldreich [40]はステートレスな署名方式を提案し、Bernstein, et al. [41]はそのアイデアを取り入れて SPHINCS を考案した。SPHINCS+はこの SPHINCS を改良したものである<sup>28</sup>。

NIST はパフォーマンスに関して、公開鍵のサイズは非常に短いが署名のサイズは非常に長いとし、鍵生成と検証の速度は署名よりはるかに高速と評している。セキュリティに関しては、設計の複雑さが潜在的な問題となる可能性を指摘しつつも、ハッシュ関数の安全性にのみ依存することから、CRYSTALS-Dilithium や FALCON 等に対する予期しない攻撃が発見された際のバックアップとして有用であるとしている。

## (5) BIKE

BIKE (Bit Flipping Key Encapsulation) はバイナリ線形 QC-MDPC (quasi-cyclic moderate density parity check) 符号を利用した鍵カプセル化メカニズムで、Misoczki, et al. [42]で提案された。

最初に誤り訂正符号を利用した暗号を提案したのは McEliece [43]で、Shor のアルゴリズムよりも前である。この McEliece 暗号は、暗号文である符号とランダムな符号を識別する問題の困難性と線形符号の復号<sup>29</sup>問題の困難性に基いている。特に後者は NP 困難であることも示されている [44]。McEliece 暗号の問題は効率的である一方、鍵サイズが非常に大きくなるという問題がある。準巡回符号を利用することで鍵サイズの削減は可能であり、実際にいくつか提案されたものの、多くは構造的代数攻撃によって破られている。そこで、BIKE は代数的構造を持たない LDPC (low-density parity check) 符号にいくつかの工夫を加えた MDPC 符号を利用した。この工夫の採用は誤り訂正の能力は著しく低下させるが、安全性を線形符号の復号問題に帰着できることからセキュリティに関するメリットは大きい。

NIST はパフォーマンスに関して、サイズは構造化格子による鍵カプセル化メカニズムと同程度とする一方で、速度は鍵生成やカプセル解除で遅く実行されると評している。セキュリティに関しては、一般に復号失敗率が十分低くなければ鍵回復攻撃に繋がると述べたうえで、復号失敗率の上限が示されていないことを指摘している。そして、格子ベースでない暗号としては最も競争力があるとしている。

## (6) Classic McEliece

Classic McEliece は McEliece 暗号を利用した鍵カプセル化メカニズムである。

<sup>28</sup> <https://huelsing.net/wordpress/?p=558>, accessed August 2024.

<sup>29</sup> 暗号における復号は通常 decrypt のことであるが、符号ベースの場合は decode を使うことが一般的である。

McEliece 暗号自体はあまり安全ではない OW-CPA となるように設計されている。この McEliece 暗号に Niederreiter によって考案されたデュアルPKEや IND-CCA2 への変換を適用することで、量子ランダムオラクルモデルに対して耐性のある暗号となっている<sup>30</sup>。

NIST はパフォーマンスに関して、まず公開鍵のサイズが非常に大きいことと鍵生成の速度がかなり遅いことを指摘している。一方で、暗号文サイズは耐量子計算機暗号候補の中で最も小さいことも指摘している。そのため、公開鍵が再利用されて通信の度に再送信する必要が無い場合や非常に小さな暗号文を必要とする場合には有用であると示唆している。また、セキュリティ面は自信がある (confident) と評している。

## (7) HQC

HQC (Hamming Quasi-Cyclic) は QC-MDPC 符号を利用した鍵カプセル化メカニズムで、Aguilar, et al. [45] で提案された。

BIKE の項目で述べている通り、符号に代数的構造がある暗号はそこから破られている。BIKE は工夫をしているものの、Aguilar, et al. [45] は隠された構造が明かされることによって破られる可能性が常にあることを指摘している。この隠された構造の問題に対し、Alekhovich [46] はランダムな符号語にランダムな誤差ベクトルを追加したものを秘密鍵として採用するアイデアを提案した。この着想は Ajtai-Dwork 暗号から部分的に影響を受けている。Ajtai-Dwork 暗号以降の格子暗号の発展は符号暗号にも模倣され、LPN (learning parity with noise) 問題が提案されている。HQC はランダムな準巡回符号の復号の困難性に基づいた暗号で、鍵回復攻撃に繋がる復号失敗率を正確に解析しやすい誤差項を与えられる。

NIST はパフォーマンスに関して、公開鍵と暗号文のサイズは準巡回構造によって小さくなるものの格子ベースの鍵カプセル化メカニズムや BIKE と比較して大きいのが、鍵生成やカプセル化解除の速度は BIKE より速いとし、全体として許容範囲ではあるものの最適ではないと評している。セキュリティに関しては、各セキュリティカテゴリで復号失敗率が異なるが、それぞれ攻撃を回避できる程度に十分小さいとしている。

## (8) SIKE

SIKE (Supersingular Isogeny Key Encapsulation) は超特異楕円曲線を利用した鍵カプセル化メカニズムで、Jao and De Feo [47] 及び Jao, De Feo, and Plût [48] で提案された。

DH 鍵共有は有限体で考えられていたが、その後、楕円曲線を利用した ECDH 鍵共有が考案された。Stolbunov [49] は楕円曲線間の同種写像の計算の困難性に基づく鍵カプセル化メカニズムを提案しようとした。しかし、鍵交換に 229 秒要することに加え、Childs, Jao, and Soukharev [50] により、一般化 Riemann 予想を仮定はしているものの、準指数時間で解読する量子アルゴ

<sup>30</sup> <https://classic.mceliece.org/>, accessed August 2024.

リズムを提示された。SIKE は超特異楕円曲線間の同種写像を使用してそれらの問題の克服を試みたものである。

NIST からは高価な計算を指摘されながらも通信コストの低さや ECDH 鍵共有との相性の良さから期待されていたが、Castryck and Decru [15]によってその安全性が破られた。

## 引用文献

- [1] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, D. Smith-Tone, “Report on Post-Quantum Cryptography,” National Institute of Standards and Technology, Gaithersburg, MD, 2016.
- [2] Department of Commerce; NIST, *Request for Comments on Post-Quantum Cryptography Requirements and Evaluation Criteria*, 2016.
- [3] M. Mosca, “Cybersecurity in an era with quantum computers: will we be ready?,” IEEE, 2018.
- [4] NIST, “NIST Asks Public to Help Future-Proof Electronic Information,” 20 12 2016. Available: <https://www.nist.gov/news-events/news/2016/12/nist-asks-public-help-future-proof-electronic-information>. [アクセス日: 8 2024].
- [5] NIST, “NIST Reveals 26 Algorithms Advancing to the Post-Quantum Crypto ‘Semifinals’,” 30 1 2019. Available: <https://www.nist.gov/news-events/news/2019/01/nist-reveals-26-algorithms-advancing-post-quantum-crypto-semifinals>. [アクセス日: 8 2024].
- [6] NIST, “PQC Standardization Process: Second Round Candidate Announcement,” 30 1 2019. Available: <https://csrc.nist.gov/News/2019/pqc-standardization-process-2nd-round-candidates>. [アクセス日: 8 2024].
- [7] NIST, “PQC Standardization Process: Third Round Candidate Announcement,” 22 7 2020. Available: <https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement>. [アクセス日: 8 2024].
- [8] NIST, “PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates,” 5 7 2022. Available: <https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4>. [アクセス日: 8 2024].
- [9] NIST, “Comments Requested on Three Draft FIPS for Post-Quantum Cryptography,” 24 8 2023. Available: <https://csrc.nist.gov/News/2023/three-draft-fips-for-post-quantum-cryptography>. [アクセス日: 8 2024].
- [10] Department of Commerce; NIST, *Announcing Issuance of Federal Information Processing Standards (FIPS) FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard, FIPS 204, Module-Lattice-Based Digital Signature Standard, and FIPS 205, Stateless Hash-Based Digital Signature Standard*, 2024.
- [11] NIST, “Security (Evaluation Criteria),” 3 1 2017. Available: [https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-\(evaluation-criteria\)](https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria)). [アクセス日: 8 2024].
- [12] NIST, “Cost (Evaluation Criteria),” 3 1 2017. Available: [https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/cost-\(evaluation-criteria\)](https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/cost-(evaluation-criteria)). [アクセス日: 8 2024].
- [13] NIST, “Algorithm and Implementation Characteristics (Evaluation Criteria),” 3 1 2017. Available: <https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/algorithm-and-implementation-characteristics>. [アクセス日: 8 2024].
- [14] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, J. Kelsey, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone, “Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization

- Process,” National Institute of Standards and Technology, Gaithersburg, MD, 2020.
- [15] W. Castryck, T. Decru, “An Efficient Key Recovery Attack on SIDH,” Springer, 2023.
- [16] E. Kani, “The number of curves of genus two with elliptic differentials,” de Gruyter, 1997.
- [17] IEEE Spectrum, ““Quantum-Safe” Crypto Hacked by 10-Year-Old PC,” 19 8 2022. Available: <https://spectrum.ieee.org/quantum-safe-encryption-hacked>. [アクセス日: 8 2024].
- [18] W. Beullens, “Breaking Rainbow Takes a Weekend on a Laptop,” Springer, 2022.
- [19] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone, “Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process,” National Institute of Standards and Technology, Gaithersburg, MD, 2022.
- [20] R. Barnes, K. Bhargavan, B. Lipp, C. A. Wood, “Hybrid Public Key Encryption,” 2022.
- [21] F. Driscoll, M. Parsons, “Terminology for Post-Quantum Traditional Hybrid Schemes,” 2024.
- [22] 伊藤忠彦, 宇根正志, 清藤武暢, “量子コンピュータによる脅威を見据えた暗号の移行対応,” 2019.
- [23] L. Valenta, V. Gonçalves, B. Westerbaan, “NIST’s first post-quantum standards,” 20 8 2024. Available: <https://blog.cloudflare.com/nists-first-post-quantum-standards/>. [アクセス日: 9 2024].
- [24] IBM, “Cryptography Bill of Materials,” 29 1 2023. Available: <https://github.com/IBM/CBOM>. [アクセス日: 8 2024].
- [25] IBM, “Sonar Cryptography Plugin,” 31 7 2024. Available: <https://github.com/IBM/sonar-cryptography>. [アクセス日: 8 2024].
- [26] E. Barker, “Recommendation for Key Management: Part 1 - General,” National Institute of Standards and Technology, Gaithersburg, MD, 2020.
- [27] CRYPTREC, “暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準,” 2022.
- [28] CRYPTREC, “素因数分解問題調査研究報告書,” 2001.
- [29] C. H. Bennett, E. Bernstein, G. Brassard, U. Vazirani, “Strengths and Weaknesses of Quantum Computing,” 1997.
- [30] 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, 2018.
- [31] M. Ajtai, C. Dwork, “A public-key cryptosystem with worst-case/average-case equivalence,” Association for Computing Machinery, New York, NY, 1997.
- [32] O. Regev, “New lattice-based cryptographic constructions,” Association for Computing Machinery, New York, NY, 2004.
- [33] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” Association for Computing Machinery, New York, NY, 2009.
- [34] Z. Brakerski, C. Gentry, V. Vaikuntanathan, “(Leveled) fully homomorphic encryption without bootstrapping,” Association for Computing Machinery, New York, NY, 2012.

- 
- [35] A. Langlois , D. Stehlé, “Worst-case to average-case reductions for module lattices,” Springer, 2015.
- [36] M. Ajtai, “Generating Hard Instances of Lattice Problems (Extended Abstract),” Association for Computing Machinery, New York, NY, 1996.
- [37] C. Gentry, C. Peikert , V. Vaikuntanathan, “Trapdoors for hard lattices and new cryptographic constructions,” Association for Computing Machinery, New York, NY, 2008.
- [38] L. Lamport, “Constructing digital signatures from a one way function,” SRI International Computer Science Laboratory, 1979.
- [39] R. Merkle, “A certified digital signature,” Springer, New York, NY, 1990.
- [40] O. Goldreich, Foundations of Cryptography: Volume 2, Basic Applications, Cambridge, UK: Cambridge University Press, 2004.
- [41] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider , Z. Wilcox-O’Hearn, “SPHINCS: Practical Stateless Hash-Based Signatures,” Springer, Berlin, Heidelberg, 2015.
- [42] R. Misoczki, J.-P. Tillich, N. Sendrier , P. S. L. M. Barreto, “MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes,” IEEE, 2013.
- [43] R. J. McEliece, “A public-key cryptosystem based on algebraic coding theory,” 1978.
- [44] E. R. Berlekamp, R. J. McEliece , H. C. A. van Tilborg, “On the inherent intractability of certain coding problems,” IEEE, 1978.
- [45] C. Aguilar, O. Blazy, J.-C. Deneuville, P. Gaborit , G. Zémor, “Efficient Encryption From Random Quasi-Cyclic Codes,” IEEE, 2018.
- [46] M. Alekhnovich, “More on average case vs approximation complexity,” IEEE, 2003.
- [47] D. Jao , L. De Feo, “Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies,” Springer, Berlin, Heidelberg, 2011.
- [48] L. De Feo, D. Jao , J. Plût, “Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies,” de Gruyter, 2014.
- [49] A. Stolbunov, “Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves,” American Institute of Mathematical Sciences, 2010.
- [50] A. Childs, D. Jao , V. Soukharev, “Constructing elliptic curve isogenies in quantum subexponential time,” de Gruyter, 2013.