公立はこだて未来大学 2016 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2016 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.

15-A

プロジェクトリーダ/Project Leader

1014129　池野竜將　Ryusuke Ikeno

グループリーダ/Group Leader

1014129　池野竜將　Ryusuke Ikeno

グループメンバ/Group Member

1014068　駒ヶ嶺壮　Sou Komagamine
1014109　伊藤有輝　Yuki Ito
1014129　池野竜將　Ryusuke Ikeno
1014137　千葉大樹　Daiju Chiba
1014164　橋本和典　Kazunori Hashimoto
1014168　山下哲平　Teppei Yamashita
1014209　源啓多　Keita Minamoto
1013150　亀谷浩也　Hiroya Kametani

指導教員

白勢政明　由良文孝

Advisor

Masaaki Shirase　Fumitaka Yura

提出日

2016 年 7 月 27 日

Date of Submission

July 27, 2016

概要

私たち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．素因数分解は，約 30 年前から重要になってきている．その理由は，RSA 暗号にある．RSA 暗号は，安全性を 2 つの大きな素因数からなる合成数の素因数分解が難しいことに依存している．しかし，技術の発展とともに素因数分解が従来よりも容易になってきてしまっているため，RSA 暗号が破られる可能性が高くなっている．そこで今注目されているのが楕円曲線暗号である．楕円曲線暗号は，RSA 暗号と同じ鍵長で高い安全性を保障することができる．そこで私たちは素因数分解をより簡単なものとすることで，RSA 暗号から楕円曲線暗号を主流とさせたい．

私たちは，大きな素因数の発見のために，色々な文献を読んでその中から素因数分解を行うプログラムの改良法を発見する理論班と，それらの理論を利用して実際にプログラムの実装・改良を行い，プログラムを高速化させるプログラム班に分かれて活動を行った．

理論班は，素因数分解がより高速に行われるようなアルゴリズムの発見を目標とした．楕円曲線法のプログラムは点の加算の繰り返しで行われるため，加算の計算コストを減らすことで高速な計算を可能とするための活動を行った．Atkin Morain ECPP を利用することで，従来の楕円曲線法よりも計算コストを削減できることを発見した．

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．前年度と同様に大きな数を扱うために，任意精度演算ライブラリの GMP を使用した．また，プログラムの並列実行を行うために，並列プログラムの為の API である OpenMP を導入した．前年度に実装されたエドワーズ曲線よりも効率よく計算を行うため，extended twisted Edwards coordinates を実装した．同じ合成数に対してプログラムを実行する際の因数の発見確率をあげるために，Y の値をランダムに設定した．

また，理論班とプログラム班で情報の交換を行ったり，協力を行ったりなど，2 つの班の活動により，素因数分解を高速に行うことができ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, 射影座標,RSA 暗号

（※文責: 山下哲平）

Abstract

The goal of our project team is to find prime factor as large as possible. Factorizations in prime numbers have become more important since about thirteen years ago because the difficulty of factorization in prime numbers is related to Internet security. The reason lies in the RSA. The asymmetry of RSA is based on the practical difficulty of factoring the product of two large prime numbers. However, prime factorization is getting to easier by development in technology. Therefore, RSA is less secure compared to previously. That's why Elliptic Curve Cryptography (ECC) is paid more attention than RSA now. ECC ensure safety better than RSA cryptosystem with same key length.Accordingly, we make factorization in prime numbers simplify, we would like to change main cryptosystem from RSA cryptosystem to elliptic curve cryptography.

In order to find prime factor as large as possible, we divided two groups, one is "theory group" that is to read various literature and to find algorithm of factorizations in prime numbers to calculate faster, the other is "programming group" that is to make program to base on algorithm.

"Theory group" aims to find algorithm of factorizations in prime numbers to calculate faster. The ECM program repeats process of addition law many times over, therefore we reduced calculation. To access Atkin Morain construction, we were successful in calculation faster compared to previously.

"Programming group" aim to improve program of last year project team faster than before. To treat large number likewise last year, we used arbitrary-precision arithmetic library called GMP. Also, we parallelize the program, we introduce Open MP is API for parallel program. We implement extended twisted Edwards coordinates efficient than Edwards curve implemented last year. Also, we set random Y' s value to raise finding assembly towards same composite numbers. We exchange information and cooperate "theory group" and "programming group", we made a program that is to perform factorization in prime numbers fast.

Keyword Elliptic Curve Method, prime factorization, ECMNET, Twisted Edwards Curve, Extended Twisted Edwards Coordinates, RSA cryptosystem

（※文責: 山下哲平）

目次

第 1 章　背景

ECM(楕円曲線法) を利用した素因数分解は近年重要になっており，それを利用し ECM-NET にランクインすることが私たちの目的である．

（※文責: 駒ヶ嶺壮）

1.1　本プロジェクトの背景

現在インターネットを含む通信での暗号技術においての主流は RSA 暗号である．RSA 暗号とは公開鍵暗号の一つで，大きな合成数を素因数分解することの難しさを安全性の根拠にした暗号である．しかし，スーパーコンピューターの並行処理能力と計算能力の向上等で鍵長 1024 ビットの RSA 暗号方式は解読される危険性が指摘されるようになった．ここで，今後の暗号技術には RSA に変わるものとして楕円曲線暗号が使われて始めている．楕円曲線暗号は現在の暗号技術において最も重要とされている手法である．これは，ある楕円曲線における有限体上の楕円曲線の点の加算を用いることにより，RSA 暗号と同じ鍵長でより解読が難しくなるからである．ここで私たちはこの楕円曲線暗号の中で核となる楕円曲線を用いた素因数分解のアルゴリズムについて考え，FUN-ECM が ECM-NET にランクインを目指すことで函館から楕円曲線，素因数分解，暗号技術の重要性について発信することを目標として掲げた．

（※文責: 山下哲平）

1.2　ECM-NET とは

ECM-NET とは，楕円曲線法を用いて大きい桁数の素因数を見つけることを目的とした競争サイトである．ECM-NET には現在登録されている素因数分解よりも大きな素因数を見つけることで誰でもランクインすることが可能である．

（※文責: 駒ヶ嶺壮）

1.3　課題の概要

FUN-ECM が ECM-NET へのランクインを目指すには大きい桁数の素因数を見つけなければいけないことから楕円曲線を用いた素因数分解のプログラムの並列処理と高速化を目指す．また，本プロジェクトの活動を Web サイト等を用いて外部に発信する．

（※文責: 駒ヶ嶺壮）

第 2 章　到達目標

2.1　本プロジェクトにおける目的

FUN-ECM が ECM-NET にランクインするためには去年のプログラムをより改善する必要がある．この目標を達成するにあたって，2 つの目標を立てることにした．

（※文責: 伊藤有輝）

2.1.1　プログラムの高速化

ECM-NET にランクインするためには，巨大な素因数を発見しなければならない．巨大な素因数を発見するためには桁数の大きい合成数を素因数分解する必要があるが，それには多大な時間がかかってしまう．また，ECM は 1 度の試行で素因数を必ず発見できるとは限らず，複数回の試行が必要となる．そのためプログラムの処理を効率の良いアルゴリズムに変更し，処理を高速化させる必要がある．この目標を達成するにあたって，2 つの目標を立てることした．

- 昨年度のプログラムのアルゴリズムの理解
- 昨年度のプログラムの書き換えたものの実装

まず，昨年度のプログラムを高速化するにはアルゴリズムの理解が必要である．また，楕円曲線法では大学までの学習で使用していない数学の概念を使用するため，基礎学習を行う．

（※文責: 伊藤有輝）

2.1.2　FUN-ECM の活動発信

今年度では，ただランクインを目指すだけでなく，函館から楕円曲線，素因数分解の重要性について発信することに決め，ホームページを設立することとした．

（※文責: 伊藤有輝）

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

そのため，本プロジェクトでは 5 月中旬まで全員で最低限の基礎学習を行い，そこから理論班とプログラミング班の 2 つに分けて作業を行うこととした．また，後期には広報班を作成し，3 つの作業を並行で行うこととした．以下にそれぞれの班の課題について述べる．

理論班

ECM について理解を深め，高速化の新たなアルゴリズムを提案する．

プログラミング班

基礎学習や理論班がまとめたアルゴリズムを実装し高速化を行う．

広報班

ECM について理解してもらえるような Web ページの作成をする．

（※文責: 伊藤有輝）

第 3 章　前期活動内容

プロジェクトが始まった当初，ほぼ全員楕円曲線についての前提知識がなかったため，昨年も前提知識を身に着けるために使われた全員楕円曲線についての資料を全員で輪読し，理解した．その際，理解できなかったところを由良先生，白勢先生に解説してもらった．それにより，楕円曲線法のアルゴリズムを理解するためにあたっての基礎知識を学んだ．その後，プロジェクト全体をプログラムの高速化につながる理論を探し，学習してアルゴリズムをノートにまとめる理論班，理論班がノートにまとめ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めた．

（※文責: 伊藤有輝）

3.1　基礎学習

去年のプログラムを理解するために 5 月の中旬まではメンバ全員が教授の指導のもとで楕円曲線法のアルゴリズムや基礎知識ついての基礎学習を行った．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余りが 0 になるまで行うことによって a，b の最小公倍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

a, b の最大公約数を $\gcd(a, b)$ と表記できる．

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 $ax + by = c$ の整数解を求める問題を一次不定方程式という．ここで，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 は拡張Euclid の互除法を用いることで効率よく求めることができる．これは Euclid の互除法で行った操作を逆に行うことで解を得る．$gcd(174,\ 69) = 3$ を例にとって考える．

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

となるので

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

以上より,$174x + 69y = 3$ の解 $(x, y) = (2, -5)$ を得ることができる。有限体 $\mathbb{F}_p$ において除算 a/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, y) を求めることができる。このとき $px + by = 1$ となるので, 有限体 $\mathbb{F}_p$ 上では $by = 1$ となり,両辺を b で割ることで,$b^{-}1 = y$ が成立する。したがって $a \div b = a \times b - 1 = a \times y$ と変形することで, 除算を乗算に置き換えて計算できる。プログラムにおいて、除算を乗算に置き換えることは計算量の削減につながるが、今回のプロジェクトでは GMP ライブラリを用いたことでこれを実装することはなかった。

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・2 倍算

(加算) 楕円曲線上のある 2 点 P, Q を通る直線をとると,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。このとき 2 点 P, Q の和を $R = P + Q$ と定義し、楕円曲線の加算という。

(2 倍算) 楕円曲線上の 1 点 P で加算を考えるときは 2 点 P, P の通る直線 $(= P$ の接線) をとして考える。この時、楕円曲線と直線 ℓの P 以外の交点の x 軸に関する対称点を R としたとき、$R = P + P = 2P$ とできる。これが楕円曲線の 2 倍算である。

楕円曲線のスカラー倍

点 P と整数 m を使用して，$mP = P + P + P + P +$・・・・$+ P(m$ 個の和) と表すことができる．これを楕円曲線のスカラー倍という.

楕円曲線法のアルゴリズム

N を素因数分解したい合成数とする．$\mathbb{Z}/N\mathbb{Z}$ 上で，楕円曲線 E を構成して，点

$$P \in E(\mathbb{Z}/N\mathbb{Z}) \tag{3.1}$$

をとる．初めに P の座標を決めてから E を構成しても良い.

次に適切な B_1，$L = 2, 3,$・・B_1の最小公倍数とする．LP の計算の過程で生じる点の座標の分母 d が$\gcd(N, d) \neq 1$ となると N の約数を発見できる.

最期まで$\gcd(N, d) = 1$ ならば，E と P を選びなおしてやり直す．適切な B_1を選ぶことで，ECM は高速な素因数分解法になることが知られている.

以上のことを基礎学習として学んだ．以下の章ではに 2 班に分かれた後の理論班の活動内容を記述する.

（※文責: 橋本和典）

3.2 理論班

理論班では新たなアルゴリズムを探し，プログラミング班に新たな高速化手法の提案を行った．以下に具体的な内容を述べる．

3.2.1 Twisted Edwards Curve の理解

ECM の高速化アルゴリズムを実装するにあたって，先人の知恵を得ようと思いインターネットで類似研究の論文を検索し，その論文を解読することによって高速化アルゴリズムをプログラムに導入しようと考えた．その際，Twisted Edwards Curves Revisited というエドワーズ曲線についての英語の論文が見つかったため，私たちはこの論文を読解することにした．

この論文は，最初に一般的な楕円曲線アルゴリズムより，エドワーズ曲線の方が計算コストは低く，速いスピードで素因数を求めることができるということが説明されており，そのエドワーズ曲線の数学的な理論とプログラム実装のためのアルゴリズムが書かれていた．

エドワーズ曲線については基礎学習で学んでいなかったため，私たちはエドワーズ座標を学習した．その中では射影座標が使用されていた．射影座標とは一般的なの座標 (x,y) に対して $x = \frac{X}{Z}, y = \frac{Y}{Z}$を満たす X, Y, Z を用いて（X, Y, Z）と表す座標であり，射影座標を用いるとECMアルゴリズムを高速化することができる．具体的な定義は以下の通りである．

射影座標

$$(X, Y, Z) = (\lambda X, \lambda Y, \lambda Z) = (\frac{X}{Z}, \frac{Y}{Z}, 1)\ (Z \neq 0)$$

理論班では，この拡張エドワーズ座標の理論を学ぼうとしたが，知識が乏しく，わからない変数が出てきたため，アルゴリズムだけを理解し，定義，証明などの理論を理解することはあきらめた．

（※文責: 伊藤有輝）

3.2.2 Atkin-Morain ECPP

次に Atkin-Morain ECPP という ECM の初期座標を決定するアルゴリズムの理解に励んだ. 昨年度までは ECM の初期座標として (2,2) を用いて, 素因数分解が完了できなければ (2,3),(2,4)… といったように Y 座標を 1 ずつ動かすようにしていたが, 今年度では少しでもを因数を見つける確率を上げることが見込める Atkin-Morain ECPP を理解することにした.Atkin-Morain ECPP では新たな楕円曲線 $T^2 = S^3 - 8S - 32$ の点を用意し, $(S, T) = (12, 40)$ に対して $n(S, T)$ の座標 (s, t) を用いて以下を定める.

$$\alpha = \frac{(s-9)+1}{t+25}, \beta = \frac{2\alpha(4\alpha+1)}{8\alpha^2-1} \tag{3.2}$$

これらを用いることによって、素因数分解に用いる楕円曲線の初期座標を求めることができる。具体的には以下の通りである。

Algorithm 2 Atkin-Morain ECPP Algorithm

Require: $\alpha, \beta, s, t, \in \mathbb{N}$

Ensure: (X, Y)

$(s, t) \leftarrow (12, 40)$

while Prime factor is not found **do**

$\alpha \leftarrow \dfrac{(s-9)+1}{t+25}$

$\beta \leftarrow \dfrac{2\alpha(4\alpha+1)}{8\alpha^2-1}$

$d \leftarrow \dfrac{2(2\beta-1)^2-1}{(2\beta-1)^4}$

$E : x^2 + y^2 = 1 + dx^2y^2$

$X \leftarrow \dfrac{(2\beta-1)(4\beta-3)}{6\beta-4}$

$Y \leftarrow \dfrac{(2\beta-1)(t^2+50t-2s^3+27s^2-104)}{(t+3s-2)(t+s+16)}$

Run ECM with $E : x^2 + y^2 = 1 + dx^2y^2$ and (X, Y)

$(s, t) \leftarrow 2(s, t)$

end while

このアルゴリズムを用いると具体的には従来の 1.5 倍ほど高速化できる見込みであるが、これは論文上のデータである。したがって、後期はプログラム班が実装し、どのくらい速くなるかどうかを検証したいと考えている。

（※文責: 伊藤有輝）

3.3 プログラミング班

プログラミング班では，昨年度の FUN-ECM プロジェクトで作成した ECM プログラムをさらに高速化するために，4 月から 5 月にかけて行った全体での基礎学習や，理論班がまとめた理論・アルゴリズムを元にプログラムを変更した．主に，射影座標や extended twisted Edwards coordinates を用いて乗算・除算を減らすことによって高速化を図った．また，前年度のプログラムの不具合等も改善した．具体的には以下の通りである．

（※文責: 源啓多）

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アフィン座標から射影座標に変換することで計算効率を上昇させていた．このアフィン座標から射影座標への変換は複数回呼び出される為，ECM プログラムの計算コストに影響する．Algorithm 3 にアルゴリズムを記す．

Algorithm 3 Affine Coordinates to Projective Coordinates (Past ver.)

Require: (AX, AY) is Affine, (PX, PY, PZ) is Projective, $N \geq 2$

Ensure: (PX, PY, PZ)

$Z \leftarrow Random(0 \leq Z < N)$

if $Z = 0$ **then**

$Z \leftarrow 1$

end if

$AX \leftarrow AX \times Z$

$AY \leftarrow AY \times Z$

$AX \leftarrow AX \mod N$

$AY \leftarrow AY \mod N$

$(PX, PY, PZ) \Leftarrow (AX, AY, Z)$

前述の冗長部分として乗算が 2 回と *mod* の計算が 2 回発生している．プログラミング班では，Z の値を 1 に設定することで乗算と *mod* の計算を省略できると考えた．プログラムを一通り読み直し，問題が発生しないことを確認したのち，新たなアルゴリズムを実装した．Algorithm 4 に新しいアルゴリズムを示す．

Algorithm 4 Affine Coordinates to Projective Coordinates (New ver.)

Require: (AX, AY) is Affine, (PX, PY, PZ) is Projective, $N \leq 2$

Ensure: (PX, PY, PZ)

$Z \leftarrow 1$

$(PX, PY, PZ) \Leftarrow (AX, AY, Z)$

（※文責: 源啓多）

3.3.2 Extended twisted Edwards coordinates の実装

前年度のプロジェクトで作成された ECM プログラムでは，twisted Edwards curve を利用している．今回のプログラミング班ではさらに extended twisted Edwards coordinates を用いた．extended twisted Edwards coordiantes はエドワーズ曲線のスカラー倍を高速化するための座標系であり，以下で定義される補助座標 T を加えた 4 つの座標でスカラー倍を行う．

Extended twisted Edwards coordinates

射影座標（X,Y,Z）をに対し，T=$\dfrac{XY}{Z}$ という補助座標を加える．これを Extended twisted Edwards coordinates と呼ぶ．

$$(X, Y, Z) \rightarrow (X, Y, T, Z)$$

（※文責: 源啓多）

3.3.3 楕円曲線の生成法の変更

楕円曲線法を利用した ECM プログラムは，楕円曲線を生成しその座標を利用し素因数分解を行うプログラムである．また，本プロジェクトで素因数分解しようと試みている合成数は 200 桁前後のため，1 度の試行では素因数分解できないことが多くある．よって，同じ合成数に対して複数回の試行をすることを想定してプログラムを作成する必要がある．前年度のプログラムでは，楕円曲線を生成する際に，Y 値を for 文のカウンタを利用して 1 から順に決めるアルゴリズムを採用していた．そのため，複数回試行した際に同じ曲線を使用してしまうことが多くあり，効率が落ちていたと仮定した．そこで曲線を生成する際に使用している Y 値に乱数を使用することとした．

（※文責: 源啓多）

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，誤字脱字等を修正した．しかしポスターレビューが不十分だったため，最終的に完成したポスターで誤植が見つかってしまった．

（※文責: 亀谷浩也）

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には，ポスターだけでは足りないと判断しプレゼンテーション資料を作成することに決定した．作成にあたって，まず 1 名がプレゼンテーションの大まかな流れを作成し，各自作成する章を分担した．プレゼンテーション資料の作成には「Microsoft PowerPoint」というソフトウェアを使用した．また，一度完成したプレゼンテーション資料を先生にレビューしていただき，資料中のグラフの不備や内容についての助言を受けた．それを受け，文章や図の修正を行った．これにより，より見やすいプレゼンテーション資料が完成した．

（※文責: 亀谷浩也）

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った．こちらも 1 名が大まかな流れを作成し，各自作成する章を分担した．特に楕円曲線法については，何も知らない聴衆でもわかりやすく説明できるように，専門的な用語を最小限にするように注意して作成した．何度か原稿とプレゼンテーション資料を使用しプレゼン練習を行い，伝わりにくい表現や冗長な表現を修正した．

（※文責: 亀谷浩也）

3.4.2 発表

発表は前後半で 4 人ずつに分かれ，発表を行った．それぞれが自分の担当する部分を読み上げ，その間他の 3 人は評価アンケート配布や，ポスターに関しての質問に対応した．発表途中にプロジェクターの電源が落ちてしまうというアクシデントがあったが，落ちている間は PC の画面を直接見せることでプレゼンを行い，他の 3 人で復旧作業を行った．発表後に評価アンケートの集計を行った結果、発表技術は 10 点中平均 7.1 点、発表内容は 10 点中 7.5 点だった。コメントでは内容を理解していた人と全く理解できない人が分かれていたため、さらに前提知識のない聴衆にも伝わるような内容にしていきたい。

（※文責: 亀谷浩也）

第 4 章　後期活動内容

4.1　理論班

4.1.1　プログラムの検証

4.2　プログラミング班

4.2.1　Atkin-Morain ECPP の実装

4.2.2　ECM における Stage2 の実装

4.3　広報班

4.3.1　ウェブページの作成

第 5 章　プロジェクト内のインターワーキング

- 池野竜將（プロジェクトリーダー・プログラミング班）
  (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) 源と協力して前年度の ECM プログラムを理解した.
  (4) 源のコーディング作業にアドバイスをした.
  (5) 理論班からのプログラミング班に関しての質問に回答し，必要があれば聞かれた内容を源に伝えた.
  (6) 中間発表会に向けて，プレゼンテーション資料・原稿の原案を作成した.
  (7) 中間発表会に向けて,「プログラミング班」の部分のプレゼンテーション資料を作成した.
- 源啓多（プログラミング班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 池野と協力して前年度の ECM プログラムを理解した.
  (3) ECM プログラムのバージョン管理の為，Git を学んだ.
  (4) 前年度の ECM プログラムの実装上のミス（3.3.1）を改善した.
  (5) ECM プログラム改善のために，新たなアルゴリズム（3.3.2, 3.3.3）の実装を行った.
  (6) 中間発表会に向けて，プログラミング班のプレゼンテーション資料・原稿を作成した.
- 山下哲平（理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 伊藤・駒ヶ嶺と協力して Edwards Curve を利用した ECM アルゴリズムの読解を行い，プログラミング班に提案を行った.
  (3) 伊藤・駒ヶ嶺と協力して Atkin-Morain ECPP アルゴリズムの理解に取り組んだ.
  (4) 中間発表会に向けて,「背景」の部分についてポスターをを作成した.
  (5) プログラミング班の要請でプログラムの速度について簡易的な検証を行った.
- 伊藤有輝（理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 駒ヶ嶺と協力して，エドワーズ曲線の式が導き出される過程を学んだ.
  (3) 駒ヶ嶺･山下と協力し, Edwards Curve を利用した ECM アルゴリズムの読解を行った.
  (4) 駒ヶ嶺・山下と協力し，Atkin-Morain ECPP アルゴリズムの理解に取り組み，プログラミング班に提案を行った.
  (5) 中間発表会に向けて,「理論班」の部分のプレゼンテーション資料を作成した.
- 駒ヶ嶺壮（理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 伊藤と協力して，エドワーズ曲線の式が導き出される過程を学んだ.
  (3) 山下・伊藤と協力して Edwards Curve を利用した ECM アルゴリズムの読解を行った.
  (4) 山下・伊藤と協力して Atkin-Morain ECPP アルゴリズムの理解に取り組んだ.
  (5) 中間発表会に向けて,「理論班」の部分のポスターを作成した.

- 橋本和典（理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 千葉・亀谷と協力して入門書を読み，基礎学習を行った.
  (3) 亀谷と協力して基礎学習を簡潔にまとめた解説ノートを作成した.
  (4) 中間発表会に向けて，千葉・亀谷と協力して来るであろう質問を予測して対策を行った.
  (5) ECM の改善に直結するような文献を探した。
  (6) 中間発表会に向けて、ポスターの「理論班」の章を英訳した。
- 千葉大樹（理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 亀谷・橋本と協力して入門書を読み，基礎学習を行った.
  (3) 中間発表会に向けて，ECM についての英論文から重要な単語を抜粋し解説した.
  (4) 中間発表会に向けて，亀谷・橋本と協力して来るであろう質問を予測して対策を行った.
  (5) 中間発表会に向けて、ポスターの「プログラミング班」の章を英訳した。
- 亀谷浩也（理論班）
  (1) 楕円曲線法の基礎を学んだ.
  (2) 橋本・千葉と協力して入門書を読み，基礎学習を行った.
  (3) 橋本と協力して，基礎学習を簡潔にまとめた解説ノートを作成した.
  (4) 中間発表会に向けて，橋本・千葉と協力して来るであろう質問を予測して対策を行った.
  (5) 中間発表会に向けて、ポスターの「概要・基礎学習」の章を英訳した。

（※文責: 池野竜將）

第 6 章　前期活動成果

本プロジェクトでは，理論班で理解することに成功した高速化アルゴリズムをプログラミング班に伝え，プログラミング班がそのアルゴリズムを実装することにより ECM プログラムを作成した．

（※文責: 千葉大樹）

6.1　理論班

理論班は，活動内容で示した，エドワーズ曲線においての射影座標を用いたスカラー倍楕円曲線プログラムでの変数の点の与え方のアルゴリズムの改善点を発見した．乗算の回数，除算の回数が減少したことにより素因数を発見する効率が理論上 1.5 倍減少したが，実装前との計算コストの実数値の比較についてはまだできていない．また，Atkin-Morain ECPP のアルゴリズムの理解をすることに成功した．これを実装することにより，ECM によって素因数 p が見つかる確率は，位数があらかじめ小さな因数 d を持つ曲線のみを使用した場合，ランダムに動く部分のサイズが p から p=d に減少するため因数分解に成功する確率を高めることができる．しかし，Atkin-Morain ECPP の理論については理解することができなかった．そのため，プログラミング班には Atkin-Morain ECPP の実装のためのアルゴリズムを書き起こしレポート用紙を渡すことにより，ECM USING EDWARDS CURVE の読解を終了した．

（※文責: 駒ヶ嶺壮）

6.2　プログラミング班

プログラミング班では，新たなアルゴリズムを実装し，理論上は 6.1 のように計算量が減少することが分かった．詳細な実験は行っておらず有意な差があるかどうかは確認できていない．だが，実際に素因数分解を行った結果，処理が早くなっていることが確認できた．

また，実際に巨大な合成数を分解し，昨年度のプログラムとの性能を比較することにした．評価するにあたって，2015 年度に作成されたプログラムでテストに使用されていた合成数 $10^{306}+1$ を素因数分解することで，以前のプログラムとの比較をすることとした．2015 年度のプログラムでこの合成数を分解した結果，発見されたもっとも大きな素因数は 15753898031981660 7(21 桁) で

表 6.1　昨年度と今年度のプログラムの計算コストの比較

	2 倍算	2 倍算→加算
昨年度	3**M**+4**S**+1**D**[*1]	13**M**+5**S**+3**D**
今年度	3**M**+4**S**+1**D**	12**M**+4**S**+1**D**

[*1] **M**:乗算，**S**:2 乗算，**D**:楕円曲線の係数 a,d を用いた乗算

あった．同様に今年改善されたプログラムで分解した結果，発見されたもっとも大きな素因数は 112544281755782732673671367061(30 桁) であり，より大きな素因数を見つけることができるように改善された．

（※文責: 源啓多）

第 7 章　後期活動成果

7.1　理論班

（※文責: 橋本和典）

7.2　プログラム班

（※文責: 橋本和典）

7.3　広報班

（※文責: 橋本和典）

第 8 章　まとめ

8.1　前期活動結果

前期は参考資料，論文，担当教員の白勢先生の講義による楕円曲線法の理解から始め，楕円曲線が楕円曲線法においていつどのように使われるかを理解した．その後，理論班，プロジェクト班の 2 班に分かれ作業を行った．理論班は，論文，入門書の読解をし，プログラム高速化のための改善案を出すことに成功した．しかし，前期中にプログラミング班が実装することはできなかった．プログラミング班は前年度のプロジェクトで作成された ECM プログラムを理解した．その後，実装ミスの改善や，新たなアルゴリズムの実装を行い，計算コストの減少に成功した．

（※文責: 千葉大樹）

8.2　後期の展望

後期は，理論班が作成した Atkin-Morain ECPP アルゴリズムを実装し，さらに ECM プログラムの改善を図る．また，大きな合成数の分解を続け ECMNET へのランクインを目指す．加えて，前期中に活動できなかった広報について新たに班を設置し活動していく．

（※文責: 橋本和典）

8.3　後期活動結果

（※文責: 橋本和典）

8.4　全体を通して

（※文責: 橋本和典）

参考文献

[1] ECMNET. https://members.loria.fr/PZimmermann/ecmnet/, (最終アクセス 2016 年 7 月 20 日)

[2] Bernstein, D.J. , Birkner, P. , Lange, T. , Peters, C. ECM USING EDWARDS CURVES. Mathematics of Computation, 2013.

[3] Hisil, H., Wong, K.K.-H., Carter, G., Dawson, E. Twisted Edwards curves revisited. Advances in Cryptology - ASIACRYPT 2008, 2008.

[4] STUDIO KAMADA. http://stdkmd.com/, （最終アクセス 2016 年 7 月 15 日）.

[5] Joseph H. S., John T. 楕円曲線論入門丸善出版, 2012.

[6] 國廣 昇 , 鶴岡行雄, 小山謙二. 適切な位数を持つ楕円曲線に基づく素因数分解. SCIS, 1997.

[7] Kris Gaj, Soonhak Kwon, Patrick Baier, Paul Kohalbrenner, Hoang Le, Mohammed Khaleeluddin, Ramakrishna Bachimanchi. Implementing th Elliptic Curve Methof od Factoring in Reconfogurable Hardware. CHES-2006, 2006.

[8] 森下拓也,Jibhui Chao. 疑似的 2 次拡大環上の楕円曲線法. FIT2015, 2015.

[9] Henriette Heer, Gary McGuire, Oisin Robinson. JKL-ECM: an implemention of ECM using Hessian curves. LMS Journal of Computation and Mathmatcis, 2016.