暗号と数理とセキュリティ

Cryptography, Mathematics and Security

プロジェクトリーダー:後藤瞭介/Ryousuke Goto

１．背景

現代社会において、情報通信技術の発展は目覚ましいものがあり、それに伴いネットワーク社会の形態も急速に進化している。インターネットを介した情報のやり取りが日常生活の一部となり、個人や企業にとってネットワークの利用は不可欠なものとなっている。しかし、このような利便性の裏には、セキュリティに関する多くの課題が存在する。データ漏洩やサイバー攻撃のリスクは年々増加しており、セキュリティ対策の重要性が一層高まっている。セキュリティの重要性については広く認識されているものの、その具体的なシステムやコストに関する理解は一般的には十分とは言えないだろう。例えば、ブロックチェーン技術に関連する話題としてビットコインが有名だが、その裏で国一つ分に匹敵する電力がただの計算処理のために消費されていることを知っている人は少ないだろう。このような技術の持つ社会的課題に対する認識を深め、解決策を模索することが求められているといえるのではないだろうか。本プロジェクトでは、公開鍵暗号方式やブロックチェーンなど世の中にある暗号技術について学習し、それらに関連のある社会に存在する課題を踏まえ、解決することを目的とした。

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することが可能であり、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り、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署名が挙げられる。

特に近年の技術革新においては、楕円曲線暗号のハードウェア実装が重要な役割を果たしている。例えば、自動運転技術においては、車両が相互に通信し、瞬時に意思決定を行うために、安全な署名プロセスを極めて高速で実行する必要がある。このような環境では、従来のソフトウェアベースの暗号処理では処理速度に限界があり、ハードウェアでの実装が不可欠となる。ハードウェアアクセラレーションを活用することで、署名生成および検証の速度を飛躍的に向上させ、リアルタイムでの安全な通信を実現できる。

自動運転車では、膨大な数のセンサーが取得するデータを処理し、道路状況や他の車両との距離を判断しなければならない。これらのデータの信頼性を確保するためには、送受信される情報に対してデジタル署名を付与し、改ざんされていないことを保証する必要がある。特に、車両同士が数ミリ秒単位で情報を交換する場合、署名処理の遅延は致命的なリスクを生じる可能性がある。そのため、楕円曲線暗号を用いた効率的なデジタル署名のハードウェア実装は、自動運転技術の普及において非常に重要な課題であるといえる。

具体的な実装例としては、専用の暗号処理チップを車両の通信モジュールに組み込むことが挙げられる。このチップは、ECDSAなどの楕円曲線暗号を高速に処理できるよう最適化されており、署名生成や検証の処理時間を最小限に抑える。また、これにより電力消費も抑えられるため、車両の省エネ性能にも寄与する。今後、自動運転技術の進展とともに、楕円曲線暗号を用いたハードウェア実装はますます重要な位置を占めることになるだろう。

２．課題の設定と到達目標

本プロジェクトは、デジタル社会における情報セキュリティの課題を解決し、より安全で信頼性の高い情報社会の構築に貢献することを目的とする。情報セキュリティは、現代のデジタル社会において重要な要素であり、暗号技術はその根幹を支える基本技術の一つである。特に、データの保護、認証、プライバシーの確保といった分野において、暗号技術

は不可欠な存在であり、その理解と応用は、社会のあらゆる分野での安全性向上に資する。

具体的に前期では、暗号技術の基礎である数理の学習を行うことを主眼とする。これには、数論、代数、離散数学などの基礎的な数理知識の修得が含まれ、これらの知識をもとに暗号アルゴリズムの動作原理を理解することを目指す。また、情報セキュリティの理論的裏付けを深めることで、今後の応用研究に必要な基礎を確立することが期待される。
後期は、現代社会における実用的な課題として、自動運転技術やIoTデバイスなどの分野に着目する。これらの分野では、膨大なデータがリアルタイムで処理される必要があり、暗号処理の高速化が重要な課題となっている。特に、自動運転車においては、通信データの暗号化と復号を低遅延で行うことが求められ、安全性と効率性の両立が必須である。このような背景を踏まえ、ハードウェアによる暗号処理の最適化が重要な研究テーマだと考えている。そこで、前期で得た数理的基礎を活用し、後期において楕円曲線暗号のハードウェア実装を行うことを目指す。楕円曲線暗号は、従来の公開鍵暗号方式と比較して、同等の安全性をより短い鍵長で実現できる点で注目されている。また、FPGA(Field-Programmable Gate Array) を用いたハードウェア実装は、柔軟性と性能の両面で利点があり、暗号技術の高速処理において効果的な手法である。加えて、設計フローの実践を通じて、効率的なハードウェア開発プロセスを追求することは、ソフトウェアとハードウェアの設計プロセスにおける共通点や相違点を理解し、ハードウェア実装における課題を特定することにつながる。また、暗号技術の実用化に向けたハードウェア最適化の可能性を探り、情報セキュリティ分野における研究の発展に寄与し、デジタル社会のさらなる安全性向上に向けた貢献を果たすことにつながると考えた。
そのため、ソフトウェア開発における設計フローを活かしながら、高位合成を通して、ハードウェア実装に取り組むことで、その有用性を確認して、記録しておく。さらに、実装したものとソフトウェアベースの暗号処理との性能比較を通じて、ハードウェア実装の有効性を検証する。これにより、実用的な暗号技術の社会実装に向けた知見を深めることを最終的な目的とした。

３．課題解決のプロセスとその結果

楕円曲線は現代社会の暗号技術において、多岐にわたる分野で利用されている。具体的には、ブロックチェーンの台帳技術やWeb通信技術における重要な証明書であるTLS1.3などに応用されている。本プロジェクトでは、現代社会で広く普及している公開鍵暗号技術である楕円曲線暗号およびその応用技術に焦点を当てることとした。

そのため、前期においては楕円曲線の基礎理論の理解を目的とし、Joseph H. Silverman氏の著書『はじめての数論』第4版を基に学習を進めた。特に公開鍵暗号において必要不可欠な整数論の知識に重点を置きながら、楕円曲線に関連する数理の理解を深めた。
具体的には、数論的な考え方を身につけるために、円周上の有理点を求めることから始めた。この考え方を応用することで、楕円曲線上の点PとQに対して、P+Qを定めることが出来る。この性質は、楕円曲線暗号の計算において重要な役割を果たしている。

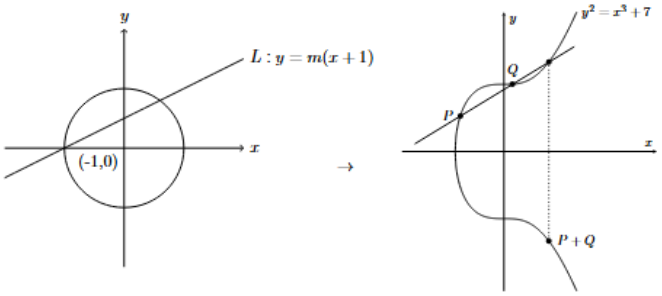

図1 円と楕円曲線のグラフ

また、RSA暗号や楕円曲線暗号には剰余の概念が深く関わっている。従って、合同式の復習を行い、剰余の世界での計算方法について学んだ。また、剰余の概念を応用し、中国の剰余定理についても学んだ。この定理はRSA暗号の計算において重要な役割を果たしている。その後、白勢先生による講話で実際にこれらの数理がどのように暗号技術で用いられているのか理解を深めた。
さらに、担当教員の元、公開鍵暗号について、楕円曲線暗号(ECC)は公開鍵暗号の1つであり、楕円曲線上の有理点の加算を用いて暗号化と復号を行うものである。

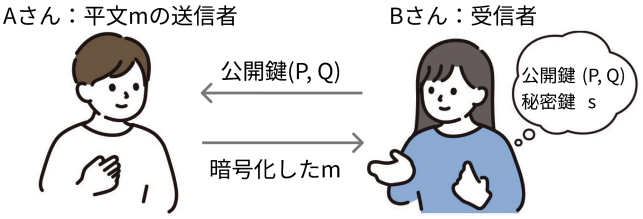

図2 公開鍵暗号のしくみ

ECCは、RSA暗号に比べてより短い鍵長で同等の暗号強度を実現できるため、高速処理が可能である。ECCの安全性の根拠は、公開鍵から秘密鍵を求める計算に膨大な時間がかかることにあると学んだ。
後期のハードウェア実装にはAMD(旧:Xilinx)のFPGAを利用した。従って、AMDが提供しているソフトウェアで高位合成を行う。今回のプロジェクトでは2023年に新しく登場したVitis統合IDEを主に使用した。Vitis統合IDEは既存のVitis HLSとVitis IDE (Vitis Classic) を融合したものであり、Xilinxが今後サポートし続ける高位合成ツールである。Vitis統合

IDEの学習はAMDが提供するユーザーガイドやリファレンスマニュアルをもとに行った。また学習の一貫として、今後本プロジェクトが発展することがあれば、それが容易になるようハードウェア実装の方法を記録に残した。まずはじめに開発に必要なソフトウェアをダウンロードする。インストールの完了後、Vitisの機能をすべて利用する為にVitis HLSの開発者認証を行う。これにより、高位合成の結果を比較するために必要なFunction Call GraphやSchedule Viewerの利用が出来る。続いて、ハードウェア実装に向けて、まずは HLS コンポーネントを作成する。HLS コンポーネントでは、アルゴリズムの設計、C シミュレーション、高位合成、C/RTL 協調シミュレーションを行う。その後、FPGAに書き込んで動作確認するという流れである。

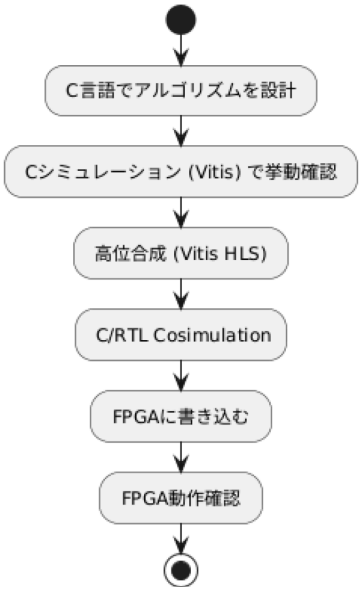

図3 ハードウェア実装の流れ

結果として、目標である高位合成を用いた楕円曲線暗号のハードウェア実装による処理速度の確認を行うため、楕円曲線暗号の支配的な処理であるスカラー倍算のアルゴリズムの作成を行った。はじめにスカラー倍算のアルゴリズムをC言語で作成し、動作確認を行った。続いてVitis上で、アルゴリズムを設計、アルゴリズムのシミュレーション、高位合成、ハードウェア記述言語で記述された回路のシミュレーションを行った。FPGA本体での動作確認まではたどり着かなかったものの、楕円曲線暗号の主要な計算のシミュレーションは正確に行うことができた。スカラー倍算のアルゴリズムの動作確認の過程において、安全強度に比例する数pを大きくするにつれてCシミュレーションを行うのが困難になる問題が生じた。問題解決の為にアルゴリズムの高速化を行った。Vitis上のヒートマップを用いて処理時間を要している部分を確認をしたところ、スカラー倍算で利用される逆元計算のアルゴリズムに原因があった。シンプルなアルゴリズムを利用していた為、pが増加するにつれて指数関数的に処理時間を要するものであった。具体的には、安全強度λ=128の場合約1.15×10^77回の処理を要していた。これを解決する為に、逆元計算のアルゴリズムをバイナリGCD法を利用する物に変更した。これにより、処理時間は直線的な増加をする物に改善した。λ=128の場合約512回の処理で済むようになった。また、楕円曲線暗号において暗号化の支配的な処理は2回のスカラー倍算、復号の支配的な処理は1回のスカラー倍算であることから、結果的にアルゴリズムの高速化は楕円曲線暗号そのものを高速化したといえる。特に、λ=128以上の安全強度を考慮した場合にも、処理時間の増加を抑えることに成功した点は、今後の実装において重要な意義を持つ。また、FPGA実機でのさらなる最適化に向けた検討が必要であり、異なる曲線や安全強度における性能比較を進めることで、実用的な暗号システムの開発が期待される。今回の手法が他の暗号プロトコルにも応用可能かについても、今後の課題として検討する価値がある。

図4 アルゴリズムの高速化

4. 今後の課題

本研究の成果は、ECCのハードウェア実装における効率化の実現可能性を示したという点で意義がある。スカラー倍算の高速化を基盤として、点加算や点二重化といった他の計算処理の最適化に着手することで、さらに実用的で高性能な暗号システムの開発が期待される。しかし、実際にFPGAに書き込んで実装する所までは至らなかったので、Vitisを使用したCシミュレーションの結果をもとに、今後は実際のFPGAハードウェア実装を進めていくことが必要である。シミュレーションで得られた処理時間と、実際のハードウェアでの実行時間が一致するかどうか検証し、ハードウェア実装における効率性の確認を行う。また、この比較検証を通じて性能向上を目指す。加えて、Vitisのヒートマップ機能を活用して、コードのボトルネックを特定しアルゴリズムの改良により、安全性強度λ=128の場合でも処理時間が現実的な範囲内に収まることが確認されたが、このスケーラビリティが他の安全強度設定や異なる楕円曲線においても同様に機能するかについては、さらなる検証が必要である。特に、安全強度の向上に伴う計算資源の消費バランスを考慮し、アルゴリズムのさらなる改良の可能性を検討する必要がある。バイ

ナリGCD法の適用により効率化が実現されたことについても、楕円曲線暗号の異なるパラメータセットや計算環境において同様の成果を得られるかどうかは未解明の部分が多い。本手法が適用可能な領域を明確にし、場合によっては他の補完的なアルゴリズムの採用を検討することが、さらなる性能向上につながると考えられる。加えて、本研究では、C言語で作成したアルゴリズムをVitis上で高位合成することで効率的な開発が可能であった。この手法は、開発期間の短縮や設計の可視化において大きな利点を持つが、開発環境に依存した最適化の限界や、FPGA実機での実行時の動作特性への影響を十分に評価する必要がある。特に、Vitisのヒートマップ機能を用いたボトルネックの特定が効率的であった一方、これがFPGA上での物理的な動作特性とどの程度一致するかは、今後の詳細な評価が求められる。また、実機の実装まで行えなかった要因として、Vitis周りのソフトウェアについてVitisIDEがここ数年で新しくなっており、ドキュメントがあまり充実しておらず苦労した点が挙げられる。FPGA開発における最新のソフトウェアツールは、高位合成の効率化に貢献する一方で、開発者の学習コストが増加している現状も課題である。今後は、異なる開発環境の比較も行い、最適な開発フローを見出すことで、FPGA実装をより円滑に進めるための基盤を構築していく必要がある。

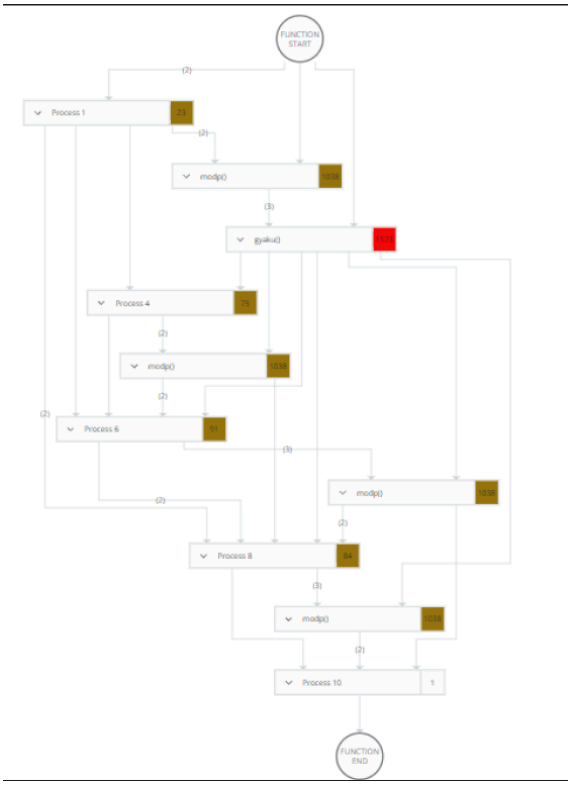

図5 Vitisのヒートマップの例

赤くなっている所がボトルネックとなっている部分である。

参考文献

[1]はじめての数論 原著第4版 J.H.シルヴァーマン著 鈴木治郎訳

[2]FPGA大全 第2版 小林優著

[3]Stoll, C., Klaaßen, L. & Gallersdörfer, U. The carbon footprint of bitcoin. Joule 3, 1647-1661 (2019).

[4]Vitis統合IDEおよびコマンドラインリファレンスマニュアル(UG1553), Advanced Micro Devices Inc., https://docs.amd.com/r/ja-JP/ug1553-vitis-ide, Jan. 2023 (accessed Jan. 2025).

[5]Vitis高位合成ユーザーガイド(UG1399), Advanced Micro Devices Inc., https://docs.amd.com/r/2023.1-日本語/ug1399-vitis-hls, Jan. 2023 (accessed Jan. 2025).