プロジェクト報告書(最終) Project Final Report

提出日(Date) 2023/1/17

NO.17 暗号と数理とセキュリティ プロジェクト報告書

No.17 Cryptography, Mathematics, and Security project report

猪股莉記 Riki Inomata

1.背景

[GroupA(格子暗号班)]

インターネットが普及し、ネットワーク社会と化した現代において、様々なサービスのセキュリティを確保するための基盤技術として暗号技術が用いられている。中でも、公開鍵暗号方式と呼ばれる仕組みの暗号技術は、電子署名や暗号通信など私たちの身の回りの様々な場面で活用されている。一方で、量子コンピュータによってRSA暗号や楕円曲線暗号などの主流な公開鍵暗号の安全性が危殆化されている。実際に、米国立標準技術研究所は2030年までに現在主流である鍵長が2048bitのRSA暗号の使用の停止を推奨しており[1]、日本でも同様の基準を設けている[2]。そこで、私たちは量子コンピュータが発達した社会でも活躍されると考えられる耐量子計算機暗号に注目した。中でも格子暗号と呼ばれる暗号技術は量子アルゴリズムへの耐性を有するのみならず、暗号化状態処理技術という特性を持ち、実用化が期待されている。私たちは格子暗号によるテキストの暗号化と復号を可能とすることを目的とした。

[GroupB(楕円曲線暗号班)]

楕円曲線暗号にかかわるものとして、暗号、楕円曲線暗号、デジタル署名、ブロックチェーンについて説明し、現在存在する利用例についても紹介する。

現在用いられている暗号は、共通鍵暗号と公開鍵暗号の2種類に大別することができる。共通鍵暗号は、暗号化と復号に同一の鍵を用いる方式であり、公開鍵暗号は、公開鍵と秘密鍵という2種類の鍵を用いて暗号化と復号を行う方式である。

楕円曲線暗号は公開鍵暗号の中で主流と呼ばれるものの1つで、楕円曲線と呼ばれる曲線を用いた暗号である。公開鍵暗号の他の例としては素因数分解の困難性を用いたRSA暗号が存在するが、RSA暗号の安全性は、鍵の長さに強く依存している。現在ではRSA暗号の鍵長は2048ビットが推奨されることが多い。しかし、楕円曲線暗号は256ビットの鍵長で3072ビットの鍵長のRSA暗号と同等の高い安全性を実現できる。鍵長が短ければ、暗号の計算を高速化が可能であり、また、鍵の保管に必要な容量も小さくなる。楕円曲線暗号の主な用途としては、SSHやSSL/TLSのような安全な通信プロトコルや、デジタル署名が挙げられる。

デジタル署名は、公開鍵暗号に類似した改ざん防止手法である。デジタル署名の大きな特徴は本人証明と、署名したメッセージの内容が改ざんされていないことの証明を同時に行うことができる点である。デジタル署名では、署名者の秘密鍵とメッセージを用いて署名を生成するので、他のメッセージに同じデジタル署名を用いることはできない。これは、紙の書類への署名や捺印では実現できないデジタル署名特有の利点である。楕円曲線暗号を用いたデジタル署名アルゴリズムの例としてはECDSAやBLS署名が挙げられ、前者はBitcoin、後者はEthereumという暗号資産に利用されている。

ブロックチェーンは分散型のデジタル台帳と呼ばれる技術であり、ある一定の期間内の取引情報をまとめた台帳を1つのブロックとして、そのブロックをつなげたものである。ブロックチェーンの最大の特徴は改ざんが実質的に不可能ということである。このようなブロックチェーンの安全性を保証する技術の一つがデジタル署名である。ブロックチェーンの活用例としては、BitcoinやEthereumといった暗号資産が挙げられる。ブロックチェーンの問題点は、新たにブロックチェーンに参加するユーザーに専用のソフトウェアのダウンロードと、大容量のストレージを要求することである。大容量のストレージはブロックチェーンに記録されている過去から現在までの取引情報をダウンロードして保持するために要求される。例えば、Bitcoinでは、Bitcoin Coreと呼ばれるソフトウェアを公式に提供しているが、このソフトウェアには過去の取引情報も含

まれているため、ダウンロードには 350GB 以上のストレージが要求される[3]。また、このダウンロードには 1 週間以上の時間がかかる場合もある。もちろん、ブロックチェーン上での取引が続けば、それらの取引情報も記録する必要があるため、ブロックチェーンの取引に必要なストレージは今後も増え続けていくことがわかる。また、ソフトウエアのダウンロードといった行為は技術的な知識のない人々にとっては容易なことではなく、PC に関する基礎的な知識がない人がブロックチェーンに参加して取引を行うことは難しい。以上の理由から、ブロックチェーンは新規参入のための障壁が低いとは言えず、ブロックチェーン技術の普及を行うためには、これらの問題点を解決する必要がある。そして、これらの問題点に加えて、暗号資産の抱える問題点として、人々の持つ暗号資産に対する漠然とした不安がある。日本においてこの漠然とした不安を生み出した要因の一つとして、暗号資産取引所「Coincheck(コインチェック)」が受けたハッキングによる暗号資産の流出が挙げられる。この事件は、Coincheck に対する不正アクセスが原因で発生したとされていて、この事件で流出した暗号資産自体の安全性はその原因ではない[4]。取引所の安全性と暗号資産自体がもつ安全性を切り分けて考える必要があることを、人々に啓蒙していく必要がある。

2. 課題の設定と到達目標

[GroupA(格子暗号班)]

本プロジェクトでは次世代の暗号である格子暗号を用いてテキストの暗号化/復号を可能にすることを目標とした。目標を達成する上での課題が 2 つある。一つ目は今現在標準化が推進されている途中である格子暗号について研究をすすめることと、web アプリケーションの作成技術を身に着けることである。二つ目は格子暗号の方式の一つである GGH 方式が整数を、LWE 方式がビットを暗号化/復号の対象としているためそれらをどのようにテキストの暗号化/復号と結びつけるかというものがあげられる。

[GroupB(楕円曲線暗号班)]

本グループの課題は、ユーザーすなわち人々にソフトウェアのダウンロードや大容量のストレージを要求することなく、ブロックチェーン上での取引を体験してもらうことである。疑似的な暗号資産を用いたブロックチェーン上での取引を体験することで、人々のブロックチェーン上での取引とブロックチェーン技術そのものに対する理解促進を図る。以上の課題を解決するために、本グループの目的はブロックチェーン技術とそれにかかわるマイニングなどの作業が体験できる Web アプリケーションを作成し人々に体験してもらうことで、ブロックチェーン技術を啓蒙することである。既存の暗号資産のようなブロックチェーンに新たに参加するために必要な、専用のソフトウェアの導入や大容量のストレージを要求しない Web 上で完結したアプリケーションを作成することで、人々が手軽にブロックチェーン技術を体験できるようになることを目指す。

3. 課題解決のプロセスとその結果

[GroupA(格子暗号班)]

今回 web 上での暗号化に利用した方式の一つである GGH 方式は最近ベクトル問題の困難性を利用した格子暗号の方式の一つである。我々はこの方式を用いて文字列の暗号化を行うことを目指した。具体的なアルゴリズムとして、直交基底を秘密鍵、非直交基底を公開鍵として、平文に乱数を加えたモノを暗号文として扱う。この暗号文を復号するする際には、秘密鍵によって最も近い格子点を発見し、公開鍵によって連立方程式を解くことで復号を行っている。

ここからは、どのように文字を座標に当てはめて、GGH 方式を用いて暗号化し復号してるのか、具体例を使いながら説明していく。例えば、「あいう」という文字列を暗号化する場合、まず「あいう」を整数値に変換すると、「4196743176255353094534」となる。この整数列を 2 桁ずつに分割して「41,96,74,‥,45,34」のようし、配列にこの順番で格納する。その後に、この「配列の長さ」を頭に、「バイト列の長さ」と「整数列の長さ」を順に配列の後ろに格納する。例を用いると、「13,41,96,74,‥,45,34,22,9」のようになる。その後に、格子の次元に合わせるために 0 を格納していく。例を用いると、「13,41,96,74,‥,45,34,22,9,0,0,0‥,0」となる。その後に、この配列を暗号化する。復号した後の処理では、配列から、「配列の長さ」、「バイト列の長さ」、「整数列の長さ」の値を排を行った。更に余分な 0 を排除して、「41,96,74,‥,45,34」という状態に戻し、これらを結合することによって日本語に戻している。また、格子基底の次元は 200 と設定した。GGH 方式は最近ベクトル問題の困難性を利用した格子暗号の方式の一つである。我々はこの方式を用いて文字列の暗号化を行うことを目指した。具体的なアルゴリズムとして、直交基底を秘密鍵、非直交基底を公開鍵として、平文に乱数を加えたモノを暗号文として扱う。この暗号文を復号するする際には、秘密鍵によって最も近い格子点を発見し、公開鍵によって連立

方程式を解くことで復号を行っている。
ここからは、どのように文字を座標に当てはめて、GGH方式を用いて暗号化し復号してるのか、具体例を使いながら説明していく。例えば、「あいう」という文字列を暗号化する場合、まず「あいう」を整数値に変換すると、「4196743176255353094534」となる。この整数列を2桁ずつに分割して「41,96,74,‥,45,34」のようし、配列にこの順番で格納する。その後に、この「配列の長さ」を頭に、「バイト列の長さ」と「整数列の長さ」を順に配列の後ろに格納する。例を用いると、「13,41,96,74,‥,45,34,22,9」のようになる。その後に、格子の次元に合わせるために0を格納していく。例を用いると、「13,41,96,74,‥,45,34,22,9,0,0,0‥,0」となる。その後に、この配列を暗号化する。復号した後の処理では、配列から、「配列の長さ」、「バイト列の長さ」、「整数列の長さ」の値を排を行った。更に余分な0を排除して、「41,96,74,‥,45,34」という状態に戻し、これらを結合することによって日本語に戻している。また、格子基底の次元は200と設定した。
また、もう一つの方式であるLWE方式はLWE問題という数学問題の困難性を安全性の根拠としている。座標が整数全体で表される格子ではなく素数qにより制限された、有限体上の格子を利用している。米国立標準技術研究所は耐量子暗号アルゴリズムとしてこのLWE問題の一つを利用した方式を標準化することを発表している。[5]
LWE問題とはノイズが入った連立合同式を解くことを指す(図1)。これを行列とベクトルで表すと図2のようになり、つまりLWE問題とは素数qを法とする有限体Zq上からとった誤差eを付加した連立合同式について、A,bが与えられたとき、As + e ≡ b (mod q) を満たすベクトルsを求める問題と言える。

$$\begin{cases} 14s_1 + 15s_2 + 5s_3 + 2s_4 + e1 \equiv 8 \pmod{17} \\ 13s_1 + 14s_2 + 14s_3 + 6s_4 + e2 \equiv 16 \pmod{17} \\ 6s_1 + 10s_2 + 13s_3 + s_4 + e3 \equiv 12 \pmod{17} \\ 10s_1 + 4s_2 + 12s_3 + 16s_4 + e4 \equiv 3 \pmod{17} \end{cases}$$

図1. LWE問題の例

$$\mathbf{A} \cdot \mathbf{s} + \mathbf{e} = \mathbf{b} \pmod{q}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_m \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{bmatrix}$$

図2. LWE問題を行列とベクトルで表した図
今回LWE方式をWebアプリケーションに導入する上で工夫した点は、入力されたテキストについて、unicode文字で暗号分を出力できるようにし、入力されたunicode暗号文を復号できるようにした点である。具体的には、入力された文字をビット列に変換し、ビット一つ一つについて暗号化と復号を行った。その際、従来の暗号化方法だと暗号文が整数で出力されるので、それをさらにUnicode化することで暗号文をUnicodeテキスト出力できるようにした。復号の際は、入力されたUnicode暗号文を一つのビットを暗号化することによってできる整数暗号文のサイズごとに区切り、それぞれで復号することによりもともとの整数暗号文を導出し、従来のLWE方式の復号方法を用いることでビット列を導出した。最後にビット列をテキストに変換することで復号を可能とした。

[GroupB(楕円曲線暗号班)]
ブロックチェーン技術について理解しないまま、その実装を行うことは容易ではないため、我々はブロックチェーンに関する情報収集を十分に行ったうえで、どのようにブロックチェーンを実装するべきかを検討し、実装を行った。
まず、ブロックチェーンに関する情報収集は主に情報ライブラリーで関連書籍を検索することから始めた。複数の関連書籍が見つかったので、グループメンバーが各々別の書籍を借りて読み、それぞれの情報を共有することで、学びを深めた。その後、現状のブロックチェーンの応用例である暗号資産やサプライチェーンマネジメントなどについて主にインターネットを活用して調べた。一般の人々にはブロックチェーンと暗号資産はほぼ同じものであると認識されていることから、その誤解を解いたうえでブロックチェーン技術の啓蒙を行うには暗号資産に関する製品を作る必要があると私たちは考えた。しかし、実際に使われている通貨と対応させた暗号資産を扱うのは簡単ではないため、実際に使われている通貨とは関係ない疑似的な暗号資産を実装することが望ましいという結論に達した。そして、暗号資産の応用例や暗号資産の実装例について調べることで、どのような手法でブロックチェーン、そして独自の暗号資産を実装するべきかを理解した。既存のライブラリ、例えばデジタル署名のライブラリなどを用いることで、簡単なブロックチェーンを実装することは難しくないことが分かったため、ライブラリを用いずにデジタル署名を実装することと、ブロックチェーンの全体を実装することを切り分けて考えることで独自の暗号資産を実装するに至るのではないかという考えに至った。最後に、ブロックチェーンの全体

を実装することについては、クラウド上に疑似的なブロックチェーンを構築することで、取引やブロックが可視化するようにした。クラウド上のブロックチェーンについては研究があまり進んでいないこともあり、様々な文献を参考にしながら実装をした。デジタル署名のライブラリを用いない実装は、主に[6]と[7]を参考にした。ライブラリを用いないデジタル署名の具体例について学んだ後、その具体例を応用してブロックチェーンにはほとんど用いられていないが有望なデジタル署名を自らの手で実装した。

4. 今後の課題

[GroupA(格子暗号班)]

全体の課題として「テキストファイルを介さずweb上で格子暗号の暗号化、復号をするシステムを作成」が挙げられる。後期に作成した格子暗号を用いての web上での文字の復号に関して、現在は暗号化情報のテキストファイルを選び、秘密鍵のテキストファイルを選んでからでないと平文への復号はできない。現在のこのシステムをファイルを介さずに行うことが課題である。

次はGGH方式における課題で、一つ目は実行時間の長さである。公開鍵の作成においての実行時間がとても長くなってしまっているためこの問題を解決する必要がある。二つ目は乱数の設定に関してである。送信者が送る最近ベクトル問題の暗号化に用いられる乱数の範囲が短すぎると容易に解読されてしまうが、大きすぎても復号に時間がかかってしまう。この乱数の扱いについて考える必要がある。三つ目は暗号文への変換についてである。私たちは文字列を暗号化する際に、文字列全体を整数化し、その値を2桁ずつ分割して平文を作成することで文字列の大きさを小さくしたが、これでは文字列が長かった場合に、二つの暗号文を送らなければならない。しかし、一文字ずつ座標の値としてしまえば、暗号文の長さを大きくしてしまう。この暗号文の大きさにバランスについて考察していかなければならないと考える。

最後にLWE方式ページについての今後の課題が2つある。1つ目は暗号文のサイズが非常に大きくなってしまうことである。二つ目の課題はパラメータの適切な設定についてである。今後、別種類のLWE暗号について研究を進め、暗号文のサイズや鍵長を小さくしたり、パラメータを変更させたりしながら、安全性の評価を行うことで最適なパラメータを発見したい。

[GroupB(楕円曲線暗号班)]

今後の展望としては、EC-Shcnorrを本プロジェクトで実装したブロックチェーンに組み込むことが挙げられる。これにより、一人で複数の秘密鍵や公開鍵を利用することがあっても、鍵を線形和でまとめて用いることができる。また、ECDSAにはある条件を満たす2つの署名が同一の正しい署名として認識されてしまう脆弱性が存在するが、EC-Schnorrにはその脆弱性が存在しないため従来のブロックチェーンよりも優れた安全性を持ったブロックチェーンを実装できる可能性がある。

参考文献

[1]経済産業省(2022) 暗号強度要件(アルゴリズム及び鍵長選択)に関する設定基準.
https://www.cryptrec.go.jp/list/cryptrec-ls-0003-2022.pdf (2023/07/14アクセス)

[2]NIST(2020) Recommendation for Key Management.
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf(2023/07/14 アクセス)

[3] Bitcoin.org, "Bitcoin Coreのダウンロード",
https://bitcoin.org/ja/download (参照 2023/12/22)

[4] Coincheck, "暗号資産NEMの不正送金に関する質問", https://coincheck.com/ja/info/faq¥_nem (参照 2023/12/22)

[5] digicert(2023) 米国商務省標準化技術研究所 (NIST) が耐量子暗号標準を発表：最新の状況.
https://www.digicert.com/jp/blog/nist-pqc-standards-are-here(2023/8/26アクセス)

[6] Jimmy Song著, 星野靖子訳 (2020) "プログラミング・ビットコイン ： ゼロからビットコインをプログラムする方法", オライリー・ジャパン

[7] 山崎重一郎, 安土茂亨, 金子雄介, 長田繁幸 (2021) "ブロックチェーン技術概論 ： 理論と実践", 講談社