公立はこだて未来大学 2017 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2017 System Information Science Practice

Group Report

プロジェクト名

FUN-ECM プロジェクト

Project Name

FUN-ECM Project

グループ名

A グループ

Group Name

A Group

プロジェクト番号/Project No.

11-A

プロジェクトリーダ/Project Leader

1015014　中島俊平　Shunpei Nakajima

グループリーダ/Group Leader

1015014　中島俊平　Shunpei Nakajima

グループメンバ/Group Member

1015014　中島俊平　Shunpei Nakajima
1015082　福永慧　Kei Fukunaga
1015112　小澤貴也　Takaya Ozawa
1015182　落合航平　Kouhei Otiai
1015195　広瀬大樹　Taiki Hirose
1015202　金子真澄　Masumi Kaneko
1015237　水上敬介　Keisuke Mizukami
1015260　外山拓　Taku Toyama

指導教員

白勢政明　由良文孝

Advisor

Masaaki Shirase　Fumitaka Yura

提出日

2017 年 7 月 26 日

Date of Submission

July 26, 2017

概要

私達のプロジェクトの目的は，より大きな桁数の素因数を見つけることである．素因数分解が重要であることの背景として，RSA 暗号がある．RSA 暗号は，40 年前に考案された初めての公開鍵暗号で，現在でもデジタル署名などで利用されている．RSA 暗号は，「大きい桁数の 2 つの素数からなる合成数を素因数分解することが難しい」ということで安全が保証されているが，近年，より高い安全性を持つ楕円曲線暗号が利用されてきている．そこで FUN-ECM プロジェクトでは，楕円曲線法（ECM）を用いて，より大きい桁数の素因数分解を行うプログラムを作成し，RSA 暗号の安全性について検証を行う．更に私たちは，大きい数の素因数分解をランキングしたサイトである，ECMNET[1] や STUDIO KAMADA[2] へのランクインを目標として掲げ活動を行った．

まず私達は，理論班とプログラム班に分かれ活動を行っている．理論班の目的は，ECM の高速化のための論文を理解し，プログラム班に引き継ぐことである．プログラム班の目的は，理論班が準備したアルゴリズムを実装し，高速化できているか確かめることである．

理論班は，Stage2 をより高速に行えるアルゴリズムの発見・理解を目標とした．昨年度のプログラムに使われていた理論の理解，Baby-step Giant-step 法の理解，PARI/GP での実装を行った．

プログラム班は，前年度に作成された素因数分解プログラムをさらに高速化することを目標とした．昨年度のプログラムで余分に座標変換を行っていた部分のコードの修正，並列処理の強制終了の機能の追加，Baby-step Giant-step 法の実装，Stage2 の実行速度のテストを行った．

以上の 2 つの班が互いに足りない部分を補完しあい活動を行った．その結果，Stage2 の高速化に成功し，素因数分解を高速に行えるプログラムが完成した．

キーワード　素因数分解，楕円曲線法，ECMNET，エドワーズ曲線，拡張射影座標，RSA 暗号，Baby-step Giant-step 法

（※文責: 中島俊平）

Abstract

The goal of our project team is to find prime factor as large as possible. Factorizations in prime numbers have become more important because of RSA cryptosystem. RSA cryptosystem which was invented 40 years ago is still being used for digital signatures. RSA cryptosystem has been seemed to be safe because it is difficult to prime factorize a composite number composed of two prime numbers with large number of digits. However, elliptic curve cryptography has been used in recent years because it's safety is higher than RSA. Therefore, we create a program that performs prime factorization with a larger number of digits by the elliptic curve method. Then, we verify safety of the RSA cryptosystem. For that purpose, we have carried out activity with the goal of ranking in ECMNET[1] and STUDIO KAMADA[2]. ECMNET and STUDIO KAMADA is a website ranked by large number of prime factorization.

First, we divided into two group that Theory team and Program team. The purpose of Theory team was to understand essays for ECM speed up, and to hand over it to Program team. The purpose of Program team is to implement algorithm which prepared Theory team and to validate whether or not we can do increase in speed of program.

The theory group aimed for algorithmic discovery and understanding to be able to perform Stage2 speed up. We understood a theory used for a program of last year, we understood Baby-step-Giant-step, we implemented PARI/GP.

Programming group aimed to speed up the prime factorization program created in last year. We modified the code of the part where the coordinate conversion was extra last year program, implemented Baby-step Giant-step method, added the function of forcibly terminating parallel processing, and tested speed of Stage 2.

These two groups complemented each other missing parts and carried out activities. As a result, we succeeded in speed up Stage 2, and completed a program capable of fast prime factorization.

Keyword Elliptic Curve Method, prime factorization, ECMNET, Twisted Edwards Curve, Extended Twisted Edwards Coordinates, RSA cryptosystem, Baby-step Giant-step method

（※文責: 中島俊平）

目次

第 1 章　背景 **1**
1.1　本プロジェクトの背景 1
1.2　ECMNET とは 1
1.3　課題の概要 1

第 2 章　到達目標 **3**
2.1　本プロジェクトにおける目的 3
2.2　課題達成の為の班分け 3

第 3 章　活動内容 **4**
3.1　基礎学習 4
3.2　理論班 6
3.2.1　ECM の原理の理解 6
3.2.2　Baby-step Giant-step 法の理解 6
3.2.3　Montgomery curve の導入の検討 7
3.3　プログラミング班 8
3.3.1　座標変換の際の冗長なコストの削減 8
3.3.2　ソースコードの改善 8
3.3.3　計算方法の確立 9
3.4　中間発表 10
3.4.1　準備 10
3.4.2　発表 11

第 4 章　プロジェクト内のインターワーキング **12**

第 5 章　活動成果 **14**
5.1　理論班 14
5.2　プログラミング班 14

第 6 章　まとめ **16**
6.1　前期活動結果 16
6.2　後期の展望 16

参考文献 **17**

第 1 章　背景

大きい桁数の素因数分解は近年重要になっている．そこで，楕円曲線法を利用し素因数分解を行い，ECMNET にランクインすることが私たちの目的である．

（※文責: 中島俊平）

1.1　本プロジェクトの背景

暗号化技術は，情報の保護やコンピュータセキュリティにおいて欠かせない技術である．ファイルの暗号化の他に，HTTPS や，無線 LAN における通信など多くの場面で暗号化技術が利用されている．しかし，暗号化技術は常に進化する攻撃方法により解読の脅威に晒されている．様々な攻撃方法から安全な暗号アルゴリズムを作成するためには，作成する側が暗号解読の方法を知る必要がある．暗号の安全性評価には暗号解読の技術が利用されていて，暗号の強度は暗号解読に必要な情報量と計算量によって評価される．今回のプロジェクトでは，その暗号解読アルゴリズムの 1 つである，楕円曲線法を学ぶ．

現在，有名な公開鍵暗号の 1 つに RSA 暗号がある．RSA 暗号は，桁数が大きい合成数の素因数分解が困難であることを安全性の根拠とした暗号である．RSA 暗号を解読する時は合成数の元となる 2 つの素因数を見つけ出す必要がある．ECM では，与えられた曲線の点が無限遠点になることによって，因数が発見される．この性質を利用して RSA 暗号を解読する．

楕円曲線法には Stage1 と Stage2 があり，Stage1 で素因数分解できなかった場合，Stage2 で素因数分解を試みる．前年度のプロジェクトでは Stage1 のプログラムは完成していたため，今年度では Stage2 の完成させることを課題とした．また，ECMNET という暗号解読した素因数の大きさを競うサイトがあり，Stage2 を実装することで ECMNET でのランクインをすることを目標として掲げた．

（※文責: 福永慧）

1.2　ECMNET とは

ECMNET とは，楕円曲線法を用いて解読された素因数の大きさをランキング形式で競う Web サイトである．ECMNET にランクインするためには，現在登録されている素因数よりも大きな素因数を見つける必要がある．

（※文責: 福永慧）

1.3　課題の概要

本プロジェクトでは既に Stage1 におけるプログラムが完成していたため，Stage2 を作成する．Stage2 を完成させることで，より大きな合成数を素因数分解することを目指す．また，本プロジェ

クトでの活動を Web サイトを用いて，FUN-ECM の情報を外部に発信する．

（※文責: 福永慧）

第 2 章　到達目標

2.1　本プロジェクトにおける目的

FUN-ECM で ECMNET に記載されるためには既存のプログラムを改善し，単位時間ごとの計算速度を向上させる必要がある．そのため，この目的を達成するために 2 つのことを実施した．

- 既存のプログラムの高速化を図る
- Stage2 においての新たなアルゴリズムの実装

（※文責: 水上敬介）

2.2　課題達成の為の班分け

前年度のプロジェクトでは前期で楕円曲線法についての学習を行い，後期でアルゴリズムの提案・実装を行っていた．しかし，このような日程でプロジェクトを進行していくと以下のような問題が発生した．

- 実際にプログラムを実装する期間が少ない
- 完成したプログラムを試行する期間が少ない
- 巨大な合成数の分解を行いにくい

本プロジェクトでは，5 月上旬まで全員で楕円曲線法についての基礎学習を行った．5 月中旬以降，既存のプログラムを改善するためにグループ全員でアルゴリズムの理解と実装をするのは効率が悪いと判断し，プログラム班，理論班の 2 つのグループに班分けをした．以下にそれぞれの班の課題を述べる．

理論班

ECM について理解を深め，Stage2 での新たなアルゴリズムの発見，理解に取り組んだ．理解したアルゴリズムをプログラム班に説明した．

プログラミング班

既存のプログラムの内容を理解し，プログラムの動作内容の理解に取り組んだ．理論班から提示されたアルゴリズムをもとに，既存のプログラムに実装を行った．

（※文責: 水上敬介）

第 3 章　活動内容

プロジェクトが発足した当初，楕円曲線についての基礎知識がなかったため，去年のプロジェクトでも基礎知識を身につけるために使われた資料を用いて，理解した．理解できないところはプロジェクトリーダーが主体となって解説をしてもらい楕円曲線についての基礎知識を学んだ．その後，アルゴリズムの提案をする理論班，提案されたアルゴリズムをプログラムに実装するプログラミング班に分けてプロジェクトを進行した．

（※文責: 水上敬介）

3.1　基礎学習

去年のプログラムを理解するために 5 月の中頃までメンバー全員が楕円曲線法のアルゴリズムや基礎知識についての学習を行った．具体的な内容は以下の通りである．

有限体

素数 p に対し，0 から $p-1$ までの整数の集合 $\mathbb{F}_p = \{0, 1, \cdots, p-1\}$ を有限体と言う．$\mathbb{F}_p$ では四則演算が可能であり，ECM ではこの範囲で考える．

Euclid の互除法

自然数 a，$b(a \geqq b)$ に対して以下の操作を繰り返し行い，余りが 0 になるまで行うことによって a，b の最大公約数を求めるものである．

Algorithm 1 Euclidean Algorithm

Require: $a, b \in \mathbb{N}, \quad a, b \neq 0, \quad a \geq b$

Ensure: $\gcd(a, b)$

while $b \neq 0$ **do**

$q \leftarrow a/b$

$r \leftarrow a \mod b$

$a \leftarrow b$

$b \leftarrow r$

end while

以後，a, b の最大公約数を $\gcd(a, b)$ と表記することとする．

拡張 Euclid の互除法

与えられた整数 a，b，c に対し，未知数 x，y に関する一次方程式 $ax + by = c$ の整数解は，1 組存在すれば無数に存在する．この方程式を一次不定方程式という．一次不定方程式の解を求めるには，拡張 Euclid の互除法が有効である．拡張 $Euclid$ の互除法は，自然数 a，b に関する一次不定方程式 ax+by=gcd(a，b) を満たす無数の整数 x，y を効率よく求めることができるというものである．例として 174x+69y=3 の整数解を求める．まず $Euclid$ の

互除法を用いて 174 と 69 の最大公約数を求める．

$$\begin{aligned} 174/69 &= 2 * 69 + 36 \\ 69/36 &= 1 * 36 + 33 \\ 36/33 &= 1 * 33 + 3 \\ 33/3 &= 11 \end{aligned}$$

となる．そしてこれらは

$$\begin{aligned} 3 &= 36 - 33 * 1 \\ 33 &= 69 - 36 * 1 \\ 36 &= 174 - 69 * 2 \end{aligned}$$

と表せるので

$$\begin{aligned} 3 &= 36 - 33 * 1 \\ &= 36 - (69 - 36 * 1) * 1 \\ &= 36 * 2 + 69 * (-1) \\ &= (174 - 69 * 2) * 2 + 69 * (-1) \\ &= 174 * 2 + 69 * (-5) \end{aligned}$$

以上より,$174x + 69y = 3$ の整数解は $(x, y) = (2, -5)$ と求めることができる．有限体 $\mathbb{F}_p$ において除算 $a \div b$ を計算する場合, p と b は互いに素なので, 拡張 Euclid の互除法により不定方程式 $px+by = 1$ の解 (x_0, y_0) を求めることができる．このとき $px_0+by_0 = 1$ となるので, 有限体 $\mathbb{F}_p$ 上では $by = 1$，つまり $b^{-1} = y_0$ が成立する．よって $a \div b = a \times b^{-1} = a \times y_0$ と変形することで, 除算を乗算に置き換えて計算できる．

楕円曲線の定義方程式

$a, b \in \mathbb{F}_p$に対して $y^2 = x^3 + ax + b$ で定義される曲線を素体 Fp 上の楕円曲線という．

楕円曲線の加算と逆元

(加算) 楕円曲線上のある 2 点 P, Q を通る直線をとすると, 楕円曲線と直線 ℓの 3 つ目の交点 R' $(= P \times Q)$ の x 軸に関する対称点を R とする．これで得られた点 R を R'の逆元と呼び，$R = -R'$が成り立つ．また，R を P と Q を加算した点と定義し，$R = P + Q$ が成り立つ．

無限遠点

楕円曲線上の点 P とその逆元 $-P$ をとり，$P + (-P)$ を考える．そうすると，2 点を通る直線と楕円曲線には P と $-P$ の他には交点が存在しない．このような状態のときに，存在しない点を仮想的に考え，それを無限遠点と呼び，$P + (-P) = O$ が成り立つ．

楕円曲線の２倍算

楕円曲線上の点 P の接線を ℓとし，楕円曲線と直線 ℓの P 以外の交点を R'とし，R'の逆元を R とする．この R は $R = P + P = 2P$ であり．楕円曲線の 2 倍算と定義する．

楕円曲線のスカラー倍算

点 P と自然数 d に対して，点 P を d 倍 ($dP = P + P + P + P +・・・・+ P$($d$ 個の和)) することを，楕円曲線のスカラー倍算という．

以上のことを基礎学習として学んだ．以下の章では，２つの班のそれぞれの活動内容を記述する．

（※文責: 落合航平）

3.2 理論班

理論班では，新しいアルゴリズムを探し，プログラム班に提案した．以下は具体的内容である．

3.2.1 ECM の原理の理解

まず，私たちは，ECM を用いた素因数を求めるプロセスについて，論文を読むことで学習した．

合成数 N について，その素因数 p を見つけることを考える．十分大きな自然数 n，楕円曲線上のある点 P に対して，mod$N$ で $nP = (\frac{X}{Z}, \frac{Y}{Z})$ を計算する．$gcd(Z, N)$ により，p が確率的に求まる．（もしも，$Z \equiv 0 \pmod{p}$ ならば，成功する）

ECM の Stage1 では，楕円曲線上のある点 P，ある適切な自然数 $B1$ に対して，2〜$B1$ までの最小公倍数 L が存在する．mod$N$ で，$Q_0 = LP$ を計算する．ここで，$LP = O \pmod{p}$ が成立するとき，素因数が求まる．

ECM の Stage2 では，ある適切な自然数 $B2$ に対して，$B1 < s < B2$ を満たす全ての素数 $s_1, s_2, ..., s_n$ について，それぞれ $s_i Q_0$ $(i = 1, 2, ..., n)$ を計算する．$sQ_0 = O \pmod{p}$ を満たす s が存在するとき，素因数が求まる．

（※文責: 金子真澄）

3.2.2 Baby-step Giant-step 法の理解

Stage2 の高速化を図るため，基礎学習で学習した Baby-step Giant-step 法をプログラムに導入しようと考えた．その際，基礎学習だけでは Baby-step Giant-step 法の理解が不足していたため，他の文献を探した．担当教員が紹介してくださった”Implementing the Elliptic Curve Method of Factoring in Reconfigurable Hardware”[3] という論文を深く読むことにした．

$0 < a < B2$ を満たすある自然数 a，$B1 < s \leq B2$ を満たす素数 s に対して，適当な自然数 v，u を用いて s は次のように表すことができる．

$$s = av \pm u \quad (0 < u < a)$$

Stage2 が成功するときの条件 $sQ_0 = O(\mathrm{mod}p)$ は，上式を用いて次のように変形できる．

$$sQ_0 = O(\mathrm{mod}p)$$
$$\Longleftrightarrow (av \pm u)Q_0 = O(\mathrm{mod}p)$$
$$\Longleftrightarrow avQ_0 = \pm uQ_0$$

Algorithm 2 Baby-step Giant-step 法

Require: N : 合成数, E : 楕円曲線, $Q_0 = kP_0$: $Stage1$ の結果, a, $B1$, $B2$: ある適当な自然数

```
for each i = 1 to a − 1 do
    H[i] ← i ∗ Q_0
end for
Q ← aQ_0
d ← 1
for each prime s = B1 to B2 do
    u ← s/a
    v ← s%a
    G ← v ∗ Q
    d ← d ∗ (G_x − H[i]_x)
end for
q ← gcd(d, N)
if q > 1 then
    return q
else
    return FAIL
end if
```

さらに，私たちは，a の値を小さな素数の積とすること，配列 H の要素数を減らすことを提案した．もしも，$s = av + u$ が素数であるならば，$gcd(a, u) = 1$ が成り立つ．このため，配列 H に $gcd(a, u) = 1$ のときのみ uQ_0 を保存するとよい．加えて，a が小さな素因数の積とすると，a と u が共通の因数を多く持つため，uQ_0 を計算する回数が少なくなる．PARI/GP で実装し，理論の理解を深め，プログラム班に伝えることができた．

（※文責: 金子真澄）

3.2.3 Montgomery curve の導入の検討

私たちは，昨年度のアルゴリズムを理解するため，Weierstrass 型以外の形式の楕円曲線について調べ，その中で Montgomery curve を発見した．Montgomery curve は加算の一回あたりの計算量が小さいため，これを導入することで Stage1 の高速化が可能かどうか検討した．

Montgomery curve は，方程式

$$By^2 = x^3 + Ax^2 + x \quad s.t.\ B(A^2 - 4) \neq 0$$

で定義され，Weierstrass 型と同値な曲線である．

（※文責: 金子真澄）

3.3 プログラミング班

プログラム班では，昨年度の FUN-ECM プロジェクトで作成したプログラムを使用し，単位時間あたりの計算量を増幅させるために，プログラムの改善を行った．高速化するために，Stage2 での Baby-step Giant-step 法の実装，去年のソースコードの不具合の修正を行った．また，昨年度のプログラムと今年度のプログラムの速度を比較するために，計測方法を確立させた．具体的には以下の通りである．

（※文責: 水上敬介）

3.3.1 座標変換の際の冗長なコストの削減

前年度のプロジェクトで作成された ECM プログラムでは，スカラー倍をする際の座標を射影座標から拡張射影座標に変換していた．スカラー倍を行う処理は ECM プログラムを実行する際に何度も使われるので，処理速度に影響する．そのため，最初から拡張射影座標を用意することによって座標変換する分のコストを減らすことが出来た．射影座標から拡張射影座標に変換する際に使われたアルゴリズムを Algorithm 3 に示す．

Algorithm 3 Projective Coordinates to Extended Coordinates

Require: (PX,PY,PZ) is Projective, (EX,EY,ET,EZ) is Extended, $N \geq 2$

Ensure: (EX,EY,ET,EZ)

$EX \leftarrow PX \times PZ$
$EX \leftarrow EX \bmod N$
$EY \leftarrow PY \times PZ$
$EY \leftarrow EY \bmod N$
$ET \leftarrow PX \times PY$
$ET \leftarrow ET \bmod N$
$EZ \leftarrow PZ \times PZ$
$EZ \leftarrow EZ \bmod N$

Algorithm 3 では，乗算を 4 回と mod の計算を 4 回行っている．プログラミング班では，スカラー倍を行う前に行われていた Algorithm 3 を省略することによって，計算コストを削減することが出来た．

（※文責: 福永慧）

3.3.2 ソースコードの改善

scalar 関数における引数の修正

前年度のプロジェクトではスカラー倍を行うときに座標Ｐのみを用意して，座標Ｐをスカラー倍したものを座標Ｐに代入していた．そのため，スカラー倍の処理を複数回行った場合，スカラー倍された座標にスカラー倍を行っていたことが分かった．しかし，スカラー倍を複数回行った場合でも，1 回分のスカラー倍の座標が必要であった．この問題はスカラー倍が行われる座標Ｐの他に，

その結果を代入するための座標を用意することで解決した．

Extended dedicated add 関数の修正

前年度のプロジェクトで作成された ECM プログラムでは，拡張射影座標を用いて加算をしていた．ECM プログラムで加算を行う関数の名前を Extended dedicated add と名付けていた．しかし，関数内の記述に書き損じがあり，正しく加算が出来ていなかった．今年度のプロジェクトではその記述を発見し，修正した．

main 関数における並列処理部分の変更

ECM プログラムでは，OpenMP を導入して並列処理が行なえるようにしていた．並列処理は main 関数内にある for 文に用いられていた．

今年度のプロジェクトでは main 関数内にある for 文の冗長性を発見し，ソースコードの修正を行った．前年度のプロジェクトで記述されていた処理を Listing 3.1 に記す．

Listing 3.1　前年度の並列処理

```
1  int found=0;
2  for(i=0;i<number_of_elliptic_curves;i++) {
3          mpz_t factor;
4          mpz_init(factor);
5          if(found == 0){
6                  ecm(factor, N, X, Y, d, B1, B2, fp, window_size );
7                  if(mpz_cmp_ui(factor, 1) != 0 && mpz_cmp_ui(factor, N) != 0)
8                          found = 1;
9          }
10 }
```

Listing 3.1 の概要を以下に記す．

- 関数 ecm は与えられた合成数Nの素因数分解を行う関数である．関数 ecm では，素因数分解された素数を引数 factor へ返す．
- 関数 ecm で素因数分解された素数が 1 かつ合成数 N でない場合，found に 1 を代入する．
- found が 1 の時，for 文の処理が終わるまで ecm 関数を計算しない．

今年度のプロジェクトでは，このソースコードの改善点を 2 つ見つけた．1 つ目に引数 found が 1 でも，ループから抜け出せていない点があった．2 つ目にこの for 文は並列処理で行われているので，他のスレッドの処理が終わるまでプログラムの処理が終わらないという点があった．今年度のプロジェクトではこれらの 2 点を改善するために，Listing 3.1 の 9 行目と 10 行目の間に exit 関数を用いた．また，found が 1 の時のみ exit 関数を呼び出すように記述した．

（※文責: 福永慧）

3.3.3　計算方法の確立

FUN-ECM のプログラムは桁数が大きくなるにつれ実行時間が指数関数的に増大していく．プログラムの実行を人間が手で行うことは非効率であるため，テスト用のスクリプトを作成した．

スクリプトは入力として一行に以下の内容を含む CSV ファイルを受け取る．

桁数，素数 A，素数 B，合成数 (A × B)，B1，B2

スクリプトは各行に記載されている合成数に対して指定された B1,B2 を用いて，今年度と昨年度の FUN-ECM のプログラムを実行する．すべての行の処理が終了すると，実行結果の出力ファイルを解析し，各桁の平均を算出しファイルに保存する．なお，スクリプトの実行には nohup コマンドを用いることでログアウト後も処理を継続することができる．

今回は実行時間を測定するためにスクリプトを使用した．20 桁から 50 桁までの 5 桁刻みで各桁 7 個の合成数を対象とした．すべての実行には 503257.096[秒] かかり，これは 5.8[日] かかった計算となる．この間スクリプトは正常に動作し，自動的に実行を行うことができた．

（※文責: 外山拓）

3.4 中間発表

3.4.1 準備

ポスター

初めに，前年度のプロジェクトで作成されたポスターを参考に構成を決定した．次に，概要，基礎学習，理論班，プログラミング班の 4 つの項目に分け，作成を分担した．ポスターの作成には「Microsoft PowerPoint」というソフトウェアを使用した．ポスターが完成次第，理論班・プログラミング班でレビューを行い，誤字脱字やフォントの違い等を修正した．しかし，中間発表当日に誤字が発見された．

（※文責: 小澤貴也）

プレゼンテーション資料

本プロジェクトの内容を説明するのにポスターだけでは不十分と判断しプレゼンテーション資料を作成することにした．理論班，プログラム班から一人ずつメインに作成する人を決め，他メンバーと話し合いながら，発表内容を考えた．プレゼンテーション資料の作成には「Microsoft PowerPoint」を使用し，共有に「SharePoint Online for Office 365」を使用した．また，先生から，内容やデザインの指摘を受け，改良することで，どんな人でも理解できるように努めた．

（※文責: 金子真澄）

原稿

前述のプレゼンテーション資料の作成と並行して，発表用の原稿の作成を行い，グループ内で担当を決め各ページの原稿を作成した．楕円曲線方法を理解してもらうようになるため，基礎学習の部分を最低限の知識だけを伝えるように作成した．作成した原稿は，担当教員に確認していただき，伝わりにくい表現の修正を行った．

（※文責: 水上敬介）

3.4.2　発表

発表はスライドの説明をメインとし，ポスターにはスライドの内容をより詳しくしたものを用意した．中間発表会での発表を行い，その結果，スライドに数式が多く，発表時間の短さも相まって，数学・楕円曲線法をよく知らない人にとって分かりにくい発表になってしまっていた．発表の反省を行った結果，後期末の発表では数学に興味のない人でも概観が分かるようにすることに決めた．具体的には，スライドでは極力数式を出さずに説明し，興味を持った人に対して，ポスターを用いて専門的な説明をする，という形式をとることにした．

（※文責: 中島俊平）

第 4 章　プロジェクト内のインターワーキング

- 中島俊平（プロジェクトリーダー・理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 大まかな作業スケジュールを作成し，進捗管理を行った.
  (3) ”Implementing the Elliptic Curve Method of Factoring in Reconfigurable Hardware” を落合，金子と協力して読解した.
  (4) PARI/GP で Baby-step Giant-step 法のプログラムを作成し，メンバーに共有することで，理解を深めた.
  (5) 中間発表会に向けて，進捗管理を行った.
  (6) 中間発表会に向けて，「理論班」の部分のポスターの原案を作成した.
- 金子真澄（理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) ”Implementing the Elliptic Curve Method of Factoring in Reconfigurable Hardware” を落合，中島と協力して読解した.
  (3) 落合，中島と協力して，PARI/GP で ECM を実装した.
  (4) 落合，中島と協力して，Baby-step Giant-step 法のアルゴリズムをプログラム班に教えた.
  (5) 中間発表に向けて，「理論班」の部分のプレゼンテーション資料を作成した.
- 落合航平（理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) ”Implementing the Elliptic Curve Method of Factoring in Reconfigurable Hardware” を金子，中島と協力して読解した.
  (3) 金子，中島と協力して，Baby-step Giant-step 法のアルゴリズムをプログラム班に教えた.
  (4) 金子，中島と協力して，PARI/GP で ECM を実装した.
  (5) 中間発表会に向けて，ポスターの「概要」の部分についてポスターを作成した.
- 広瀬大樹（理論班）
  (1) 楕円曲線法の基礎を学んだ.
- 水上敬介（プログラム班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) プログラム班と協力して昨年度のプログラムの不具合の修正を行った.
  (3) プログラム班でのスケジュール管理を行った.
  (4) プログラム班での進行役を務め，課題に対してのメンバーの役割を決めた.
  (5) 理論班から PARI/GP を用いて実装されたアルゴリズムをプログラムに実装した.
  (6) 中間発表に向けてのスライド資料・プログラム班の原稿の原案を作成した.
- 福永慧（プログラム班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) Baby-step Giant-step 法の文献を読み，原理を理解した.

(3) 昨年度の FUN-ECM で作られたプログラムを理解した.
(4) Baby-step Giant-step 法のアルゴリズムを実装した.
(5) 中間発表に向けて，評価アンケートの作成をした.
(6) 中間発表に向けて，ポスターの英訳を行った.

- 小澤貴也（プログラム班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 昨年度の ECM プログラムを理解した.
  (3) 昨年度のプログラムの修正をした.
  (4) Baby-step Giant-step 法のアルゴリズムを実装した.
  (5) 中間発表に向けて，ポスターを作成した.
- 外山拓（プログラム班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) Baby-step Giant-step 法の原理を学んだ.
  (3) Baby-step Giant-step 法の実装を行った.
  (4) プログラムを実行するサーバの管理を行った.
  (5) 昨年度の FUN-ECM のプログラムを理解した.
  (6) 昨年度のプログラムの並列処理を高速化した.

（※文責: 中島俊平）

第 5 章　活動成果

本プロジェクトでは，前期はメンバー全員で楕円曲線法の学習から初め，楕円曲線法の概要を理解した．その後は理論班とプログラム班に分かれた．

（※文責: 外山拓）

5.1　理論班

理論班の成果として，まず Baby-step Giant-step 法の理解がある．基礎学習を終えた後，理論班は Baby-step Giant-step 法について勉強した．その後，理論班のメンバーそれぞれがそれに関する論文を探し，読むことで Baby-step Giant-step 法をより深く理解することができた．そして，その知識を生かし，理論班で最適なアルゴリズムを考え，プログラム班に伝えることができた．また，もう一つの成果として，昨年度のプログラムに誤りがあったため，改良すべき点としてプログラム班に提案をしたことが挙げられる．

（※文責: 落合航平）

5.2　プログラミング班

プログラム班では，Stage2 において Baby-step Giant-step 法を実装した．確立した計測方法を実証したところ，昨年度のプログラムより Stage2 の精度が向上し，高速化していることが確認できた．実証した結果は図 5.1，図 5.2 のようになった．

図 5.1，図 5.2 の結果から，Stage2 においての計算速度が 6〜7 倍向上したことが分かり，プログラム全体の速度の大幅な改善が確認された．

（※文責: 水上敬介）

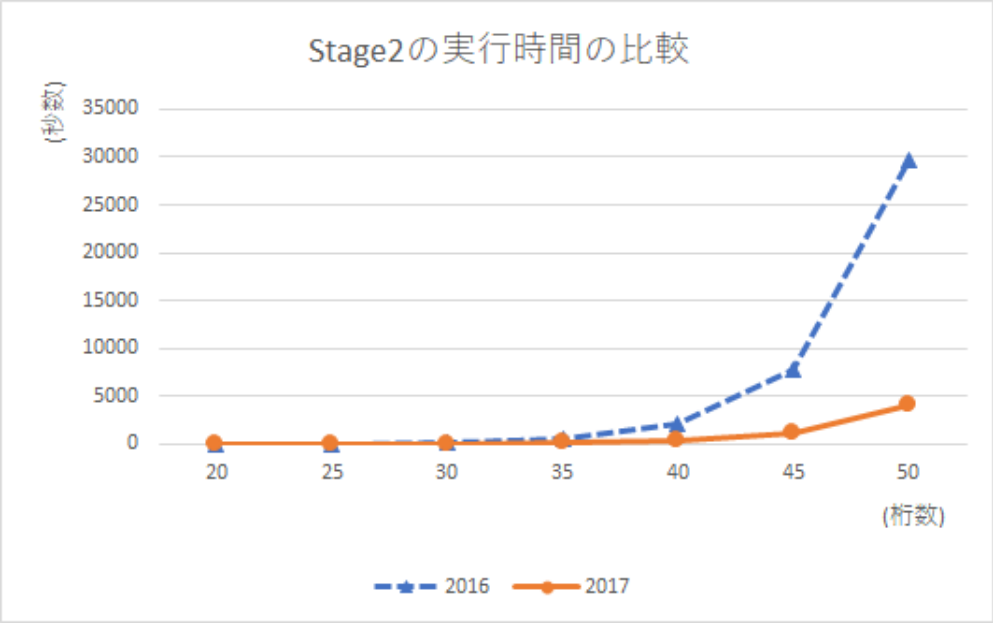

図 5.1　Stage2 の実行にかかった時間

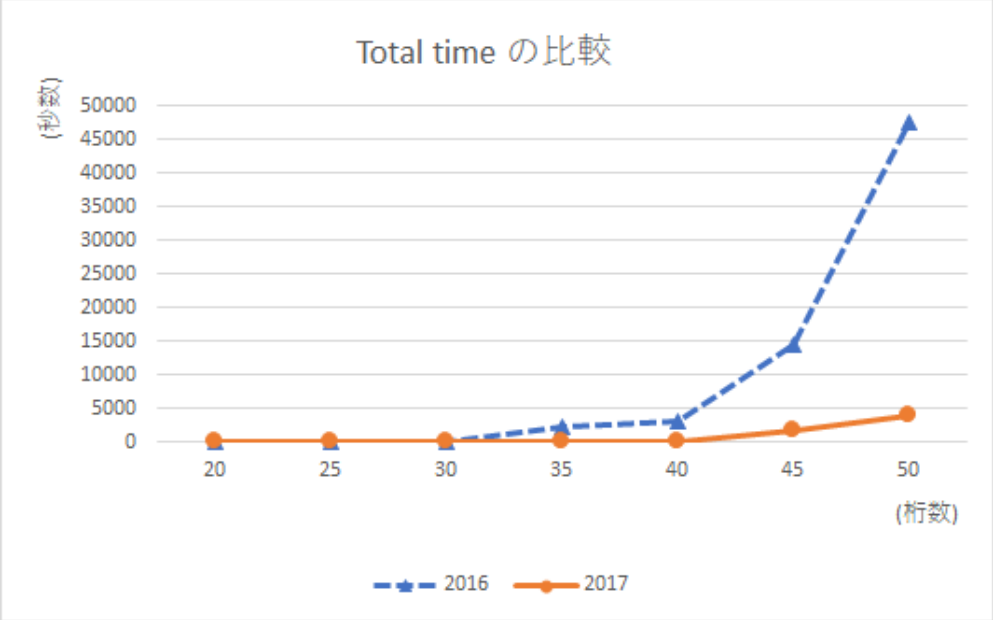

図 5.2　プログラム全体の実行にかかった時間

第 6 章　まとめ

6.1　前期活動結果

前期はメンバー全員で楕円曲線法の学習から初め，楕円曲線法の概要を理解した．その後は理論班とプログラム班に分かれた．

理論班は Baby-step Giant-step 法について理解し，その内容をプログラム班に教えた．また，論文を探して効率の良い曲線についての考察を行った．

プログラム班は昨年度のプログラムの解読を行い，省メモリ化，高速化を行なった．また，理論班の説明を受け Baby-step Giant-step 法のアルゴリズムの実装を行った．

（※文責: 外山拓）

6.2　後期の展望

後期の活動として，理論班は，Baby-step Giant-step 法の素数ペアリングの理解，符号付き 2 進展開や Rucus chain を用いたスカラー倍算の理解などが挙げられる．プログラム班は，素数テーブルからの読み出しやスカラー倍算の並列化による高速化の確認が挙げられる．以上の活動を行い，ECMNET，STUDIO KAMADA へのランクインを目指す．

（※文責: 中島俊平）

参考文献

[1] ECMNET. https://members.loria.fr/PZimmermann/ecmnet/, (最終アクセス 2017 年 7 月 20 日)

[2] STUDIO KAMADA. http://stdkmd.com/, （最終アクセス 2017 年 7 月 20 日）

[3] Kris Gaj, Soonhak Kwon, Patrick Baier, Paul Kohlbrenner, Hoang Le, Mohammed Khaleeluddin, Ramakrishna Bachimanchi. Implementing the Elliptic Curve Method of Factoring in Reconfigurable Hardware. Cryptographic Hardware and Embedded Systems - CHES 2006, 2006

[4] 楕円曲線法（ECM）faireal.net. http://www.faireal.net/articles/6/07/,（最終アクセス 2017 年 7 月 20 日）.