IMES DISCUSSION PAPER SERIES

量子コンピュータの解読に耐えうる暗号アルゴリズム「格子暗号」の最新動向

清藤武暢（せいとうたけのぶ）・青野良範（あおのよしのり）・四方順司（しかたじゅんじ）

Discussion Paper No. 2015-J-9

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES

BANK OF JAPAN

日本銀行金融研究所

〒103-8660 東京都中央区日本橋本石町 2-1-1

IMES Discussion Paper Series 2015-J-9
2015 年 7 月

量子コンピュータの解読に耐えうる暗号アルゴリズム「格子暗号」の最新動向

清藤武暢(せいとうたけのぶ)*・青野良範(あおのよしのり)**・四方順司(しかたじゅんじ)***

要　旨

金融分野では、各取引におけるデータの安全性を確保するために、公開鍵暗号等の暗号アルゴリズムが広く利用されている。公開鍵暗号は、公開鍵から秘密鍵を求めることが困難であるという仕組みによりその安全性を保証しており、この仕組みの実現には数学的な問題が利用されている。しかし、量子力学の性質を演算処理に応用した「量子コンピュータ」が実現すると、現在主流の公開鍵暗号（RSA 暗号等）が安全性の根拠とする数学的問題が容易に解かれることが知られており、その安全性を確保できなくなるという潜在的な脅威が存在する。現時点では、量子コンピュータはまだ広く利用可能な状態ではないため、RSA 暗号等が直ちに危殆化する状況にあると考えられているわけではないが、既に利用されている暗号アルゴリズムの移行には、綿密な長期計画が必要となることが多い。このため、量子コンピュータの実用化を予め見据えたうえで、同コンピュータに対して安全性を確保できる暗号アルゴリズム「耐量子コンピュータ暗号」の準備を今から進めていくことは重要である。こうした状況下、耐量子コンピュータ暗号の 1 つである「格子暗号」は、データを暗号化した状態のまま処理できる技術（「暗号化状態処理技術」）を実現できるという特長を有することもあって、近年、研究が活発化している。一方で、同暗号の原理や安全性の根拠となる数学的問題が複雑であるため、馴染みのない企業や組織等が、将来的に利活用するためには各種ハードルがあると考えられる。そこで、本稿では、格子暗号の概要や実用化動向を中心に紹介するとともに、安全に利用する際の留意点や同暗号の研究動向等について説明する。

キーワード：公開鍵暗号、RSA 暗号、楕円曲線暗号、量子コンピュータ、耐量子コンピュータ暗号、格子暗号

JEL classification: L86、L96、Z00

*　日本銀行金融研究所（E-mail:takenobu.seitou@boj.or.jp）
**　国立研究開発法人情報通信研究機構（E-mail:aono@nict.go.jp）
***　横浜国立大学大学院環境情報研究院（E-mail:shikata@ynu.ac.jp）

本稿の作成に当たっては、九州大学の安田雅哉准教授から有益なコメントを頂いた。ここに記して感謝したい。ただし、本稿に示されている意見は、筆者たち個人に属し、日本銀行、国立研究開発法人情報通信研究機構あるいは横浜国立大学の公式見解を示すものではない。また、ありうべき誤りはすべて筆者たち個人に属する。

目 次

1. はじめに

金融分野では、各種金融取引の安全性を確保するために、共通鍵暗号や公開鍵暗号等の暗号アルゴリズムが広く利用されている。例えば、銀行 ATM とホストコンピュータ間でやり取りされるデータ（暗証番号や口座番号等）の機密性や完全性の確保、インターネット・バンキングにおける通信相手の認証等で活用されている。データの機密性を確保する手段としては主に共通鍵暗号が用いられ、AES（Advanced Encryption Standard）等が利用されている。一方、データの完全性を確保したり、通信相手を認証したりする手段としては主に公開鍵暗号が用いられ、RSA 暗号や楕円曲線暗号等が現在主流の公開鍵暗号として幅広く普及している。

一般に、公開鍵暗号は「理論的には問題の解を演算処理により求めることは可能だが、現実的には、たとえ現時点で最も高性能なスーパーコンピュータを利用したとしても、膨大な時間（数十億年程度）を要する演算処理が必要となるため、事実上解くのが難しい」という特徴を有する数学的問題を安全性の根拠として利用している。現在主流の RSA 暗号と楕円曲線暗号においては、それぞれ「素因数分解問題」、「楕円曲線離散対数問題」と呼ばれる上記の特徴を有する数学的問題を利用している。

しかし、量子力学[1]の性質を演算処理に応用したコンピュータである「量子コンピュータ（特に「量子デジタル型」と呼ばれる実装方法[2]）」が実現すると、素因数分解問題や楕円曲線離散対数問題が現実的な時間内で容易に解けることが知られている。そのため、同コンピュータを利用可能な攻撃者に対しては、RSA 暗号や楕円曲線暗号はその安全性を確保できないという潜在的な脅威が存在する。ただし現時点では、量子デジタル型の量子コンピュータは、演算処理において重要な役割を果たす「重ね合わせ状態」[3]を維持することが技術的に難しいため、実用化には相当の時間がかかる可能性が高く、RSA 暗号等が直ちに危殆化する状況にあると考えられているわけではない。しかしながら、既に利用されている暗号アルゴリズムを切り替えるためには、綿密な長期計画に基づく時間をかけた移行が必要となることが多い。例えば、2005 年に米国立標準技術研究所（National Institute of Standards and Technology、NIST）が、安全性の低下を主な理由に、当時主流で利用されていた暗号アルゴリズム（2-Key トリプル DES や短い鍵長の RSA 暗号等）を、2011 年以降、米国連邦政府のシステムで使用しない方針を各種ガイドラインにて示し、より安全性の高い暗号アルゴリズムへ

[1] 量子力学とは、物質の構成単位である原子の内部構造のような極めて微細な世界における物理現象を対象とする理論。

[2] 量子コンピュータには、「量子デジタル型（量子ゲート型）」と「量子アナログ型（量子イジングマシン型）」という 2 種類の実装方法が存在する（詳細は **2 節**参照）。

[3] 量子力学における複数の状態が同時に存在する性質（詳細は **2 節**参照）。

の移行を促した（NIST[2005a,b]、詳細は宇根・神田[2006]参照）。これを受けて、金融分野においても、2007 年に金融サービスを対象とする国際標準化機構（ISO）の専門委員会（ISO/TC68）により、暗号アルゴリズムの移行に関する推奨対応策が示された（ISO[2007]、詳細は田村[2009]参照）。その結果、国内の金融機関では、順次移行作業が進められたが、2010 年末までに完全に移行することはできなかった（2010 年 8 月時点で、調査対象 117 台の金融機関サーバのうち約 60 台が未達、武藤[2011]）。暗号アルゴリズムの移行対応に 3 年以上を要したという事実を踏まえると、量子コンピュータの実用化を予め見据えたうえで、量子コンピュータによる解読に耐えうる暗号アルゴリズム（以下、「耐量子コンピュータ暗号」と呼ぶ）の準備を今から進めていくことは、将来の不測の事態への備えとして重要であると考えられる。

これまで、いろいろな種類の耐量子コンピュータ暗号が提案されているが、その 1 つに「格子暗号」と呼ばれる公開鍵暗号がある。格子暗号は「格子点探索問題」と呼ばれる数学的問題を利用する公開鍵暗号の総称であり、量子コンピュータでも容易に解読できないと期待されており、耐量子コンピュータ暗号の有力な候補の 1 つとして学界で注目されている。さらに、データを暗号化した状態のまま処理できる技術（「暗号化状態処理技術」と呼ばれる）を実現できるという特長も有する。格子暗号は、これまでに様々な実現方式が提案されているが、その中でも「LWE（Learning with Errors）方式」と呼ばれる実現方式は、安全性と実用性のバランスの観点から、現時点では最も優れていると考えられており、近年、研究が活発化している。特に、クラウドサービスや医療分野などへの同方式の応用に関する研究は盛んに行われており、一部については製品化も始まっている[4]。

このように、格子暗号は従来の公開鍵暗号と比較したとき、量子コンピュータへの耐性や暗号化状態処理技術を実現可能といった特長を有する一方で、同暗号の原理や安全性の根拠が複雑であるため、同技術に馴染みのない企業や組織等が、将来的に利活用するためには各種ハードルがあると考えられる。そこで、本稿では耐量子コンピュータ暗号として格子暗号に注目し、同暗号の概要や実用化動向を中心に解説するとともに、同暗号の代表な実現方式である LWE 方式を安全に利用する際の留意点や最近の研究動向等について紹介する。

以下、本稿では、2 節において量子コンピュータが現在主流の公開鍵暗号の安全性に与える影響について概説したのち、3 節において耐量子コンピュータおよび格子暗号の概要や実用化動向等について紹介する。4 節において、格子暗号の代表的な実現方式である LWE 方式について、その仕組み、安全に利用するための留意点、および最近の研究動向について概説する。

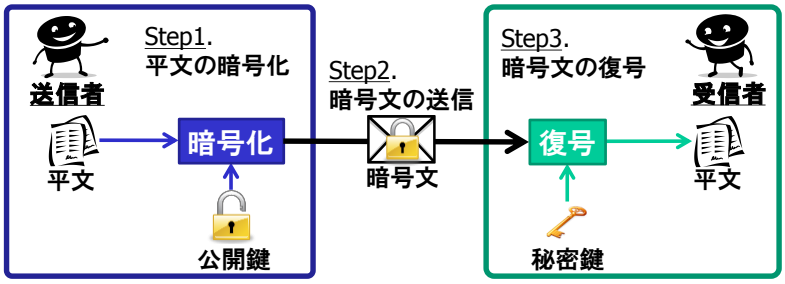
[4] 暗号化状態処理技術の概要について清藤・四方[2014]参照。

2. 現在主流の公開鍵暗号の潜在的な脅威

(1) 公開鍵暗号の概要

金融分野では、各取引においてやり取りされるデータの安全性を確保するために暗号アルゴリズムが広く利用され、公開鍵暗号は主に「データの完全性確保」、「通信相手の認証」、「共通鍵暗号において利用する鍵の共有（鍵共有）」等の機能を実現する手段として用いられている[5]。

公開鍵暗号は、データの暗号化に用いる鍵（以下、「公開鍵」と呼ぶ）と復号に利用する鍵（以下、「秘密鍵」と呼ぶ）が異なり、秘密鍵は各利用者が秘密に保管する必要がある一方、公開鍵は全ての利用者に対して公開できるという特長を有する暗号アルゴリズムである。同暗号は、事前に通信相手へ鍵を秘密に渡しておく必要がないため、例えば、インターネットのような不特定多数の利用者が存在するサービスにおける安全な通信路の確立等に適している。公開鍵暗号を用いて「送信者」がデータ（平文）を暗号化して暗号文を生成し、それを受信した「受信者」が暗号文から平文を復号する処理フローを図表 1 に示す。なお、同フローにおいては、受信者は予め公開鍵と秘密鍵を生成したうえで公開鍵を公開し、送信者は平文の暗号化を行う際、この公開鍵を入手しているものとする。



図表 1. 公開鍵暗号の処理フロー

一般に、公開鍵暗号では「公開鍵から秘密鍵を現実的な時間で求めることが難しい」ことを最低限満たすべき安全性要件としている。これは、少なくとも受信者以外の秘密鍵を持たない利用者（攻撃者）は、たとえ公開鍵と暗号文を両方とも入手したとしても、正しい平文を復号できないことを保証するためである。公開鍵暗号においては、ある種の数学的問題を用いてこの安全性の仕組みを実現している。公開鍵暗号で利用対象とされる数学的問題は、理論的には

[5] データの完全性確保や通信相手の認証については、公開鍵暗号の仕組みを利用して実現される暗号アルゴリズム「電子署名」が用いられるが、本稿では同アルゴリズムの説明については省略する。

同問題の解を演算処理により求めることは可能だが、現実的にはたとえ現時点で最も高性能なスーパーコンピュータを利用したとしても膨大な時間（数十億年程度）を要する演算処理が必要となるため、事実上解くのが難しいという特徴を有する。このような問題を利用して、「ある演算を行うことは簡単」だが、「その逆の演算（「逆演算」と呼ばれる）を行うのは難しい」という演算の一方向性の仕組みを実現することにより、上記の安全性要件を満たす公開鍵暗号を実現できる。ただし、コンピュータの演算処理性能の向上や攻撃手法の進歩等により、公開鍵暗号で利用される数学的問題を解く難しさ（すなわち、公開鍵暗号の安全性）は経年劣化するため、安全性を確保するためには同問題の難しさを決定する「鍵長（公開鍵や秘密鍵の桁数）」と呼ばれるパラメータを適宜長くしていくという運用が必要となる。

(2) 現在主流の RSA 暗号と楕円曲線暗号の概要および実用化動向

RSA 暗号は、初めての実用的な公開鍵暗号の実現方式として 1977 年に発表された（Rivest, Shamir, and Adleman[1978]）。同暗号に関しては、提案以降、安全性評価に関する研究が盛んに行われており、現在のコンピュータ環境のもとでは効率の良い解読手法が見つかっていないことや、同暗号に係る特許が 2000 年 9 月に切れたこと等から、急速に普及が進んでいった。しかし、最近、安全性の経年劣化への対応に伴う鍵長の増加により、いずれ従来のハードウェア環境での実装が難しくなる等の実用上の支障が出てくるとの指摘がある。特に、CPU の処理性能やメモリ容量等が制限された IC カードや組込み機器[6]等において RSA 暗号を利用する際に、同問題はより顕著となる。

楕円曲線暗号は、1980 年代に入ってミラーとコブリッツによりそれぞれ独立に提案された公開鍵暗号の実現方式である（Miller[1985]、Koblitz[1987]）。同暗号は、RSA 暗号と比較して 10 分の 1 程度の短い鍵長で同程度の安全性を実現できる等のメリットを有すること[7]や、最近になって学界における安全性評価の研究も進展していること等から、RSA 暗号に代わる主流の公開鍵暗号として、急速に普及が進みつつある。特に、同暗号は IC カードや組込み機器等のような CPU の処理性能やメモリ容量等が制限された環境での利用にも適した公開鍵暗号として注目されている。

RSA 暗号と楕円曲線暗号は、インターネット・バンキングをはじめとする様々

[6] 組込み機器は、特定の機能を実現するためのコンピュータ・システムが内蔵されている家電（デジタルテレビ等）や産業用機械の総称である。同機器は、一般的に生産コスト等の制約から計算能力やメモリ等に強い制約がある。

[7] この利点は、楕円曲線暗号と RSA 暗号では、安全性の根拠とする数学的問題の難しさが異なることに由来する（詳細は清藤・四方[2013]参照）。

な分野において、広く利用されている暗号通信仕様である「SSL/TLS[8]」において、利用可能な公開鍵暗号として規定されている。また、国際クレジットカード・デビットカードの業界標準である「EMV 仕様[9]」においても、IC カードを利用した取引環境の安全性を確保することを目的に、現在 RSA 暗号が採用されている。さらに、安全性および処理効率のさらなる向上を目指すために、RSA 暗号から楕円曲線暗号への移行等に関する指針が示されており（EMVCo[2009]）、同指針に基づき利用環境の整備が進められている。そのほか、国内外の金融分野における情報セキュリティ技術の国際標準（ISO、IEEE、ANSI、FIPS 等）、業界標準仕様（全銀協 IC キャッシュカード標準仕様等）や各種ガイドライン（CRYPTREC 電子政府推奨暗号リスト等）においても、RSA 暗号と楕円曲線暗号は主流の公開鍵暗号として位置づけられている。

(3) 量子コンピュータによる公開鍵暗号の解読

前述のとおり、公開鍵暗号は数学的問題を用いて安全性の仕組みを実現しており、現在主流の RSA 暗号と楕円曲線暗号は、それぞれ「素因数分解問題」と「楕円曲線離散対数問題」と呼ばれる数学的問題を利用している（図表 2）。

数学的問題の名称	概要
素因数分解問題	2 つの素数P、Qの合成数「$N = P \times Q$」から、PとQを求める問題。
楕円曲線離散対数問題	「楕円曲線」と呼ばれる特殊な曲線上の 2 点GとTから「$T = s \times G$」を満たす自然数sを求める問題。

図表 2. RSA 暗号と楕円曲線暗号で利用される数学的問題

これらの数学的問題は、現在のコンピュータ環境では現実的な時間で解くのが難しいと期待されており、RSA 暗号や楕円曲線暗号の安全性の根拠となって

[8] 同仕様は、はじめは 1990 年代に Netscape Communications 社により独自仕様「SSL（Secure Socket Layer）」として発表され、その後、IETF（International Engineering Task Force）により、この SSL をベースに変更が加えられ、インターネットの技術標準 RFC（Request For Comments）において「TLS（Transport Layer Security）」として規定されている。なお、RFC を策定する IETF は、インターネットにおける技術上の諸問題を解決することを目的として設置された委員会（Internet Architecture Board）の下部組織。

[9] EMV 仕様は、IC カードの利用を前提にクレジットカード・デビットカードのビジネスリスク管理の高度化を目的とし、IC カード内での暗号処理をも含めた仕様として、1996 年に公表された。ここで、EMVCo とは、国際的なクレジットカードブランド（Europay International、MasterCard International、Visa International）により設立された組織であり、EMV 仕様の管理運営を行っている。

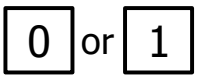
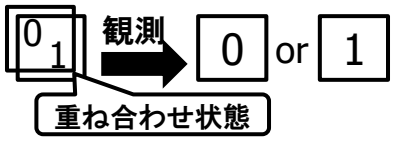
いる。しかし、量子力学の性質を演算処理に応用したコンピュータである量子コンピュータ（特に、量子デジタル型）が実現すると、たとえ鍵長を長くしたとしても、素因数分解問題や楕円曲線離散対数問題を容易に解けることが知られている（Shor[1994,1997]、四方・鈴木・今井[1999]）。そのため、将来、同コンピュータが実現すると RSA 暗号や楕円曲線暗号はその安全性を確保できないという潜在的な脅威が存在する。

イ．量子コンピュータ（量子デジタル型）の仕組み

量子コンピュータは、実装方式の違いにより「量子デジタル型（量子ゲート型）」と「量子アナログ型（量子イジングマシン型）[10]」に分類される。本稿では、RSA 暗号等の安全性に対する脅威となり得る、量子デジタル型の量子コンピュータに注目し、その概要について説明する。

量子デジタル型の量子コンピュータは、量子力学において「重ね合わせ状態」と呼ばれる複数の状態が同時に存在する性質を演算に利用した、量子コンピュータの実装方式の 1 つである。ここで、コンピュータが扱いやすい 2 進数の世界を考えたとき、従来のコンピュータにおいては、1 つのビットで「0」または「1」のどちらかの状態のみ表現できる一方、量子デジタル型の量子コンピュータでは、重ね合わせ状態を利用することにより、1 つのビット（「量子ビット」と呼ばれる）で「0」と「1」の状態を同時に表現できる。そして、この量子ビットを観測すると、重ね合わせ状態が失われ、0 か 1 のいずれかの状態に定まる（図表 3）。この性質を利用して、従来のコンピュータにおいて繰り返し処理しなければならない問題を、量子デジタル型の量子コンピュータでは 1 度の処理で答えを出すことができ、問題を高速に解くことが可能となる。

[10] 量子アナログ型（量子イジングマシン型）は、解きたい数学的問題をある種の物理問題（「スピングラス問題」）に変換し、特殊な磁石（量子効果の働く磁石）を模した実験装置（「量子イジングマシン」と呼ばれる）を用いて物理問題の実験結果を導き出した後、同結果から元の数学的問題の解を求めるという仕組みに基づくコンピュータである。なお、量子アナログ型は、特定の組合せ最適化問題のみ解くことができる専用機となっており、現時点では素因数分解や楕円曲線離散対数問題を解くのは難しいとされている（概要については、日経 BP 社[2013、2014]等を参照）。なお、カナダの D-Wave Systems 社が実用化し、2011 年に販売を開始したと発表した量子コンピュータは、量子アナログ型に属する（D-Wave[2011]）。また、最近、量子アナログ型の量子コンピュータの処理過程を、従来の半導体技術を用いて擬似的に再現することにより、処理速度の向上や電力消費量を低減が可能な技術が開発されたとの発表があった（日立[2015]）。

従来のコンピュータ	量子コンピュータ
1 ビットで「0」か「1」のどちらかの状態のみ表現。 	1（量子）ビットで「0」と「1」の状態を同時に表現可能。同ビットを観測すると状態が 0 または 1 のどちらかに確定。 

図表 3. 従来のコンピュータと量子コンピュータにおけるビットの概念

例えば、100 ビット演算の場合、従来のコンピュータでは、正しい解を得るために最大で 2^{100} 回（10 進数で 30 桁程度）という膨大な回数の演算処理が必要となる。一方、量子デジタル型の量子コンピュータでは、100 個の量子ビットを用いることにより、1 回で 2^{100} 通りの組み合わせを「同時に」表現可能となり、一度の演算処理で 2^{100} 通りの中から最適な解を導き出すことが可能となる。しかし、2^{100} 通りの中から最適な解を導き出すために、「最適な解以外の要素を波の干渉の原理を利用して打ち消しあう操作を行い、目的の解を導き出す」という特殊な処理が必要となり[11]、そのためのアルゴリズム（以下、「量子アルゴリズム」と呼ぶ）が必要となる。言い換えると、同コンピュータは、アルゴリズムを設計できれば、任意の問題を解くことができる汎用性を有している。

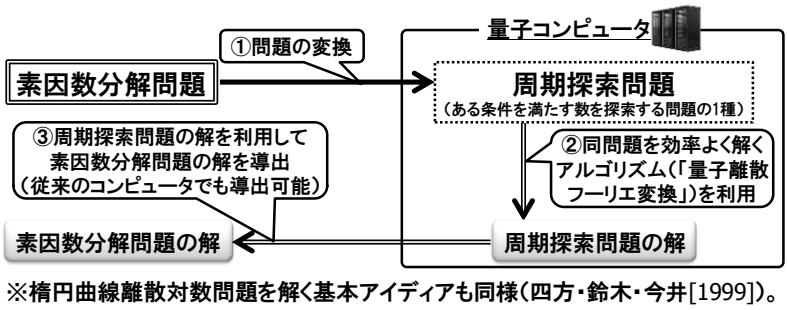
一般に、正しい解を高確率で取り出すことが可能な量子アルゴリズムの設計は難しく、現時点では数個の量子アルゴリズムしか提案されていないが、その 1 つが 1994 年にショアにより発表された「ショアのアルゴリズム」である（Shor[1994,1997]）。同アルゴリズムは、量子デジタル型の量子コンピュータ上で利用することを前提とした量子アルゴリズムであり、RSA 暗号や楕円曲線暗号の安全性の根拠としている数学的問題（素因数分解問題や楕円曲線離散対数問題、図表 2 参照）を容易に解くことができるため、その影響の大きさから学界を中心に注目されてきた。

ロ．ショアのアルゴリズムの影響

ショアのアルゴリズムは、従来のコンピュータでは現実的な時間で解くのが困難な数学的問題（素因数分解問題と楕円曲線離散対数問題）を、①「量子デジタル型の量子コンピュータを利用して効率良く解く方法が知られている数学

[11] 厳密には、重ね合わせ状態となっている「0」と「1」の情報については、観測したときにどちらかの状態に確定する確率（「振幅」と呼ばれる）が与えられており、単純に演算処理を行った場合には各状態は同確率で観測される。量子アルゴリズムは、計算結果から正しい解を観測できる確率をできるだけ高め、それ以外の解候補が観測される確率をできるだけ小さくするような処理が演算処理の過程に組み込まれているアルゴリズムといえる。

的問題（「周期探索問題」と呼ばれる）」に変換し、②同問題を量子コンピュータ上で解いた後、③その結果から元の問題の解を導出するという基本アイディアに基づいている（図表 4）。同アルゴリズムを利用することにより、これらの数学的問題を現実的な時間で解けることが知られている。



図表 4. ショアのアルゴリズムの基本アイディア（イメージ）

もっとも、量子デジタル型の量子コンピュータについては、現時点では実用化には程遠い状況と考えられている。その理由の 1 つは、同コンピュータにおける演算処理において重要な役割を担う量子ビットの重ね合わせ状態が外部からの影響を受けやすく、量子ビットの重ね合わせ状態を維持しつつ演算処理を行うことができる時間が非常に短いことである[12]。この時間が短いと、多くの量子ビットに対して量子アルゴリズムのような複雑な演算処理を行うのが難しくなる。そのため、同コンピュータの実用化については、15（=3×5）や 21（=3×7）といった非常に小さな値（すなわち少ない量子ビット）に対する演算処理を、物理実験で確認するという段階にとどまっているのが現状である（Vandersypen *et al.*[2001]、Martín-López *et al.*[2012]）。こうした開発状況を踏まえると、量子デジタル型の量子コンピュータの実用化には相当の時間がかかる可能性が高く、学界や産業界においても RSA 暗号等が直ちに危殆化する状況にあると考えられているわけではない。もっとも、前述のとおり、既に利用されている暗号アルゴリズムを切り替えるためには、綿密な長期計画に基づく時間をかけた移行が必要であるため、量子コンピュータの実用化を予め見据えたうえで、量子コンピュータによる解読に耐えうる暗号アルゴリズム（耐量子コンピュータ暗号）の準備を進めておくことは重要である。

[12] 前述のとおり、重ね合わせ状態にある量子ビットは観測するとある状態に確定するが、周囲の環境から発生する様々なノイズが、量子ビットに対して観測と同様の影響を与えてしまう（この現象は「デコヒーレンス」と呼ばれる）ため、重ね合わせ状態を長時間維持するのは難しいとされている。このデコヒーレンスが発生する原因について不明な点が多くあるため、同原因を明らかにすることも量子デジタル型の量子コンピュータを実現するうえでの課題の 1 つとなっている。

3. 耐量子コンピュータ暗号と格子暗号

(1) 耐量子コンピュータ暗号

これまで、さまざまな種類の耐量子コンピュータ暗号が提案されているが、これらは暗号アルゴリズムの種類により「計算量的暗号型」と「情報理論的暗号型」に大別される。

イ．計算量的暗号型

計算量的暗号型に分類される耐量子コンピュータ暗号は、「理論的には解読できるが、たとえ量子コンピュータを利用したとしても、現時点では効率的に解くアルゴリズムが見つかっておらず、解読に膨大な時間を要する演算処理が必要なため、事実上解読するのが難しい」という特徴を有する暗号アルゴリズムの総称である。この分類には、「現時点では、量子コンピュータでも現実的な時間で解けないと期待される数学的問題」を安全性の根拠とする公開鍵暗号（図表 5）が含まれる[13]。

暗号アルゴリズムの名称	利用される数学的問題
格子暗号（Regev[2009]等）	格子点探索問題
符号ベース暗号（McEliece[1978]等）	線形符号の復号問題
多変数暗号（Matsumoto and Imai[1988]等）	多変数連立方程式を解く問題

図表 5. 計算量的暗号型に分類される主な公開鍵暗号

ロ．情報理論的暗号型

情報理論的暗号型の耐量子コンピュータ暗号は、「攻撃者に対して、平文の推測に必要となる情報を与えない仕組みを実現することで、たとえ攻撃者が無限の計算能力を利用可能であったとしても、解読が原理的に不可能な安全性を確保する」という特長を有する暗号アルゴリズムの総称である（Shannon[1949]、

[13] なお、共通鍵暗号における量子コンピュータ（量子デジタル型）を利用した攻撃手法は、これまでいくつか提案されており（Grover[1996]、Bennett *et al.*[1997], Brassard, Hoyer , and Tapp[1997], Kuwakado and Morii[2010]）、鍵長がkビットの鍵の探索に要する手間（計算量）を全数探索（2^k程度の計算量が必要）と比べて、「$2^{k/2}$～$2^{k/3}$」程度の計算量に削減できることが知られている。これらの攻撃手法への対策については、現時点では、鍵長を従来の 2～3 倍程度長くすることで、同程度の安全性を確保できるため、RSA 暗号や楕円曲線暗号と比較して、共通鍵暗号では量子コンピュータの影響を受けにくい。ただし、NIST FIPS197 に規定されている共通鍵暗号 AES で利用可能な鍵長は、128 ビット、192 ビット、256 ビットの 3 種類のみとなっているため（FIPS[2001]）、量子コンピュータの実用化を予め見据えると、鍵長の拡張等に向けた検討が今後行われていくかと考えられる。

詳細については Shikata[2015]参照）。そのため、量子コンピュータを利用したとしても、平文や鍵のランダムな推測の確率以上で解読できない。同アルゴリズムの例として、ワンタイムパッド暗号（Vernam[1926]）[14]が挙げられる。もっとも、ワンタイムパッド暗号においては、送信者と受信者間で予め鍵を共有する必要があり、この鍵は少なくとも平文と同じ長さであり、かつ毎回使い捨てにしなければならないため、この鍵をどのようにして安全に共有するかが実装する際の課題となる。この課題を解決する方法の 1 つとして、量子力学の物理的性質を利用して、安全な鍵の共有を実現する「量子鍵配送」と呼ばれる方式[15]が提案されている（Bennett and Brassard[1984]等）。同方式の安全性は量子力学の物理的性質に基づいており、量子コンピュータを用いても破ることは原理的に不可能である。このような量子通信を利用した暗号アルゴリズムは「量子暗号」と呼ばれ、同暗号も耐量子コンピュータ暗号に分類される。

本稿では、近年、学界での研究が活発化している計算量的暗号型の耐量子コンピュータの 1 つである「格子暗号」に着目する。同暗号は、暗号化状態処理技術を実現できるという特長をも有しており、利便性が高いため学界からも注目されている。同じく、耐量子コンピュータ暗号としては、量子暗号についても学界での研究が活発化している[16]。

(2) 格子暗号

イ．格子暗号の特徴および実用化動向

前述のとおり、格子暗号は「格子点探索問題」と呼ばれる数学的問題（詳細は後述）を安全性の根拠として利用する公開鍵暗号の総称である。これまでに提案されている主な実現方式は「AD 方式（Ajtai and Dwork[1997]）」、「GGH 方式（Goldreich, Goldwasser, and Halevi[1997]）」、「NTRU 方式（Hoffstein, Pipher, and Silverman[1998]）」、「LWE 方式（Regev[2009]）」の 4 種類存在する（各方式の詳細は後述）。ここで、格子暗号と RSA 暗号等の現在主流の公開鍵暗号を比較し

[14] ワンタイムパッド暗号では、平文をビット列とみなし、同じ長さのランダムなビット列を鍵として準備したうえで､平文と鍵によるビットごとの排他的論理和をとることで暗号化を行う。暗号文を復号する際には、 暗号化に用いた鍵と暗号文の排他的論理和をとればよい。なお、鍵は毎回使い捨てるため、新たに暗号化を行う際には、別のランダムなビット列を鍵として新たに準備する。

[15] 量子鍵配送は､量子力学における「もとの量子状態を変化させずに観測することはできない」という物理的性質を利用して、送信者と受信者間で安全に鍵を共有するプロトコルである。前述の物理的性質により、送信者が量子通信路を介して受信者に送った情報（光子を用いて実装される）を攻撃者が途中で盗聴した場合、それを検知できる。そのため、盗聴された情報を破棄し、盗聴されていない情報を用いて鍵を生成することにより、攻撃者に鍵に関する情報を与えることなく送信者と受信者間で安全に鍵を共有できる。

[16] 同暗号の詳細については、後藤[2009]等を参照されたい。

た際の主なメリットとデメリットを図表 6 にまとめる。格子暗号については、特に暗号化状態処理技術を実現できるというメリットを有する点が学界で注目されており、近年、クラウドサービス等への応用を想定した研究や製品開発も活発化する等（図表 7)、同暗号の利用環境は整備されつつある[17]。

メリット	・量子コンピュータでも容易に解読できないと期待されている。 ・同暗号（特に LWE 方式）を利用して、データを暗号化したまま処理を行う技術（暗号化状態処理技術）を実現できる。
デメリット	・RSA 暗号や楕円曲線暗号と比較して、同等の安全性を確保するためには、現時点では数倍から数兆倍の鍵長が必要となる[18]。

図表 6. 現在主流の公開鍵暗号に対する格子暗号のメリットとデメリット

処理の名称	主な用途	処理性能
秘匿検索[19]	ビッグデータ分析	暗号化された 16,000 文字のデータの全数探索が約 1 秒で可能（富士通研究所[2014]）。
秘匿計算[20]		100 万件の暗号化されたデータの共分散や相関係数の計算が約 40 分で可能（安田・下山・小暮[2015]）。
		100 万件の暗号化されたデータの線形回帰係数の計算が約 40 分で可能（青野ほか [2015]）
	生体認証	生体情報の照合が約 5 ミリ秒で可能（富士通研究所[2013]）。

図表 7. 格子暗号の実用化動向

ロ. 格子暗号の概要

2 節(1)で述べたとおり、公開鍵暗号の安全性は、「公開鍵から秘密鍵を現実的な時間で求めるのが難しい」ことに基づいており、この仕組みを実現するために数学的問題を利用している。格子暗号が利用する数学的問題は、格子点探索問題と呼ばれ、格子（ベクトル空間上に規則正しく並んでいる点の集合）が与

[17] 格子暗号の実現方式の 1 つ「NTRU 方式」については、金融分野に関連する国際標準である IEEE1363.1（IEEE[2009]）や ANSI X9.98（ANSI[2010]）において規定されているが、これらの標準が規格化された後も、NTRU 方式に関する安全性評価の進展に伴い、同標準に記載されているパラメータ等の安全性は低下しうることに留意が必要である（理由は後述)。なお、現時点では同規格の利用事例は知られていない。

[18] ただし、この比較は、あくまで現在のコンピュータ環境に基づく評価手法を基準とした比較であり、量子コンピュータが実用化された環境下においては、RSA 暗号や楕円曲線暗号は安全性を確保できないことや、同環境を基準とした評価手法は定められていないことから、この比較は成立しなくなることに留意が必要である。

[19] 秘匿検索は、データを暗号化したままキーワード検索を行う処理である。

[20] 秘匿計算は、データを暗号化したまま統計解析等の数値計算を行う処理である。

えられたときに、ある条件を満たす格子点（格子上の 1 つの点）を探索する問題の総称である。探索条件の内容により複数の問題が存在し、前述した格子暗号の主な実現方式で利用される格子点探索問題としては、「最近ベクトル問題」、「最短ベクトル問題」、「Learning with Errors 問題」等が挙げられる（図表 8）。

格子暗号の 実現方式	利用している 格子点探索問題
AD 方式	近似版唯一最短ベクトル問題[21]
GGH 方式	最近ベクトル問題
NTRU 方式	最短ベクトル問題
LWE 方式	Learning with Errors 問題

図表 8. 格子暗号の主な実現方式が利用している格子点探索問題[22]

一般に、格子はその大きさや構造の複雑さ等を定める「次元n」と、格子を構成する格子点の基準を定める「基底$\mathbf{A}$」と呼ばれる 2 つのパラメータにより具体的な構造が定義される（図表 9）。ここで、基底 $\mathbf{A}$ は、格子の次元nと同じ本数の一次独立なベクトル[23]（図表 9 中の$\mathbf{a_1}, \mathbf{a_2}$等のこと。以下、「要素ベクトル」と呼ぶ）の組として定義され、これらの要素ベクトルを整数倍したうえで加算する（以下、この処理を「要素ベクトルを組み合わせる」と表現する）ことにより表現できる格子点全体の集合を格子と定義する（図表 10）。こうして定義された格子は、前述のとおりn次元ベクトル空間上に規則正しく並んだ格子点の集合となる[24]。ここで、基底を構成する各要素ベクトルの成分（n個の整数）を座標とみなして表現される点は格子点となることに留意されたい[25]。

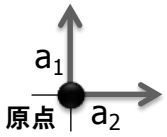
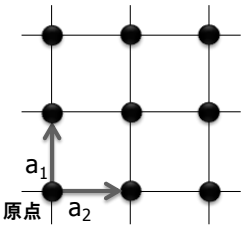
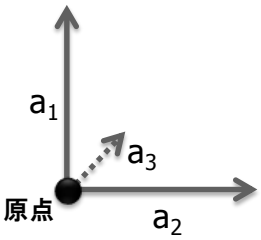
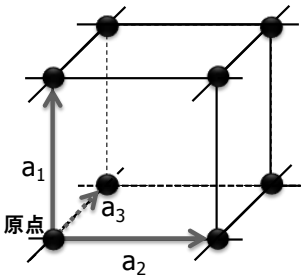
[21] 近似版唯一最短ベクトル問題は、特殊な最短ベクトル問題の 1 つであり、本来の最短ベクトル問題よりも、格子点の探索条件を少し弱めたものである。

[22] 各実現方式がどのような形で格子点探索問題を利用しているかについては、**3 節(3)**および **4 節**を参照。

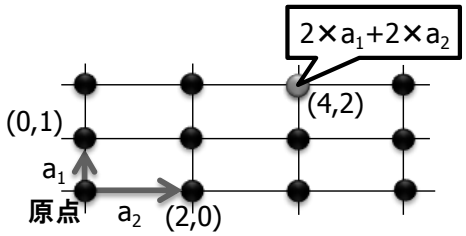
[23] 一般に、n本の要素ベクトルが 1 次独立であるとは、どの要素ベクトルも他の$n-1$本の要素ベクトルの実数倍の組合せで表現できないときをいう。

[24] 厳密には、格子の次元nに加えて、格子を定義するベクトル空間の次元mを選択する必要があり、格子暗号の具体的な実現方式においては、この 2 つのパラメータは必ずしも同じ値ではないが、説明をわかりやすくするために、本稿ではこの 2 つのパラメータは同じ値とする。

[25] 基底を構成する要素ベクトルを組み合わせる際、ある要素ベクトル以外は全て 0 倍したうえで加算すると、その要素ベクトルの成分を座標とした格子点となるため。

次元n	基底$\mathbf{A}$	左記パラメータにより 構成される格子
2 次元 （平面）	2 本の要素ベクトル$\mathbf{a_1},\mathbf{a_2}$の組 	要素ベクトル$\mathbf{a_1},\mathbf{a_2}$を組み合わせることにより表現可能な全ての格子点の集合。 
3 次元 （立体）	3 本の要素ベクトル$\mathbf{a_1},\mathbf{a_2},\mathbf{a_3}$の組 	要素ベクトル$\mathbf{a_1},\mathbf{a_2},\mathbf{a_3}$を組み合わせることにより表現可能な全ての格子点の集合。 

図表 9. 2 次元および 3 次元空間上の格子の例（イメージ）



基底$\mathbf{A}=(\mathbf{a_1},\mathbf{a_2})$の要素ベクトル$\mathbf{a_1},\mathbf{a_2}$をそれぞれ整数倍して加算することにより、格子上の点を表現可能。

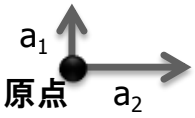
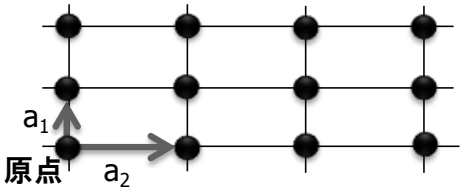
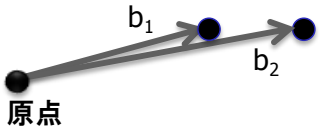
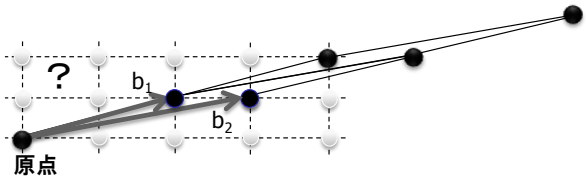
図表 10. 基底による格子点の表現の一例（イメージ）

格子暗号において利用される格子の重要な性質として、同じ格子（すなわち格子点の集合）を構成できる基底は「複数個」存在することが知られている（図表 11）。そして、基底はその特徴により、「直交型」と「非直交型」に分類できる[26]。それぞれの基底のイメージと特徴を図表 12 にまとめる。

[26] 学界では「直交型」と「非直交型」という表現は用いられず、基底において各要素ベクトルがある条件（任意の 2 本の基底が垂直に交わっている、厳密には「要素ベクトル同士の内積」と呼ばれる演算の結果が 0 となることを意味する）を満たすとき、この基底は直交していると

(a)基底$\mathbf{A} = (\mathbf{a_1}, \mathbf{a_2})$　　(b)基底$\mathbf{B} = (\mathbf{b_1}, \mathbf{b_2})$

図表 11. 同じ格子を構成する 2 つの基底**A**、**B**の例

基底の種類	イメージ	特徴
直交型		格子上の任意の格子点をマッピングしやすい。 ✓ 基底を構成する要素ベクトルの単純な組み合わせにより、原点周辺の格子点を並べていくことが容易。 
非直交型		格子上の任意の格子点をマッピングしにくい。 ✓ 基底を構成する要素ベクトルの単純な組み合わせでは、原点周辺の格子点を並べていくことが困難。これらの点を並べるためには基底を複雑に組み合わせる必要がある。 

図表 12. 直交型と非直交型の基底の例（2 次元空間の場合）

格子暗号が安全性の根拠とする格子点探索問題は、主に格子における基底（直交型と非直交型）の特徴に基づく、以下の性質を利用している。

性質：直交型の基底から非直交型の基底を演算することは容易だが、その逆演算を行うことは難しい。

呼ばれる。格子暗号で利用される基底は、厳密には直交している必要はなく、それに近い状態（ほぼ垂直に交わっている）ものでもよいが、本稿では、直感的に理解しやすくするため、真に直交した基底を用いることとする。

前述のとおり、直交型と非直交型の基底は、格子上の格子点のマッピングのしやすさが異なる。この性質は、直感的には、直交型の基底を知っている場合には格子点を構成する格子点の全体像（格子点の位置等）を把握しやすいため、「ある条件を満たす格子点を探索する問題（格子点探索問題)」を解くのが容易である。それに対し、非直交型の基底のみ知っている場合には、格子点の全体像を把握しにくいため、「ある条件を満たす格子点を探索する問題（格子点探索問題)」を解くのは難しいことを意味する。また、直交型の基底から非直交型の基底は、容易に計算できる[27]。一方、非直交型の基底から直交型の基底を求めるためには、「直交している」という条件を満たす要素ベクトル（格子点）を探索する（言い換えると、格子点探索問題を解く）必要があるが、非直交型の基底は格子上の任意の格子点をマッピングしにくいため、上記条件を満たす格子点の探索は難しい。

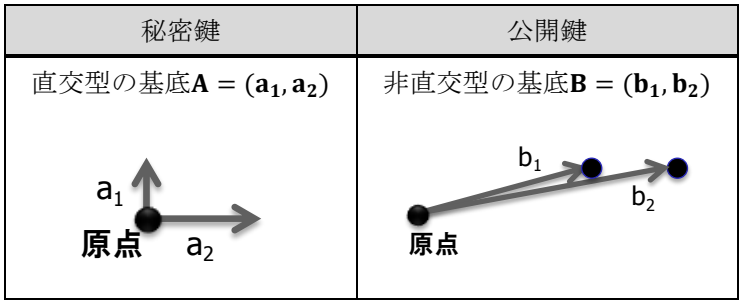
このような格子の特徴および格子点探索問題の困難性を利用して、「公開鍵から秘密鍵を現実的な時間で求めるのが難しい」という演算の一方向性の仕組みを実現し、安全性要件を満たす公開鍵暗号を構成したものが格子暗号である。

ハ．格子暗号の処理フロー

格子暗号において、送信者が公開鍵を用いて平文を暗号化して暗号文を生成し、それを受信した受信者が秘密鍵により暗号文から平文を復号する処理フローについて紹介する。ここでは、格子暗号の実現方式の 1 つであり、格子上での暗号化・復号処理を視覚的に説明しやすい「GGH 方式」に基づく直感的な処理フローを示す。同暗号における、「秘密鍵、公開鍵の設定」、「暗号化」、「復号」の処理手順を以下に示す。

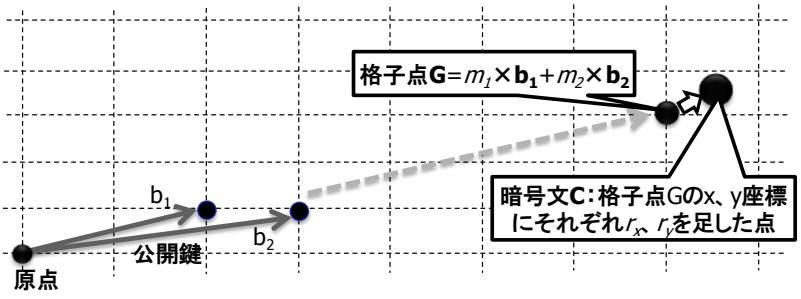
秘密鍵、公開鍵の設定手順 : 受信者は、自身の秘密鍵と公開鍵を設定する前に、次元nを定めた後、全利用者に公開する（このような、鍵設定を行う前に予め全ての利用者間で共有するパラメータを、以下「共通パラメータ」と呼ぶ。ここでは、直感的に説明するため、$n = 2$の場合の例を示す)。なお、格子暗号における鍵長は、この次元の大きさに依存する。その後、このパラメータに基づき、直交型の基底$\mathbf{A} = (\mathbf{a_1}, \mathbf{a_2})$（$\mathbf{a_1}$,$\mathbf{a_2}$はそれぞれ2次元ベクトル）を選択し、利用する具体的な格子を定める。そして、同じ格子を構成できる非直交型の基底$\mathbf{B} = (\mathbf{b_1}, \mathbf{b_2})$（$\mathbf{b_1}$,$\mathbf{b_2}$はそれぞれ2次元ベクトル）を選択する。そして、基底$\mathbf{A}$を秘密鍵とし受信者自身が秘密に保持し、基底$\mathbf{B}$を公開鍵として全利用者に公開する（図表 13)。

[27] 例えば、「ユニモジュラ行列」と呼ばれる特殊な行列を用いた線形結合により、効率良く計算する方法等がある。



図表 13. 格子暗号の秘密鍵と公開鍵

暗号化処理の手順：送信者は、受信者にデータ（平文）を暗号化して送るとき、はじめに、送りたい平文を 2 つの整数（m_1, m_2）として表現する。次に、この平文（m_1, m_2）と公開鍵$\mathbf{B}=(\mathbf{b_1},\mathbf{b_2})$を利用して、格子点$\mathbf{G}=m_1\times\mathbf{b_1}+m_2\times\mathbf{b_2}$を計算する（言い換えると、基底$\mathbf{B}$を平文（$m_1, m_2$）に基づき組み合わせて格子点$\mathbf{G}$を計算する）。さらに、2 つの小さな実数（$r_x, r_y$）をランダムに選択した後、格子点$\mathbf{G}$に（$r_x, r_y$）を加えた点を暗号文$\mathbf{C}$とする。具体的には、格子点$\mathbf{G}$の座標を (G_x, G_y)としたとき、暗号文$\mathbf{C}=(G_x+r_x, G_y+r_y)$を計算する（図表 14）。ここで、（$r_x, r_y$）は、$\mathbf{C}$に最も近い[28]格子点が$\mathbf{G}$となるような値を選ぶこととする[29]。その後、暗号文$\mathbf{C}$を受信者に送る。



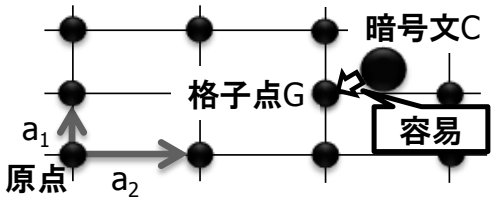
図表 14. 暗号化処理の手順（イメージ）

復号処理の手順：受信者は、はじめに、秘密鍵$\mathbf{A}=(\mathbf{a_1},\mathbf{a_2})$を用いることにより、暗号文$\mathbf{C}$から格子点$\mathbf{G}$を求める。ここで、$\mathbf{A}=(\mathbf{a_1},\mathbf{a_2})$は直交型の基底であり、格子の全体像（各格子点の位置等）を把握しやすいという性質を有

[28] 格子暗号では、一般に、点同士の距離を測る尺度としては「ユークリッド距離」が用いられる。そして、暗号文$\mathbf{C}$の最も近い格子点が$\mathbf{G}$となるとは、暗号文$\mathbf{C}$と格子点$\mathbf{G}$のユークリッド距離が、他の格子点とのユークリッド距離と比較して、最も短いことを意味する。

[29] 厳密には、暗号文$\mathbf{C}$と格子点$\mathbf{G}$の距離が近すぎる場合、公開鍵$\mathbf{B}$のみで解読できてしまうため、適切な距離を保つように（r_x,r_y）を選択する必要がある（距離の適切な選択方法については Klein[2000]、Dadush, Regev, and Stephens-Davidowitz[2014]を参照）。

するため、秘密鍵$\mathbf{A}$を用いて「暗号文$\mathbf{C}$に最も近い格子点$\mathbf{G}$」を容易に求めることができる（図表 15）。格子点$\mathbf{G}$を求めた後、公開鍵$\mathbf{B} = (\mathbf{b_1}, \mathbf{b_2})$を用いて、連立方程式を解くことにより平文$(m_1, m_2)$を復号する（図表 16）。



秘密鍵である直交型の基底$\mathbf{A} = (\mathbf{a_1}, \mathbf{a_2})$を利用すると、暗号文$\mathbf{C}$周辺に存在する格子点の位置を把握できるため、その中から$\mathbf{C}$に最も近い格子点$\mathbf{G}$を容易に特定できる。

図表 15. 暗号文$\mathbf{C}$からの格子点$\mathbf{G}$の特定

手順 1 : 格子点$\mathbf{G}$の座標を(G_x, G_y)としたとき、暗号化処理に基づき、公開鍵$\mathbf{B} = (\mathbf{b_1}, \mathbf{b_2})$と平文$(m_1, m_2)$を組み合わせて$\mathbf{G}$を以下の式で表現する。 $\mathbf{G} = (G_x, G_y)$ $= m_1 \times \mathbf{b_1} + m_2 \times \mathbf{b_2}$ $= m_1 \times (b_{1x}, b_{1y}) + m_2 \times (b_{2x}, b_{2y}).$	手順 2 : 左の式から導出される以下の 2 元 1 次連立方程式を解き、平文(m_1, m_2)を求める。 ✓ 2 個の未知数(m_1, m_2)に対して、2 本の方程式が得られるため、「ガウスの消去法」等の手法を用いて、容易に解くことができる。 $\begin{cases} G_x = m_1 \times b_{1x} + m_2 \times b_{2x}, \\ G_y = m_1 \times b_{1y} + m_2 \times b_{2y}. \end{cases}$

図表 16. 格子点$\mathbf{G}$と公開鍵$\mathbf{B}$から平文$(\boldsymbol{m_1}, \boldsymbol{m_2})$を求める手順

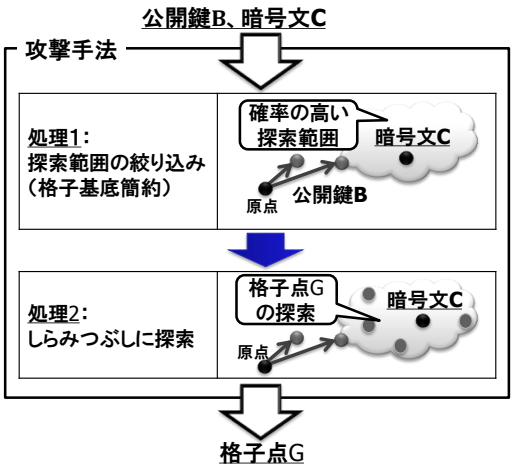
ここで、上記の格子暗号の安全性について説明する。同暗号の安全性は、主に格子の数学的特徴および格子点探索問題の困難性に起因する以下の根拠に基づき確保されている。

安全性の根拠：この格子暗号では、攻撃者は公開鍵$\mathbf{B}$から秘密鍵$\mathbf{A}$を求めることができれば、図表 16 と同様の手順により、暗号文$\mathbf{C}$から平文(m_1, m_2)を復号することが可能となる。また、攻撃者が公開鍵$\mathbf{B}$と暗号文$\mathbf{C}$から格子点$\mathbf{G}$を求めることができれば、つまり、格子点探索問題（最近ベクトル問題）を解くことができれば、秘密鍵$\mathbf{A}$を求めることなく、図表 16 の手順により平文(m_1, m_2)を復号できる。したがって、格子点探索問題を現実的な時間で解くのが困難であれば、本節で紹介した格子暗号は、公開鍵から秘密鍵を求めるのが困難という公開鍵暗号の安全性要件を満たす。

ただし、現時点では、上記の格子暗号のベースとなっている GGH 方式の安全性と格子点探索問題（最近ベクトル問題）の難しさが同程度であるか知られていない。そのため、たとえ格子点探索問題が難しいとしても、方式特有の脆弱性により、今後、安全性が低下する可能性があることに留意されたい（詳細は図表 18 参照）。

二．格子暗号に対する攻撃手法

格子暗号の安全性の根拠となっている格子点探索問題を解く一般的な攻撃手法は、「処理 1：探索範囲の絞り込み（格子基底簡約）」と「処理 2：しらみつぶしに探索」の処理から構成される（図表 17）。ここでは、例として、**3 節(2)ハ**で紹介した格子暗号に同攻撃手法を適用した場合を想定したうえで、各処理の概要を示す。



図表 17．格子点探索問題を解く一般的な攻撃手法のメカニズム（イメージ）

処理 1：探索範囲の絞りこみは、格子上で暗号文**C**に最も近い格子点**G**の候補が存在する確率の高い範囲をある程度絞り込むための処理である[30]。同処理を行う代表的な手法として、「LLL 手法（Lenstra, Lenstra, and Lovász[1982]）」、「BKZ 手法（Schnorr[1987]）」、「BKZ2.0 手法（Chen and Nguyen[2011]）」等が挙げられる。

[30] 厳密には、基底の中には非直交型の部分と直交型の部分が混在しており、格子点**G**の探索の際には、非直交型の部分をしらみつぶしに探索すれば、残りの直交型の部分の探索は比較的高速に処理できることが知られている。よって、基底**B**を直交型に近いものに変換することで、同基底の中の「非直交型」が占める割合が減るため、しらみつぶし探索を行うべき範囲を減らすことができる。

処理 2 : しらみつぶしの探索は、処理 1 で絞り込んだ探索範囲内から「暗号文$\mathbf{C}$に最も近い」という条件を満たす格子点$\mathbf{G}$を文字どおりしらみつぶしに探索する処理である。基本的な手法としては「Kannan 手法（Kannan[1983]）」があるほか、攻撃の成功確率を下げる代わりに処理に要する手間を削減する手法[31]として、「ランダムサンプリング手法（Schnorr[2003]）」や「Extreme-Pruning 手法（Gama, Nguyen, and Regev[2010]）」が知られている。

一般に、上記の処理を組み合わせた攻撃手法は次元nが大きくなるにつれ、現実的な時間で格子点$\mathbf{G}$を探索するのが困難になることが知られており、また量子コンピュータを利用したとしても、現時点では効率的な攻撃手法が見つかっていない。したがって、パラメータである次元nを適切な大きさで選択することにより、たとえ量子コンピュータでも、格子点探索問題を現実的な時間で解くことは困難であると考えられる。このことが、格子暗号が耐量子コンピュータ暗号に分類されている根拠となっている。

(3) 格子暗号の実現方式

前述のとおり、格子暗号は格子点探索問題を安全性の根拠とする公開鍵暗号の総称であり、これまでに提案されている主な実現方式は「AD 方式」、「GGH 方式」、「NTRU 方式」、「LWE 方式」の 4 種類である。これらの実現方式の概要及び研究動向について図表 18 にまとめる（LWE 方式は除く）。なお、LWE 方式については、近年研究が活発化しており、格子暗号の主流としての有力な候補となっているため、次節においてやや詳しく取り上げる。

[31] 格子暗号に限らず一般的な暗号アルゴリズムの場合、解読に成功する確率がたとえ 100%でなくても、確率がある程度高い場合には潜在的な脅威となり得るため、このような処理に要する手間（計算量）を削減する手法の研究は重要となる。

実現方式の名称	概要	安全性	研究動向
AD 方式	1997 年にアィタイとドワークにより提案。1 ビットの平文の暗号化のみ可能であり、暗号化処理は格子上のベクトル演算を利用。	AD 方式は、格子点探索問題（近似版唯一最短ベクトル問題）の困難性を安全性の根拠としており、同方式の安全性とこの格子点探索問題の難しさが同程度であることが数学的に証明されている。したがって、格子点探索問題が難しければ、AD 方式特有の要因で安全性が低下する可能性は低い。	グエンらによる厳密な安全性評価の結果（Nguyen and Stern[1998]、Nguyen[1999]）、これらの方式を安全に利用するために必要な鍵長は長いことが指摘されたほか、大幅な効率化も難しいと考えられているため、これらの方式に関する研究については大きな進展なし。
GGH 方式	1997 年にゴールドライヒ、ゴールドワッサー、ハレビにより提案。暗号化処理は格子上のベクトル演算を利用。	GGH 方式、NTRU 方式はともに格子点探索問題（最近ベクトル問題、最短ベクトル問題）を安全性の根拠としている。しかし、両方式の安全性と格子点探索問題の難しさが同程度であるかについて、現時点では知られていない。したがって、たとえ格子点探索問題が難しいとしても、各方式特有の脆弱性により、今後、安全性が低下する可能性がある。	
NTRU 方式	1997 年にホフスタイン、パイファー、シルバーマンにより提案。暗号化処理は、上記の 2 方式とは異なり、多項式同士の演算を利用。		NTRU 方式で用いている格子は、特殊な構造を有しているため、同構造に特有の脆弱性を利用した攻撃手法の提案が行われている（Coppersmith and Shamir[1997]）。

図表 18. 格子暗号の主な実現方式（LWE 方式は除く）

4. LWE 方式

LWE 方式は、2005 年にレゲフにより提案された格子暗号の実現方式の 1 つであり（Regev[2009]）、他の実現方式と比較したとき、安全性と効率性のバランスの観点から現時点では最も優れていると考えられており、学界で注目されている。特に、LWE 方式は、暗号化状態処理技術の 1 つである「秘匿計算」について、他の実現方式に比べて複雑な計算（統計解析等）を実現することも可能であることから、近年、クラウドサービス等への応用を想定した研究や製品開発等が活発化しており、格子暗号の主流として有力な候補と考えられている。

LWE 方式は、格子点探索問題（具体的には、Learning with Errors 問題）を安全性の根拠としている。しかしながら、同方式においては、利用する格子があ

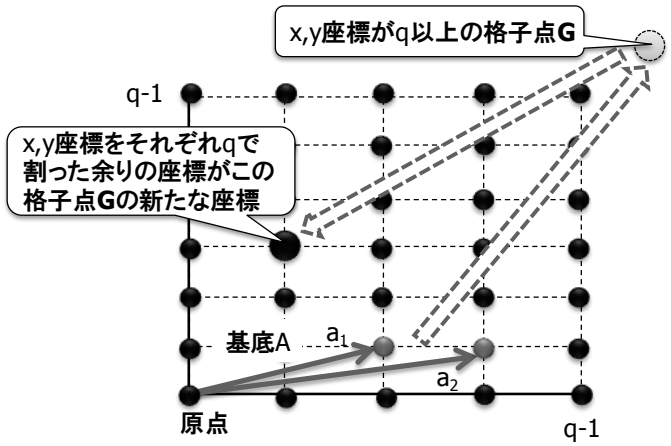
る条件を満たす場合には、たとえ大きな次元を選択しても、効率良く解読できる手法が存在する（Laine and Lauter[2015]等）。したがって、LWE 方式で利用する格子をどのように選択するかは、同方式の安全性に大きく影響を与える事項であるため、パラメータの選択には注意が必要である。そこで、本節では、LWE 方式の概要について説明した後、利用する際の安全なパラメータの選び方について整理する。

(1) LWE 方式とパラメータ

LWE 方式を実装しようとした場合、3 節(2)ハで紹介した格子暗号における鍵の設定と同様、受信者は自身の秘密鍵と公開鍵の生成を行う前に、具体的にどのパラメータで規定される格子を利用するか等を決める必要がある。なお、3 節(2)ハで紹介した格子暗号（GGH 方式）と LWE 方式では、利用する格子の種類が異なる。具体的には GGH 方式で利用した格子は、各格子点の座標が整数全体で表現される一般的な格子を用いて説明したが、LWE 方式においては、各格子点の座標としてとり得る整数の範囲があるパラメータ（「素数」）により制限されている特殊な格子（このような格子は「有限体上の格子」と呼ばれる、図表 19）を利用する。

LWE 方式での秘密鍵と公開鍵の設定において、受信者は初めに利用する格子を定める。具体的には、「次元n」に加えて、格子点の座標の範囲を定める「素数q」を選び、これらのパラメータに基づき「非直交型の基底**A**」を指定することにより具体的に利用する格子（有限体上の格子）を決定する[32]。次に、秘密鍵の選択に関連するパラメータである「実数α」を選択し、これら（次元n、素数q、基底**A**、実数α）を共通パラメータとして設定した後、全利用者に公開する。その後、受信者は、設定した同パラメータに基づき、自身の秘密鍵と公開鍵を生成した後、秘密鍵を自身で秘密に保持し、公開鍵を全ての利用者に公開したうえで、データの暗号化や復号の処理に利用する（図表 20、具体的な鍵の生成手順および暗号化等の処理フローについては Box 1、2 参照）。

[32] 厳密には、LWE 方式においては、安全性を確保するために、格子の次元nに加えて、格子を定義するベクトル空間の次元mをnよりも大きな値で選択する必要がある。しかし、本稿では、説明をわかりやすくするために、3 節(2)と同様、nとmは同じ値とする。

性質	イメージ
格子点の座標（x,y 座標）としてとり得る値の範囲が「0 から$q-1$」の範囲に制限されている。 ✓ 基底の要素ベクトルを組み合わせて表現した格子点**G**の座標（x,y 座標）がq以上となった場合、「座標をqで割った余り（すなわち、$q-1$以下の座標値）」を、格子点**G**の新たな座標とする。	

図表 19. LWE 方式で利用する有限体上の格子（イメージ）

共通パラメータ	秘密鍵	公開鍵
• 格子の次元n • 素数q • 非直交型の基底**A** • 実数α（秘密鍵**e**として選択する値の大きさを定める）	• n個の整数の組 $\mathbf{s}=(s_1,s_2\ldots,s_n)$ • n個の整数の組 $\mathbf{e}=(e_1,e_2,\ldots,e_n)$	• 点 **T**（**A**の各ベクトルを**s**に基づき組み合わせて表現される格子点**G**に**e**を加えた点[33]）。

図表 20. LWE 方式における共通パラメータ、秘密鍵と公開鍵

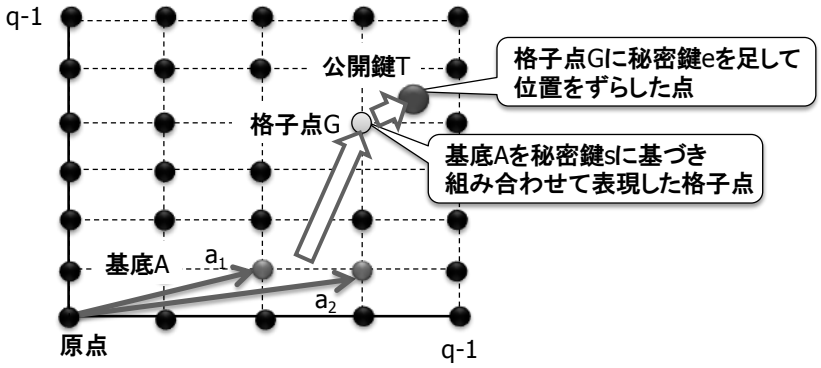
図表 20 のとおり、LWE 方式では、「n個の整数の組**s**,**e**」を秘密鍵とし[34]、共通パラメータの 1 つである基底**A**を**s**に基づき組み合わせて表現した格子点**G**（$\mathbf{G}=\mathbf{A}\times\mathbf{s}$）に**e**を加えた点**T**（$\mathbf{T}=\mathbf{G}+\mathbf{e}$）を公開鍵として利用する（図表 21）。そして、同方式の安全性と格子点探索問題の 1 つである「Learning with Errors 問題[35]」の難しさが同程度であることが数学的に証明されている。具体的には、攻撃者は公開鍵である点**T**から格子点**G**を計算することができれば、2 点の座標の差分を取ることにより秘密鍵**e**を容易に計算できるほか、共通パラメータである基底**A**を用いて、図表 16 と同様の手法で導出した連立方程式を解くことにより、容易に秘密鍵**s**を求めることができる。しかし、点**T**から格子点**G**を計算するため

[33] ここで、点**T**は格子点ではないことに注意。

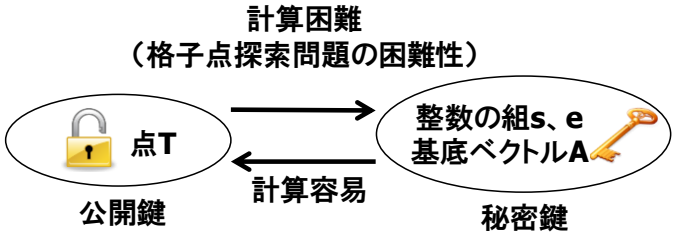
[34] 整数の組**e**については、公開鍵の生成時にのみ必要なパラメータであり復号処理に利用しないため、必ずしも秘密鍵として保管する必要はないが、本稿では LWE 方式の安全性を理解しやすくするために、秘密鍵の 1 つとして取り扱うこととする。

[35] 厳密には、LWE 問題は「特殊な連立 1 次方程式を解く問題」だが、同問題と格子点探索問題の間には関係（一方の問題が解けるのであれば、もう一方の問題が解けるという関係、「帰着関係」と呼ばれる）があることが知られていることから、LWE 方式は格子暗号に分類される。

には、格子点探索問題を解く必要があり、点**G**を計算するために攻撃者が利用可能な基底**A**は非直交型であるため、一般に点**G**を計算することは難しい。したがって、格子点探索問題を現実的な時間で解くのが困難であれば、LWE 方式は公開鍵暗号の安全性要件を満たす（図表 22）。



図表 21. LWE 方式における秘密鍵と公開鍵の関係



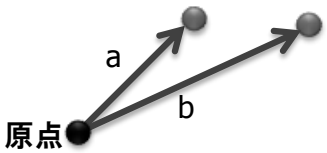
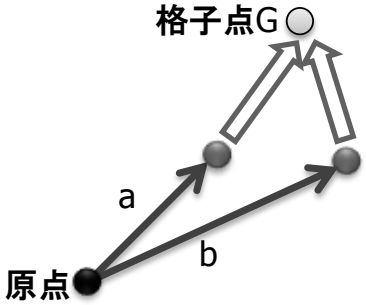
図表 22. LWE 方式の安全性と格子点探索問題の関係

Box 1 格子と行列の関係

上記の説明においては、LWE 方式の鍵生成および安全性の根拠について格子を用いて説明したが、格子を直接ソフトウェアで記述するのは難しい。そのため、一般に、LWE 方式を実装する際には、格子の構造をソフトウェアで取り扱いやすくするため、「行列」と呼ばれる表現形式を利用する（図表 A-1）。具体的には、格子の基底や格子点等を行列の形に変換したうえで、行列同士の演算を行うことにより、暗号化および復号処理が行われる（図表 A-2、行列の演算規則については**補論**参照）。

$n \times 1$行列	$1 \times n$行列	$n \times n$行列
数値（「行列の成分」と呼ばれる）を縦にn個並べた形で表現される行列。 $\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$	数値を横にn個並べた形で表現される行列。 $[z_1 \quad z_2 \quad \cdots \quad z_n]$	数値を縦、横にそれぞれn個並べた形で表現される行列。例えば、$n \times 1$行列を横にn個並べて同行列を構成可能。 $\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nn} \end{bmatrix}$

図表 A-1　行列の代表的な種類

基底		$\mathbf{a} = \begin{bmatrix} a_x \\ a_y \end{bmatrix}$、$\mathbf{b} = \begin{bmatrix} b_x \\ b_y \end{bmatrix}$ 各要素ベクトルは座標を縦に並べた行列（2×1行列）で表現される。
格子点		$\mathbf{G} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = s_1 \begin{bmatrix} a_x \\ a_y \end{bmatrix} + s_2 \begin{bmatrix} b_x \\ b_y \end{bmatrix}$ 格子点は、行列で表現された基底の各要素ベクトル**a**,**b**を整数倍して足し合わせた行列（2×1行列）で表現される（ここでs_1,s_2は整数）。

図表 A-2　格子と行列の対応関係（$n = 2$の場合）

Box 2　LWE 方式における鍵の設定手順と具体的なアルゴリズムの例

＜共通パラメータや秘密鍵、公開鍵の設定手順＞

【共通パラメータの設定手順】

Step 1. はじめに、受信者は利用する格子の次元（正整数）nと素数qを選ぶ。次に、n本の$n \times 1$行列$\mathbf{a_1}, \mathbf{a_2}, \dots, \mathbf{a_n}$を選択し、要素ベクトルとしたうえで、これらの組を$n \times n$行列**A**として表現し、これを基底とすることにより、利用する具体的な格子を決定する。ここで、行列の各成分（行列中の各数値）は、全て$q - 1$以下の正整数とし、基底**A**は非直交型であるとする。さらに、実数αを選択する（各パラメータの選択条件については**4 節(2) 参照**）。

Step 2. 格子点の次元n、素数q、非直交型の基底$\mathbf{A}$、実数αを共通パラメータとして、全ての利用者に公開する。

【秘密鍵、公開鍵の設定手順】

Step 1. 受信者は、はじめに$n \times 1$行列$\mathbf{s}$（各成分は$q-1$以下の正整数）をランダムに選ぶ。

Step 2. 次に、基底$\mathbf{A}$の各要素ベクトルを秘密鍵$\mathbf{s}$に基づき組み合わせることで、格子点$\mathbf{G}$を求める。具体的には、$n \times n$行列で表現される基底$\mathbf{A}$と$n \times 1$行列$\mathbf{s}$を用いて乗算$\mathbf{G} = \mathbf{A} \cdot \mathbf{s} \bmod q$を行い[36]、計算結果である$n \times 1$行列$\mathbf{G}$を格子点とする。

Step 3. さらに、$n \times 1$行列$\mathbf{e}$（各成分は、実数αにより定まる確率分布Ψ_αに基づき選択された整数[37]）を選択した後、$n \times 1$行列で表現される格子点$\mathbf{G}$に$n \times 1$行列$\mathbf{e}$を加えた$n \times 1$行列$\mathbf{T} = \mathbf{G} + \mathbf{e} \bmod q$を計算する。そして、($\mathbf{s}$,$\mathbf{e}$)を秘密鍵とし、$n \times 1$行列$\mathbf{T}$を公開鍵とする。

＜LWE 方式の具体的な暗号アルゴリズムの例＞

以下では、代表的な LWE 方式（Regev[2009]や Peikert[2008]）に基づく暗号化方式の暗号化及び復号の手順を示す。ここでは、最も単純な 1 ビットの平文を暗号化する方式を示す（複数ビットを暗号化する方法については、Peikert[2008]等を参照)。なお、以下で紹介する方式においては、素数qは 3 以上とする。

【暗号化処理の手順】

Step 1. 送信者は、はじめに$1 \times n$行列$\mathbf{r}$（各成分は、実数αにより定まる確率分布Ψ_αに基づき選択された整数）を選択したのち、$n \times n$行列で表現される基底$\mathbf{A}$と$1 \times n$行列$\mathbf{r}$を用いて$\mathbf{C_1} = \mathbf{r} \cdot \mathbf{A} \bmod q$を計算する。

Step 2. 次に、平文が「1」の場合には$\mathrm{M} = (q+1)/2$、「0」の場合には$\mathrm{M} = 0$と設定したうえで、$n \times 1$行列で表現される公開鍵Tと$1 \times n$行列$\mathbf{r}$を用いて、$\mathbf{C_2} = \mathbf{r} \cdot \mathbf{T} - \mathrm{M} \bmod q$を計算する。

Step 3. 得られた（$\mathbf{C_1}$,$\mathbf{C_2}$）を暗号文とする。

[36] 「mod q」は行列同士の演算（**補論**参照）を行い得られた行列の各成分をそれぞれ「素数qで割った余り」に置き換えることを意味する。

[37] この確率分布Ψ_αは、整数全体の集合からある 1 つの整数を選択する際に、その整数がどのような確率で選択されるかというルールを定めるものである。LWE 方式においては、0 に近い整数が高い確率で選択される確率分布（「離散ガウス分布」）」を利用しており、共通パラメータの 1 つである実数αは、各整数が選択される具体的な確率を定める役割を担っている。

【復号処理の手順】

Step 1. 暗号文（$\mathbf{C_1}$,$\mathbf{C_2}$）と秘密鍵$\mathbf{s}$を用いて、$\mathbf{C_1}\cdot\mathbf{s}-\mathbf{C_2}=-\mathbf{r}\cdot\mathbf{e}+\mathrm{M} \bmod q$を計算する。

Step 2. ここで、$-\mathbf{r}\cdot\mathbf{e}$は、高確率で「$-(q+1)/4<-\mathbf{r}\cdot\mathbf{e}<(q+1)/4$」となることが知られている。そこで、受信者は「$-(q+1)/2<-\mathbf{r}\cdot\mathbf{e}+\mathrm{M}<-(q+1)/4$または$(q+1)/4<-\mathbf{r}\cdot\mathbf{e}+\mathrm{M}<(q+1)/2$」の場合には「1」、「$-(q+1)/4<-\mathbf{r}\cdot\mathbf{e}+\mathrm{M}<(q+1)/4$」の場合には「0」を平文とする。

前述のとおり、LWE 方式は、他の実現方式と比較したとき、安全性と効率性のバランスの観点から現時点では最も優れていると考えられている。その根拠としては、（１）AD 方式と同様に、LWE 方式の安全性と格子点探索問題の 1 つである Learning with Errors 問題の難しさが同程度であることが数学的に証明されているため、今後、実現方式特有の要因で安全性が低下する可能性は低いこと、（２）近似版唯一最短ベクトル問題（AD 方式が安全性の根拠とする格子点探索問題）よりも Learning with Errors 問題（LWE 方式が安全性の根拠とする格子点探索問題）の方が相対的に解くのが難しいため、LWE 方式は AD 方式と比較すると、より短い鍵長で同程度の安全性を確保できることが挙げられる。

(2) LWE 方式における安全なパラメータの選択

LWE 方式の安全性の根拠となっている Learning with Errors 問題を解く攻撃手法のうち、3 節(3)二で紹介した汎用的な手法よりも効率の良いものは現時点では知られていない[38]。したがって、基本的には共通パラメータの 1 つである次元nを大きくとれば、Learning with Errors 問題を現実的な時間で解くのは難しいと考えられている。しかし、LWE 方式については、たとえ次元nとして大きな値を選択しても、それ以外のパラメータである素数q、実数αや秘密鍵の 1 つ$\mathbf{e}$の選択によっては、3 節(3)二で紹介した汎用的な手法（または、その改良手法等）が Learning with Errors 問題を効率良く解読する可能性がある。そこで、これまでの研究成果に基づいて、LWE 方式を安全に利用するための各パラメータの選択条件等を図表 23 にまとめる。ただし、LWE 方式を含む格子暗号の安全性評価については未だ発展途上であり、現時点では理論的な評価基準が定まっているとは言い難い状況である。一方、実際に攻撃手法を実装し、攻撃に要する時間を計測する等の実験による安全性評価に基づく、LWE 方式の安全な共通パラメータ

[38] 汎用的な攻撃手法について、適用対象を LWE 方式（Learning with Errors 問題）に特化することにより効率を向上させた方式として、「Decoding 手法（Linder and Peikert[2011], Bai and Galbreith[2014]）」や「Distinguishing 手法(Micciancio and Regev[2007])」が提案されている。しかし、これらの手法も次元nとして大きな値を選択することにより、現実的な時間で LWE 方式を解読することは難しい。

の候補については、Linder and Peikert[2011]、Aono *et al*.[2013]、Liu and Nguyen[2013]、青野ほか[2015]等の参考情報が存在する（図表 24）。したがって、LWE 方式の利用に際しては、これらの文献を参照するとともに、安全性評価に関する最新の研究動向をフォローすることが重要となる。

パラメータの種類	安全な選択条件	理由	主な参考文献
次元n	3 桁以上の正整数	nの値が小さいと、格子点探索問題は容易に解読可能。	Regev[2009], Micciancio-Regev[2009]等
素数q	4 桁程度の素数	LWE 方式が利用する有限体上の格子は、素数qの値が大きくなるほど、解候補となる格子点の数が減るという特徴を有しており、しらみつぶし探索が容易に可能となるため。ただし、秘匿計算などの暗号化状態処理の利用においては、qが数十桁以上になる方式も存在する。これらの方式に対しては次元nを 4 桁以上にとることで安全性を確保している。	Micciancio and Regev[2009]、青野ほか[2015]、Laine and Lauter [2015]
実数α	1.5 よりも大きな実数	実数αの値を小さくとると、それに伴い秘密鍵の 1 つである**e**の各成分が小さな値になってしまい、結果として公開鍵**T**に近い格子点**G**が容易に探索できるため。	青野ほか[2015]
秘密鍵 **e**	秘密鍵**e**の各成分を 0 または 1 のいずれかのみから選択しない。	秘密鍵**e**の各成分を 0 または 1 のいずれかのみから選択した場合、実数αのときと同様に公開鍵**T**に近い格子点**G**が容易に探索できるため。	Albrecht *et al*.[2014]

図表 23. LWE 方式におけるパラメータの選択条件とその理由

安全性評価指標[39]（ビット）	次元n	素数	実数 α
128	230	2,053	8.0
256	250	4,093	

図表 24. LWE 方式の安全なパラメータの例[40]

備考：Linder and Peikert[2011]、Aono *et al.*[2013]、Liu and Nguyen[2013]、青野ほか[2015]に基づく。

(3) LWE 方式の研究動向

学界においては、近年、鍵長を数千分の 1 程度まで削減できる LWE 方式の改良方式（「ring-LWE 方式」）が提案され（Lyubashevsky, Peikert and Regev[2010]）、実現方式や実用化等について活発に研究が行われている。ただし、ring-LWE 方式と LWE 方式の安全性が同程度であるかについて、現時点では明らかにされていないため、同方式の利用に際しては留意する必要があるとともに、今後の安全性評価の研究動向に注目する必要がある。

また、LWE 方式の安全性の根拠となっている格子点探索問題（Learning with Errors 問題）については、前述の通り効率的な攻撃手法が見つかっていないものの、LWE 方式の暗号アルゴリズムの実装上の不備を悪用した攻撃については、今後起こり得る可能性がある。そのような攻撃手法の 1 つに「サイドチャネル攻撃[41]」がある。サイドチャネル攻撃自体は、LWE 方式や格子暗号に限った攻撃手法ではなく、暗号アルゴリズム全般について適用可能なものであり、現在主流で利用されている RSA 暗号や楕円曲線暗号については、サイドチャネル攻撃に関する研究が古くから行われている[42]。LWE 方式の格子暗号については、まだその原理が提案されて年月が経っていないこともあり、現時点では、サイドチャネル攻撃に関する研究は多く見られないものの、実装の方法によってはサイドチャネル攻撃にさらされる可能性を指摘している研究者もいる（Bos *et al.*[2015]）。このため、今後も攻撃手法や安全性評価に関する研究動向に注目し

[39] 安全性評価指標（「ビット安全性」）は、公開鍵暗号等の暗号アルゴリズムの安全性を評価する際の指標の 1 つである。同指標では、ある暗号アルゴリズムについて、その安全性を破るために必要な計算量が2^k回の演算処理に相当する場合、その暗号アルゴリズムは「kビット安全」という。

[40] 前述のとおり、実際に LWE 方式を実装する際には、これらのパラメータに加えて、格子を定義するベクトル空間の次元mを選択する必要があるが、同パラメータについては「$m = n \times \log_2 q$」という値を選択することが推奨されている（Regev[2009]）。

[41] サイドチャネル攻撃は、実際に暗号処理を行っている PC や組込み機器等の消費電力や処理時間の変化等を計測し、その計測結果から秘密鍵を推定する攻撃手法である。

[42] 同攻撃手法の詳細については、例えば鈴木・菅原・鈴木[2015]を参照されたい。

ていく必要がある。

さらに、LWE 方式を利用して、データを暗号化したまま複雑な計算を可能にする技術（秘匿計算）も提案されており（Brakerski, Gentry and Vaikuntanathan[2014]等[43]）、同技術の効率性向上や実用化に関する研究も盛んに行われている。さらに、最近、秘匿計算の仕組みを応用し、データを暗号化したまま安全性のレベルを更新可能な技術が提案されている（青野ほか[2015]）。このように、LWE 方式については、量子コンピュータによる解読に耐えうるという性質とともに、秘匿計算等の高度な機能を有する暗号アルゴリズムの実現に利用可能という性質が注目され、今後、学界を中心にその安全性評価や実用化に関する研究が活発化し、利用環境が整備されることが期待される。

5. おわりに

本稿では、量子コンピュータ（量子デジタル型）の実現により、現在主流の公開鍵暗号である RSA 暗号や楕円曲線暗号が容易に解読され得る状況について紹介したうえで、同コンピュータでも容易に解読できないと期待される格子暗号の概要および研究動向等について解説した。格子暗号は、格子上の基底の性質（直交型、非直交型）と格子点探索問題の困難性を利用して実現された公開鍵暗号であり、格子点探索問題は、現時点では量子コンピュータでも現実的な時間で解くことができないと考えられている。格子暗号は、データを暗号化したまま処理を行う暗号化状態処理技術を実現できる特長も有していることから、学界等で注目されている。特に、格子暗号の実現方式の 1 つである LWE 方式は、他の実現方式と比較したとき、安全性と実用性のバランスの観点から現時点では最も優れていると考えられており、近年、同方式に関する研究が活発化している。一方、格子暗号は現在主流の RSA 暗号や楕円曲線暗号と比較して、同じ安全性を確保するために必要な鍵長が長いというデメリットを有している（図表 25）。また、同暗号の安全性評価の研究は未だ発展途上であり、理論的な評価基準が定まっている状況ではない。したがって、格子暗号を利用する際の留意点としては、①鍵長が長いというデメリットを理解したうえで、この鍵長に基づく暗号化処理が可能な計算機性能（計算能力、メモリ容量、ネットワーク帯域等）を備えた PC やサーバ等での利用を検討する、②パラメータや鍵を選択す

[43] 秘匿計算に関しては、1970 年代から「データを暗号化したまま計算する」というアイディアが知られていたが、この頃は任意の演算を暗号化したまま計算可能な方式が実現されておらず、乗算または加算の一方のみを暗号化したまま演算可能な技術（「準同型暗号」と呼ばれる）の実現に留まっていた。こうした技術では、「電子投票の集計」等の簡単な計算を実現できるものの、乗算と加算を組み合わせた複雑な計算（例えば、統計解析等）を実現することは困難であった。その後、2009 年にジェントリーにより、乗算と加算の両者を暗号化したまま計算可能な技術（「完全準同型暗号」と呼ばれる、Gentry[2009]）が提案され、同技術の応用範囲が広がるとともに、実用化に関する研究が活発化するきっかけとなった。

る際には、最新の攻撃実験に基づく安全性評価の結果（図表 23）を参考にするとともに、同評価の不確実性に伴うリスクを予め考慮したうえで、現時点の評価よりも高い安全性を確保できるパラメータを設定する、といったことが挙げられる。

将来、量子コンピュータが実用化された場合には、金融分野における情報システムの安全性を確保するために、主流の RSA 暗号や楕円曲線暗号から、格子暗号をはじめとする耐量子コンピュータ暗号への暗号アルゴリズムの移行を速やかに行う必要がある。一般論として、暗号アルゴリズムを移行する場合には、個々の金融機関における情報システムの改修が必要となるだけではなく、（通信で暗号を使用している場合には）通信の相手先のシステム改修も必要になる等、その影響範囲は大きいため、移行の準備期間も十分に確保する必要がある。RSA 暗号等から耐量子コンピュータ暗号への移行を行う際も同様の考え方に基づき、計画的に移行を進める必要があると考えられる。具体的には、①暗号アルゴリズムの変更に伴う既存システム（自社開発プログラムのみならず商用パッケージ製品も含む）における暗号アルゴリズムの利用箇所の把握や同アルゴリズムによって守られている情報資産の把握、②使用している暗号アルゴリズムが、万一、危殆化した場合における、①をもとにしたシステム毎の対応にかかる優先順位の策定、③量子コンピュータや、格子暗号をはじめとする耐量子コンピュータ暗号の研究動向を金融分野でフォローする体制の整備、④将来的に暗号アルゴリズムを変更することや、安全性の経年劣化に伴う鍵長の増加[44]等を想定したプログラム開発における拡張性確保、といった準備を今から進めていくことは、量子コンピュータ実現と主流の暗号アルゴリズムの危殆化という不測の事態への備えとして重要であろう。

[44] 格子暗号の安全性は、量子コンピュータの計算能力向上により、安全性が経年劣化することが想定されるため、従来の RSA 暗号や楕円曲線暗号と同様、安全性を確保するためには鍵長を適宜長くしていくという運用が必要と考えられる。

安全性評価指標	指標に基づく安全性を達成するために必要な鍵長（ビット）					
	現在主流の公開鍵暗号		格子暗号			
			実現方式固有の要因で安全性が低下する可能性			
			高い		低い	
	RSA 暗号	楕円曲線暗号	GGH 方式	NTRU 方式	AD 方式	LWE 方式
128	3,024	256	280×10^{6}	7,000	490×10^{15}	7.8×10^{6}
256	15,360	512	1.6×10^{9}	13,000	2.2×10^{18}	30×10^{6}

図表 25. 現在主流の公開鍵暗号と主な格子暗号の鍵長比較

備考：AD 方式、GGH 方式、LWE 方式については、Nguyen[1999]、Chen and Nguyen[2011]、Linder and Peikert[2011]、青野ほか [2015]の評価等に基づき鍵長を算出しており、NTRU 方式については、ANSI[2010]を参照した。また、RSA 暗号と楕円曲線暗号の鍵長は、NIST[2012]を参照した。

補論．行列の演算規則

ここでは、4 節(1)で紹介した LWE 方式の具体的な処理フローを理解するうえで最低限必要となる行列の演算規則を紹介する。本文中の Box 1 でも述べたとおり、$m \times n$個の数（整数や実数等）を長方形に並べた

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

は$m \times n$行列という（mは行の数、nは列の数と呼ばれる）。ここで、2 つの$m \times n$行列X,Yを、

$$X = \begin{bmatrix} x_{11} \cdots x_{1j} \cdots x_{1n} \\ \vdots \ \ddots \ \vdots \ \ddots \ \vdots \\ x_{i1} \cdots x_{ij} \cdots x_{in} \\ \vdots \ \ddots \ \vdots \ \ddots \ \vdots \\ x_{m1} \cdots x_{mj} \cdots x_{mn} \end{bmatrix}, \qquad Y = \begin{bmatrix} y_{11} \cdots y_{1j} \cdots y_{1n} \\ \vdots \ \ddots \ \vdots \ \ddots \ \vdots \\ y_{i1} \cdots y_{ij} \cdots y_{in} \\ \vdots \ \ddots \ \vdots \ \ddots \ \vdots \\ y_{m1} \cdots y_{mj} \cdots y_{mn} \end{bmatrix}$$

と表現したとき、行列の相等、加算と減算、スカラー倍、乗算はそれぞれ次のように定義される。

(1) 行列の相等

行列X,Yが以下の条件を満たすとき、XとYは等しい（相等である）という。

$$x_{ij} = y_{ij} \ (i = 1,2, \cdots, m \ , j = 1,2, \cdots, n).$$

(2) 行列の加算と減算

行列X,Y同士の加算と減算は、以下のように行われる。

$$X \pm Y = \begin{bmatrix} x_{11} \pm y_{11} \cdots x_{1j} \pm y_{1j} \cdots x_{1n} \pm y_{1n} \\ \vdots \qquad \ddots \qquad \vdots \qquad \ddots \qquad \vdots \\ x_{i1} \pm y_{i1} \cdots x_{ij} \pm y_{ij} \cdots x_{in} \pm y_{in} \\ \vdots \qquad \ddots \qquad \vdots \qquad \ddots \qquad \vdots \\ x_{m1} \pm y_{m1} \cdots x_{mj} \pm y_{mj} \cdots x_{mn} \pm y_{mn} \end{bmatrix}$$ （複号同順[45]）．

[45] 複号とは演算子「±」を意味し、複号同順とは式内の全ての複号について、上側か下側の符号のどちらかのみで解釈することを意味する。

(3)　行列のスカラー倍

行列において、行列以外の数k（整数や実数等）はスカラーと呼ばれる。そして、行列Xをk倍する演算（スカラー倍と呼ばれる）は、以下のように行われる。

$$kX = \begin{bmatrix} kx_{11} \cdots kx_{1j} \cdots kx_{1n} \\ \vdots \quad \ddots \quad \vdots \quad \ddots \quad \vdots \\ kx_{i1} \cdots kx_{ij} \cdots kx_{in} \\ \vdots \quad \ddots \quad \vdots \quad \ddots \quad \vdots \\ kx_{m1} \cdots kx_{mj} \cdots kx_{mn} \end{bmatrix}.$$

(4)　行列の乗算

$l \times m$行列 X と、$m \times n$行列 Y をそれぞれ

$$X = \begin{bmatrix} x_{11} \cdots x_{1m} \\ \vdots \quad \ddots \quad \vdots \\ x_{l1} \cdots x_{lm} \end{bmatrix}, \qquad Y = \begin{bmatrix} y_{11} \cdots y_{1n} \\ \vdots \quad \ddots \quad \vdots \\ y_{m1} \cdots y_{mn} \end{bmatrix}$$

と表現したとき、この 2 つの行列の乗算$X \cdot Y$は次のように行われる。

$$X \cdot Y = \begin{bmatrix} c_{11} \cdots c_{1n} \\ \vdots \quad \ddots \quad \vdots \\ c_{l1} \cdots c_{ln} \end{bmatrix}.$$

ここで、$c_{ij} = x_{i1}y_{1j} + x_{i2}y_{2j} + \cdots + x_{im}y_{mj}$ $(i = 1,2,\cdots,l\,;j = 1,2,\cdots,n)$である。なお、行列の乗算については、X の列の数と Y の行の数が一致していないと、上記の乗算を行うことができないことに留意されたい。

参考文献

青野良範・林　卓也・レ チュウ フォン・王　立華、「セキュリティアップデータブル準同型暗号を用いた秘匿データの線形回帰計算」、暗号と情報セキュリティシンポジウム、2015 年

宇根正志・神田雅透、「暗号アルゴリズムの 2010 年問題について」、『金融研究』、第 25 巻別冊第 1 号、日本銀行金融研究所、2006 年、31～72 頁

後藤　仁、「量子暗号通信の仕組みと開発動向」、『金融研究』、第 28 巻第 3 号、日本銀行金融研究所、2009 年、107～149 頁

四方順司・鈴木　譲・今井秀樹、「量子計算による ECDLP の効率的解法について」、電子情報通信学会技術研究報告, ISEC 99(329)、1999 年、9～15 頁

鈴木雅貴・菅原　健・鈴木大輔、「サイドチャネル攻撃に対する安全性評価の研究動向と EMV カード固有の留意点」、金融研究所ディスカッション・ペーパーNo. 2015-J-4、日本銀行金融研究所、2015 年

清藤武暢・四方順司、「公開鍵暗号を巡る新しい動き : RSA から楕円曲線暗号へ」、『金融研究』、第 32 巻第 3 号、日本銀行金融研究所、2013 年、17～50 頁

――――・――――、「高機能暗号を活用した情報漏えい対策『暗号化状態処理技術』の最新動向」、『金融研究』、第 33 巻第 4 号、日本銀行金融研究所、2014 年、97～132 頁

田村裕子、「ISO/TC68 における金融分野向け推奨暗号アルゴリズムの検討状況」、『金融研究』、第 28 巻第 1 号、日本銀行金融研究所、2009 年、173～206 頁

日経 BP 社、「量子コンピュータの開発 SF から現実に」、『日経エレクトロニクス』、No.1102、2013 年、57～68 頁

――――、「驚愕の量子コンピュータ」、『日経コンピュータ』、No.858、2014 年、24～39 頁

日立製作所、「約 1 兆の 500 乗通りの膨大なパターンから瞬時に実用に適した解を導く室温動作可能な新型半導体コンピュータを試作」、日立製作所、2015 年

富士通研究所、「世界初!暗号化したまま統計計算や生体認証等を可能にする準同型暗号の高速化技術を開発」、富士通研究所、2013 年

――――、「暗号化したまま検索が可能な秘匿検索技術を開発」、富士通研究所、2014 年

武藤健一郎、「SSL における暗号危殆化サンプル調査の報告」、PKI Day 2011、2011 年

安田雅哉・下山武司・小暮　淳、「準同型暗号による秘匿統計計算」、暗号と情報セキュリティシンポジウム、2015 年

Ajtai, Miklos and Cynthia Dwork, “A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence,” Proceedings ACM Symposium on Theory of Computing, 1997, pp.284-293.

Albrecht, Martin R., Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret, “Algebraic Algorithms for LWE Problems,” IACR Cryptology ePrint Archive, no.1018, 2014.

――――, ――――, ――――, ――――, and ――――, “On the Complexity of the BKW Algorithm on LWE,” Journal of Designs, Codes and Cryptography, 74(2), 2015, pp.325-354.

American National Standards Institute (ANSI), “X9.98: Lattice-Based Polynomial Public-Key Establishment Algorithm for the Financial Services Industry,” ANSI, 2010.

Aono, Yoshinori, Xavier Boyen, Le Trieu Phong, and Lihua Wang, “Key-Private Proxy Re-Encryption under LWE,” Proceedings of INDOCRYPT, LNCS 8520, Springer-Verlag, 2013, pp.1-18.

Bai, Shi and Steven D. Galbraith, “Lattice Decoding Attacks on Binary LWE,” Proceedings of Australasian Conference on Information Security and Privacy(ACISP), LNCS 8544, Springer-Verlag, 2014, pp.322-337.

Bennett, Charles H., Ethan Bernstein, Gilles Brassard, and Umesh Vazirani, “Strengths and Weaknesses of Quantum Computing,” SIAM Journal of Computing, 26(5), 1997, pp.1510-1523.

――――― and Gilles Brassard, “Quantum Cryptography: Public Key Distribution and Coin Tossing,” Proceedings of IEEE International Conference on Computers Systems and Signal Processing, 1984, pp.175-179.

Bos, Joppe W., Craig Costello, Michael Naehrig, and Douglas Stebila, “Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem,” IACR Cryptology ePrint Archive, no.599, 2014 (To appear in Proceedings of IEEE Symposium on Security and Privacy (IEEE S&P), 2015).

Brakerski, Zvika, Craig Gentry, and Vinod Vaikuntanathan, “(Leveled) Fully Homomorphic Encryption without Bootstrapping,” ACM Transactions on Computation Theory (TOCT), 6(3), 2014, pp. 13:1-13:36.

Brassard, Gilles, Peter Hoyer, and Alain Tapp, “Quantum Algorithm for the Collision

Problem," arXiv Quantum Physics, no.9705002, 1997.

Chen, Yuanmi and Phong Nguyen,"BKZ2.0: Better Lattice Security Estimates," Proceedings of ASIACRYPT, LNCS 7073, Springer-Verlag, 2011, pp.1-20.

Coppersmith, Don and Adi Shamir, "Lattice Attacks on NTRU," Proceedings of EUROCRYPT, LNCS 1233, Springer-Verlag, 1997, pp.52-61.

Dadush, Daniel, Oded Regev, and Noah Stephens-Davidowitz, "On the Closest Vector Problem with a Distance Guarantee," arXiv Data Structures and Algorithms, no.1409.8063, 2014.

D-Wave Systems, Inc., "D-Wave Systems sells Its First Quantum Computing System to Lockheed Martin Corporation," Press Releases, 2011.

EMVCo, "EMVCo Common Contactless Terminal Roadmap," General Bulletin, no. 43, EMVCo, 2009.

Gama, Nicolas, Phong, Nguyen, and Oded Regev, "Lattice Enumeration using Extreme Pruning," Proceedings of EUROCRYPT, LNCS 6110, Springer-Verlag, 2010, pp.257-278.

Gentry, Graig, "Fully Homomorphic Encryption using Ideal Lattices," Proceedings ACM Annual Symposium on the Theory of Computing (STOC), 2009, pp.169-178.

Goldreich, Oded, Shafi Goldwasser, and Shai Halevi, "Public-Key Cryptosystems from Lattice Reduction Problems," Proceedings of CRYPTO, LNCS 1294, Springer-Verlag, 1997, pp.112-131.

Grover, Lov K., "A Fast Quantum Mechanical Algorithm for Database Search," Proceedings of Symposium on Theory of Computing (STOC), 1996, pp.212-219.

Hoffstein, Jeffrey, Jill Pipher, and Joseph H. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem," Proceedings of Algorithmic Number Theory (ANTS), LNCS 1423, Springer-Verlag, 1998, pp.267-288.

Institute of Electrical and Electronic Engineers (IEEE), "IEEE1363.: Public-Key Cryptographic Techniques Based on Hard Problems over Lattices ," IEEE, 2009.

International Organization for Standardization (ISO), "Financial Services – Recommendations on Cryptographic Algorithms and Their Use - Standing Document," ISO, 2007.

Kannan, Ravi, "Improved Algorithms for Integer Programming and Related Lattice Problems," Proceedings of Symposium on Theory of Computing (STOC), 1983,

pp.193-206.

Klein, Philip, “Finding the Closest Lattice Vector When It’s Unusually Close,” Proceedings of ACM-SIAM Symposium on Discrete Algorithms, 2000, pp.937-941.

Koblitz, Neal, “Elliptic Curve Cryptosystems,” Mathematics of Computation, 48, 1987, pp.203-209.

Kuwakado, Hidenori and Masakatu Mori, “Quantum Distinguisher between the 3-Round Feistel Cipher and the Random permutation,” Proceedings of IEEE International Symposium on Information Theory (ISIT), 2010, pp.2682-2685.

Laine, Kim and Kristin Lauter, “Key Recovery for LWE in Polynomial Time,” IACR Cryptology ePrint Archive, no.176, 2015.

Lenstra, Arjen Klaas., Hendrik Willem Lenstra Jr, and László Lovász, “Factoring Polynomials with Rational Coefficients,” Journal of Mathematiche Annalen, 261(4), 1982, pp.515-534.

Linder, Richard and Chris Peikert, “Better Key Sizes (and Attacks) for LWE-Based Encryption,” Proceedings of CT-RSA, LNCS 6558, Springer-Verlag, 2011, pp.319-339.

Liu, Mingjie and Phong Nguyen, “Solving BDD by Enumeration: An Update,” Proceedings of CT-RSA, LNCS 7779, 2013, pp.293-309.

Lyuashevsky, Vadim, Chris Peikert, and Oded Regev, “On Ideal Lattices and Learning with Errors over Rings,” Proceedings of EUROCRYPT, LNCS 6110, Springer-Verlag, 2010, pp.1-23.

Matsumoto, Tsutomu and Hideki Imai, “Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption,” Proceedings of EUROCRYPT, LNCS 330, Springer-Verlag, 1988, pp.419-453.

McEliece, Robert J., “A Public-Key Cryptosystem based on Algebraic Coding Theory,” Jet Propulsion Laboratory DSN Progress Report, 42-44, 1978, pp.114-116.

Miller, Victor, “Use of Elliptic Curves in Cryptography,” Proceedings of CRYPTO, LNCS 218, Springer-Verlag, 1985, pp.417-426.

Martín-López, Enrique, Anthony Laing, Thomas Lawson, Roberto Alvarez, Xiao-Qi Zhou, and Jeremy L. O'Brien, “Experimental Realization of Shor's Quantum Factoring Algorithm using qubit Recycling,” Nature Photonics , 6, 2012, pp.773–776.

Micciancio, Daniele and Oded Regev, “Worst-case to Average-case Reductions based

on Gaussian Measures ," Journal of Computing, 37(1), 2007, pp.267-302.

———— and ————, "Lattice-based Cryptography," Post Quantum Cryptography, Springer-Verlag, 2009, pp.147-191.

National Institute of Standards and Technology (NIST), "Recommendation on Key Management – Part I: General," Special Publication (SP) 800-57, NIST, 2005a.

————, "Cryptographic Algorithms and Key Sizes for Personal Identity Verification," Special Publication (SP) 800-78, NIST, 2005b.

————, "Advanced Encryption Standard," Federal Information Processing Standardization (FIPS), 197, 2001.

————, "Recommendation on Key Management - Part I: General (revision 3)," Special Publication (SP) 800-57, 2012.

Nguyen, Phong, "Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto'97," Proceedings of CRYPTO, LNCS 1666, Springer-Verlag, 1999, pp.288-304.

———— and Jacques Stern, "Cryptanalysis of the Ajtai-Dwork Cryptosystem," Proceedings of CRYPTO, LNCS 1462, Springer-Verlag, 1998, pp.223-242.

Peikert, Chris, Vinod Vaikuntanathan, and Brent Waters, "A Framework for Efficient and Composable Oblivious Transfer," Proceedings of CRYPTO, LNCS 5157, Springer-Verlag, 2008, pp.554-571.

Regev, Oded, "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography," Journal of ACM, 56(6), 2009, pp.1-40.

Rivest, Ronald, Adi Shamir, and Leonard Adleman, "A Method of obtaining Digital Signatures and Public Key Cryptosystems," Communications of the ACM, 21, 1978, pp.120-126.

Schnorr, Claus Peter., "A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms," Journal of Theoretical Computer Science, 53(2-3), 1987, pp.201-224.

————, "Lattice Reduction by Random Sampling and Birthday Methods," Proceedings of Symposium on Theoretical Aspects of Computer Science, LNCS 2607, Springer-Verlag, 2003, pp.145-156.

Shannon, Claude, "Communication Theory of Secrecy Systems," Bell System Technical Journal, 28(4), 1949, pp.656-715.

Shikata, Junji, "Trends and Development of Information-Theoretic Cryptography," IEICE Transactions, 98-A(1), 2015, pp.16-25.

Shor, Peter, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring,” Proceedings of Foundations of Computer Science (FOCS), 1994, pp.124-134.

――――, “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer,” SIAM Journal on Computing, 26(5), 1997, pp.1484-1509.

Vandersypen, Lieven M. K., Matthias Steffen, Gregory Breyta, Constantino S. Yannoni, Mark H. Sherwood, and Issac L. Chuang, “Experimental Realization of Shor’s Quantum Factoring Algorithm using Nuclear Magnetic Resonance,” Nature 414, pp.883-887, 2001.

Vernam, Gilbert S., “Cipher Printing Telegraph Systems for Secret Wire and Radion Telegraphic Communications,” Journal of American Institute of Electrical Engineers, 45, 1926, pp.295-301.

以　上