

卒業研究論文

# Distorsion Map の $n$ -進展開を用いた Miller Algorithm の高速化に関する研究

Fast Computation of Miller Algorithm with  $n$ -adic expansion of Distortion maps

学籍番号 15D8102011G

増渕 佳輝

YOSHIKI MASUBUCHI

中央大学理工学部情報工学科

趙研究室

2019 年 3 月

# 概論

暗号で有用な双線形写像に, 楕円曲線上, または超楕円曲線上の Weil ペアリング, Tate ペアリングなどがある. 楕円曲線におけるペアリング演算では, Miller Algorithm や distortion map を用いた BKLS Algorithm が知られている. 本研究では BKLS Algorithm に Window 法を適用し, Tate ペアリングの高速化を行った. また sage を用いて実装し, 計算時間を求め, 計算コストと比較した.

## キーワード

- 楕円曲線
- ペアリング暗号
- Double-Base Chains
- Tate ペアリング
- Miller Algorithm

# 目次

|       |                                   |    |
|-------|-----------------------------------|----|
| 第 1 章 | 序論                                | 1  |
| 第 2 章 | 準備                                | 2  |
| 2.1   | 群の定義 . . . . .                    | 2  |
| 2.2   | 環の定義 . . . . .                    | 3  |
| 2.3   | 体の定義 . . . . .                    | 5  |
| 2.4   | 離散対数問題と ElGamal 暗号 . . . . .      | 6  |
| 第 3 章 | 楕円曲線                              | 7  |
| 3.1   | 楕円曲線の定義 . . . . .                 | 7  |
| 3.2   | 楕円曲線上の点の加算 . . . . .              | 7  |
| 3.3   | 楕円離散対数問題 . . . . .                | 9  |
| 第 4 章 | ペアリング                             | 10 |
| 4.1   | ペアリングの定義 . . . . .                | 10 |
| 4.2   | ペアリングと関係する様々な定義 . . . . .         | 10 |
| 4.3   | Weil の定理 . . . . .                | 11 |
| 4.4   | Weil ペアリング . . . . .              | 11 |
| 4.5   | Tate ペアリング . . . . .              | 12 |
| 4.6   | Miller Algorithm . . . . .        | 13 |
| 4.7   | distortion map . . . . .          | 15 |
| 4.8   | Supersingular curve . . . . .     | 15 |
| 第 5 章 | ペアリング演算の高速化                       | 16 |
| 5.1   | Signed Miller Algorithm . . . . . | 16 |
| 5.2   | 高速化手法 . . . . .                   | 17 |
| 5.3   | BKLS Algorithm . . . . .          | 18 |
| 5.4   | Window Miller Algorithm . . . . . | 18 |
| 第 6 章 | 提案手法                              | 20 |
| 6.1   | 提案手法のアルゴリズム . . . . .             | 20 |
| 6.2   | 提案手法の計算量 . . . . .                | 21 |
| 第 7 章 | 評価と考察                             | 22 |
| 7.1   | 評価 . . . . .                      | 22 |
| 7.2   | 考察 . . . . .                      | 23 |
| 7.3   | 今後の課題 . . . . .                   | 23 |

|      |    |
|------|----|
| 謝辭   | 24 |
| 参考文献 | 25 |

# 第 1 章

## 序論

インターネットを代表とするコンピュータネットワーク等の情報通信技術の発展により、インターネットは我々の生活にとって無くてはならない技術となった。その発展により、最近では様々なものに情報技術を組み込む「IoT(Internet of Things)」と呼ばれる技術なども登場し、我々に生活の利便性を向上させている。しかしその一方で、インターネット上でのクレジットカードの番号を通信する際や、機密性の高い情報を送信する際など、通信される情報が盗み取られ、複製、改ざんされる危険性がある。そのため、第 3 者に見られたくない情報を守る必要がある。これを実現し、さらには通信している相手が本当に目的の人物なのかを確かめるため、認証なども行うようにするのが情報セキュリティ技術である。この情報セキュリティ技術の核となる技術の一つが暗号であり、世界中で盛んに研究されている。

楕円曲線暗号とは有限体上の楕円曲線を用いた暗号で、これに対する攻撃方法としてペアリングが用いられた。ペアリングとは楕円曲線上で定義される双線形写像である。2000 年以降、暗号プリミティブとして広く利用されるようになった。具体例として ID を公開鍵として利用可能な Identity Based Encryption, 既存方式より短い署名長で済む Short Signature などがあり、従来にない特性を有するプロトコルを構成することが可能である。しかし、主要な暗号要素技術と比較して計算コストが大きく、効率的な演算アルゴリズムが求められている。

ペアリングの演算方法として、Miller アルゴリズムが一般的に知られている。このアルゴリズムを用いて pairing 演算を実装すると、通常の楕円スカラー倍演算などに比較して演算量が多いため、速度が遅くなることが問題である。このため、Miller のアルゴリズムの高速実装の研究が盛んに行われている。これらの既存研究として distortion map を用いることで効率化した BKLS Algorithm, 楕円スカラー倍演算の高速化などに使われる window 法と Miller のアルゴリズムを組み合わせた window Miller's Algorithm が提案されている。

本研究では、BKLS Algorithm に window 法を適用し Miller Algorithm の高速化を図った。

## 第2章

# 準備

### 2.1 群の定義

集合  $G$  の直積集合  $G \times G$  から集合  $G$  への写像が1つ与えられているとき、この写像を  $G$  の2項演算と呼ぶ。2項演算が与えられ、次の条件全てを満足する場合  $G$  はこの演算に関して群という。

1. 結合律

$\forall a, b, c \in G$  に対して常に、 $(a * b) * c = a * (b * c)$  が成立する。

2. 単位元の存在

$e \in G, \forall a \in G$  に対して  $a * e = e * a = a$  が成立する。

3. 逆元の存在

$G$  に属する任意の元  $a$  に対して  $a * b = b * a = e$  となる元  $b$  が存在する。

2項演算が1. のみを満たす  $G$  と  $*$  の組  $(G, *)$  はこの2項演算に関する半群という。

更に、群  $G$  が

4. 交換法則

$a, b \in G$  に対し、 $a * b = b * a$  を満たすとき、群  $G$  は可換群であるという。

集合  $G$  が2項演算  $*$  に対して可換群であるとき  $*$  が、 $+$  で表される場合その群を加法群と呼ぶ。そのとき  $x + y$  を  $x$  と  $y$  の和といい、単位元を  $0$ ,  $x$  の逆元を  $-x$  で表す。

群  $(G, *)$  において2項演算が明らかな場合単に群  $G$  ということもある。

また、群  $G$  に属する元の個数が有限であるとき  $G$  を有限群、そうでないとき無限群という。

可換群  $G$  の任意の元が1つの元  $a$  のべき乗で表せるとき、 $G$  を  $a$  で生成された巡回群といい、 $G = \langle a \rangle$  で表す。 $a$  を生成元、あるいは原始元という。

## 2.2 環の定義

### 2.2.1 環

2 種類の 2 項演算 (加法  $+$  と乗法  $\cdot$ ) の定義された集合  $R$  が次の条件を満足するとき,  $(R, +, \cdot)$  は環であるいう.

1. 加法に関して可換群をなす.

$$(a + b) + c = a + (b + c)$$

$$a + b = b + a$$

$$0 + a = a + 0$$

$$(-a) + a = a + (-a) = 0$$

2. 乗法に関して半群をなし乗法に関する単位元が存在する.

$a \in R$  に対して  $a \cdot e = e \cdot a = a$  となる  $e \in R$  が存在する.

3. 分配法則

$a, b, c \in R$  に対して,

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

$$(a + b) \cdot c = a \cdot c + b \cdot c$$

が成立する.

更に, 環  $R$  において,

4. 交換法則

$$a, b \in R \text{ に対し, } a \cdot b = b \cdot a$$

を満たすとき, 群  $R$  は可換環であるといい, そうでないときを非可換環という.

環における単位元は加法単位元  $0_R$  と, 乗法単位元  $1_R$  がある. 環の乗法の記号  $\cdot$  は省略されることが多い. すなわち,  $x \cdot y$  は  $xy$  と書かれる. 以下この記法で書くことにする.

### 2.2.2 部分環, イdeal, 商環

環  $R$  の部分集合でそれ自身が  $R$  の演算において, 環になるものを  $R$  の部分環という.

環  $R$  の部分集合  $I$  において,

1.  $a, b \in I$  ならば  $a + b \in I$
2.  $a \in I$  と  $r \in R$  に対して  $ra \in I$
3.  $a \in I$  と  $r \in R$  に対して  $ar \in I$

条件 1. と 2. を満たすとき,  $I$  は  $R$  の左イdeal, 条件 1. と 3. を満たすとき,  $I$  は  $R$  の右イdeal, 条件 1. から 3. まで全てを満足するとき,  $I$  は  $R$  の 両側イdeal または単にイdeal という.  $R$  が可換環の場合は左イdeal, 右イdeal, 両側イdeal は一致する.

環  $R$  中のイdeal  $I$  の生成元の集合とは,  $I$  の元の集合であって,  $I$  の任意の元がその集合の元の  $R$  係数の有限な 1 次結合であるようなものである. イdeal はもし生成元の有限集合をもつなら, 有限生成といわれる.  $I$  が元の集合  $\{f_1, \dots, f_l\} \subset I$  によって生成されるなら,  $I = \sum_{i=1}^l Rf_i$ , または単に  $I = (f_1, \dots, f_l)$  と書く.

ここで, 整数  $a, b$  の差  $a - b$  が自然数  $n$  で割り切れるとき,  $a, b$  は法  $n$  に関して合同であるといい,  $a \equiv b \pmod{n}$

と表現する. 互いに合同な整数全体の集合を剰余類という. 環  $R$  の元  $x, y$  がイデアル  $I$  を法として合同であるとは,  $x + i = y$  となる元  $i \in I$  が存在することであり, このとき  $x \equiv y \pmod{I}$  と書く. この関係は同値関係である. 環  $R$  のイデアル  $I$  による同値類を  $[x]$  と書けば,  $[x] = x + I = \{x + i | i \in I\}$  となり, 同値類の集合  $R/I$  に加法と乗法, つまり,

$$[x] + [y] = [x + y], \quad [x] \cdot [y] = [xy] \quad (2.1)$$

が定義できる. これらの演算に関して同値類の集合  $R/I$  は環になり,  $I$  を法とする  $R$  の商環または剰余環 という.

### 2.2.3 多項式環

可換環  $R$  において,  $x$  を不定元 (変数) としたとき,  $R$  上の多項式の集合は,

$$\{a_n x^n + \cdots + a_1 x + a_0 \mid a_0, a_1, \dots, a_n \in R, n \text{ は } 0 \text{ か正の整数}\} \quad (2.2)$$

と定義される.  $f(x) = b_n x^n + \cdots + b_1 x + b_0$  が  $R$  上の多項式で  $b_n \neq 0$  としたとき,  $n$  を多項式  $f$  の次数といい,  $\deg f$  と表す. 特に,  $n = 0$  のとき,  $f(x) = b_0 \in R$  となるが, これを定数と呼ぶ.  $0 \in R$  の次数は  $-\infty$  とする. また, 最高次の係数が 1 である多項式をモニック多項式という.

$x$  を不定元とする可換環  $R$  上の多項式全体の集合には,  $R$  における 2 項演算を用いて, 次のように 2 項演算を定義することができる.  $R$  上の 2 つの多項式  $f(x) = b_n x^n + \cdots + b_1 x + b_0$ ,  $g(x) = c_m x^m + \cdots + c_1 x + c_0$  に対して,

$$f(x) + g(x) = \sum_{k \geq 0} (b_k + c_k) x^k \quad (2.3)$$

$$f(x) \cdot g(x) = \sum_{k=0}^{n+m} \left( \sum_{i+j=k} b_i c_j \right) x^k \quad (2.4)$$

と 2 つの 2 項演算  $+$  と  $\cdot$  を定めると, これに関して,  $x$  を不定元とする  $R$  上の多項式の全体集合は可換環になる. ただし,  $x^0 = 1$ ,  $0 \cdot x = 0$  ( $0, 1 \in R$ ) と定める. こうして得られた環を,  $R$  上の多項式環と呼び,  $R[x]$  と表す.

可換環  $R$  上の多項式  $f(x)$  が, 1 次以上の多項式  $g(x)$ ,  $h(x) \in R[x]$  によって,  $f(x) = g(x)h(x)$  となるとき,  $g(x)|f(x)$ ,  $h(x)|f(x)$  と表し,  $g(x)$ ,  $h(x)$  を  $f(x)$  の因子と呼ぶ.  $f(x) \in R[x]$  が因子を持たないとき,  $f(x)$  は  $R$  上既約であるといわれる.  $f(x)$  が既約でないとき, 可約であるという.

$T$  を  $T \supset R$  であり,  $R$  で定義されている 2 項演算 に対して, 環になっているとする. このとき, 不定元  $x$  に  $T$  の元  $t$  を代入することにより,

$$f(t) = b_n t^n + \cdots + b_1 t + b_0 \in T, \quad b_i \in R \quad (2.5)$$

が得られる.  $f(t) = 0 \in R$  となるときの  $t$  を,  $f(x)$  の零点という.



## 2.3 体の定義

集合  $\mathbb{F}$  が次の条件を満たすとき、 $\mathbb{F}$  は体である。

集合  $\mathbb{F}$  に対して加法と乗法が定義されているとする。

1. 環である。
2. 0 以外の  $\mathbb{F}$  における全ての元には乗法に関し逆元が存在する。

このとき、環が可換環であるならば可換体という。

もし、 $\mathbb{F}$  において単位元 1 をそれ自身に加えていっても決して 0 にならないならば、 $\mathbb{F}$  の標数は 0 であるといい、 $\text{char}(\mathbb{F})=0$  と書く。この場合、 $\mathbb{F}$  に含まれる最小の体は有理数体  $\mathbb{Q}$  である。そうでない場合、 $1+1+\cdots+1$  ( $p$  回) が 0 に等しいような素数  $p$  があり、 $\mathbb{F}$  の標数は  $p$  であるといい、 $\text{char}(\mathbb{F}) = p$  と書く。この場合、 $\mathbb{F}$  に含まれる最小の体は  $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$  である。この様な最小の体を  $\mathbb{F}$  の素体という。

### 2.3.1 有限体，ガロア体

体  $\mathbb{F}$  の中で、有限個の元からなるものを有限体あるいはガロア体という。無限個の元からなる体を無限体という。元の数  $q$  の有限体を  $\mathbb{F}_q$  で表す。ガロア体  $\mathbb{F}_q$  は元の数  $q$  が素数  $p$  あるいは素数のべき乗  $p^m$  のときに限り存在する。体  $\mathbb{F}$  の元を係数とする多項式を、 $\mathbb{F}$  上の多項式という。 $\mathbb{F}_q$  上における多項式間の係数演算は  $\mathbb{F}_q$  の演算である。なお、有限体の中で最も小さい体は 0 と 1 の二つの元からなる体  $\mathbb{F}_2$  である。

### 2.3.2 拡大体

$L$  を体、 $K$  を  $L$  の部分体としたとき、 $L$  を  $K$  の拡大体という。さらに  $M$  が  $L$  の部分体であり、 $K$  の拡大体であるとき、 $M$  を  $L$  と  $K$  の中間体という。

$\mathbb{F}$  を含む拡大体  $\mathbb{K}$  の元  $\alpha$  は、もし  $f(\alpha) = 0$  である 1 変数多項式  $f(X) \in \mathbb{F}[X]$  があるなら、 $\mathbb{F}$  上代数的であるという。

この場合、 $\mathbb{F}[X]$  の中に  $\alpha$  が根である monic 既約多項式が一意に存在する。そして  $\alpha$  が満たす他のどんな多項式も、この monic 多項式で割ることができなければならない。この monic 既約多項式は、 $\alpha$  の最小多項式と呼ばれる。もし、 $\alpha$  の最小多項式が次数  $d$  を持つならば、 $\mathbb{F}(\alpha)$  における任意の元、すなわち  $\alpha$  のべきと  $\mathbb{F}$  の元に関する任意の有理式は、べき  $1, \alpha, \alpha^2, \dots, \alpha^{d-1}$  の 1 次結合として表現できる。

したがって、これらの  $\alpha$  のべきは  $\mathbb{F}$  上における  $\mathbb{F}(\alpha)$  の基底をなす。よって、 $\alpha$  を付加することにより得られる拡大次数は  $\alpha$  の最小多項式における次数と同じである。

### 2.3.3 代数的閉包

体  $\mathbb{F}$  に係数を持つ全ての多項式が、1 次因子に完全に分解するという性質を持つならば、 $\mathbb{F}$  は代数的に閉じているという。同じことであるが、 $\mathbb{F}$  に係数を持つ全ての多項式が  $\mathbb{F}$  に根を持つことを要求すれば充分である。代数的に閉じている最小の  $\mathbb{F}$  の拡大体は、 $\mathbb{F}$  の代数的閉包といい、 $\overline{\mathbb{F}}$  と表す。

## 2.4 離散対数問題と ElGamal 暗号

公開鍵暗号とは、暗号化鍵は公開し、誰もが使えるようにしておくが、復号に使う鍵は秘密にする暗号である。公開鍵から秘密鍵を求めることは困難なので、暗号文の正規の受信者以外は暗号文を解読できないという原理になっている。

公開鍵暗号の一例として ElGamal 暗号が挙げられる。ElGamal 暗号は、離散対数問題という問題の困難さに基づく公開鍵暗号である。まず、離散対数問題の定義を述べ、その後に ElGamal 暗号の暗号化と復号アルゴリズムを述べる。

### 2.4.1 離散対数問題

群  $G$  における  $g \in G$  に対する離散対数問題とは、 $y \in G$  が与えられるとき、 $g^x = y$  (演算を加法的に書くと  $xg = y$ ) である整数  $x$  が存在するとしたとき、それを求めるという問題のことである。この  $x$  を  $y$  の離散対数という。

### 2.4.2 ElGamal 暗号

使う群は、大きな素数を  $p$  として、 $\mathbb{Z}_p$  の乗法群  $\mathbb{Z}_p^* = \{1, 2, \dots, p-1\}$  である。まず最初に、受信者は  $\mathbb{Z}_p^*$  の原始元  $g$  を選ぶ。次に、 $\{0, 1, \dots, p-2\}$  から  $x$  をランダムに選び、 $y = g^x \pmod{p}$  を計算する。最後に受信者は、 $P_K = (p, g, y)$  を公開鍵として公開し、秘密鍵  $S_K = x$  を秘密に保持する。

次に暗号化であるが、送信者は受信者の公開鍵  $(p, g, y)$ 、平文  $m \in \mathbb{Z}_p$  を入力とし、暗号文  $C = (c_1, c_2)$  を、 $r \in \{0, 1, \dots, p-2\}$  をランダムに選び、 $c_1 = g^r \pmod{p}$ 、 $c_2 = my^r \pmod{p}$  として求める。これを送信者は受信者に送る。

最後に復号であるが、受信者は秘密鍵  $x$ 、暗号文  $C = (c_1, c_2)$  を入力とし、平文  $m$  を  $m = c_2 c_1^{p-1-x} \pmod{p}$  として求める。

## 第3章

# 楕円曲線

### 3.1 楕円曲線の定義

楕円曲線とは、一般的に

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}_q)$$

で与えられる  $(x, y)$  に関する方程式のことである。係数  $a_n$  が属する体  $\mathbb{F}_q$  を係数体、変数  $x, y$  が属する体を定義体と呼ぶ。このとき、体  $K$  上の楕円曲線とは、この方程式に無限遠点と呼ばれる要素  $\mathcal{O}$  を加えた  $x, y \in K$  である点  $(x, y)$  の集合を表す。

もし、 $K$  の標数が2であるとき、方程式は

$$y^2 + xy = x^3 + ax^2 + b \quad (a, b \in \mathbb{F}_q)$$

と変形され、上式を満たす点の集合となる。

また、 $K$  の標数が3であるときは、方程式は

$$y^2 = x^3 + ax^2 + bx + c \quad (a, b, c \in \mathbb{F}_q)$$

と変形され、標数が3より大きい場合は  $y^2 = x^3 + bx + c$  と変形される。

### 3.2 楕円曲線上の点の加算

楕円曲線上の有理点において、通常座標で行われる点の加算とは異なる加算の定義をする。楕円曲線上の点  $P, Q$  を取ってきたとき、まず点  $P, Q$  を通る直線を引き、第三の交点  $P * Q$  を見つける。次に、 $P * Q$  と無限遠点  $\mathcal{O}$  を通る直線 ( $x$  軸との垂線) を引き、楕円曲線と交わるもう1つの交点を楕円曲線における点  $P$  と  $Q$  が加算された点  $P + Q$  とする。

一方、 $P = Q$  である場合は楕円曲線との点  $P$  における接線を引き、その交点を  $P * Q$  とする。

また、この加算により楕円曲線は群構造をなす。例えば、点  $P = (x, y)$  と  $Q = (x, -y)$  の場合、第三の交点は無限遠点となる。よって、 $P + Q = \mathcal{O}$  となり点  $Q$  が点  $P$  の逆元、 $-P$  となる。すなわち単位元が無限遠点、逆元は  $x$  軸と対称な点になる。結合法則は、加算の定義により自明である。また、無限遠点同士の加算は無限遠点となる。

この定義を実数上で描かれた楕円曲線のグラフを用いて示す。ただし、 $P, Q \in E(\mathbb{F}_q)$  について  $P = (x_1, y_1), Q = (x_2, y_2)$  とする。

Case 1.  $P \neq Q, x_1 \neq x_2 \rightarrow$  図 3.1

Case 2.  $P = Q \rightarrow$  図 3.2

Case 3.  $P \neq Q, x_1 = x_2 \rightarrow$  図 3.3

Case 4.  $P = \mathcal{O}$  あるいは  $Q = \mathcal{O} \rightarrow$  図 3.4

Case 5.  $P = Q = \mathcal{O}$

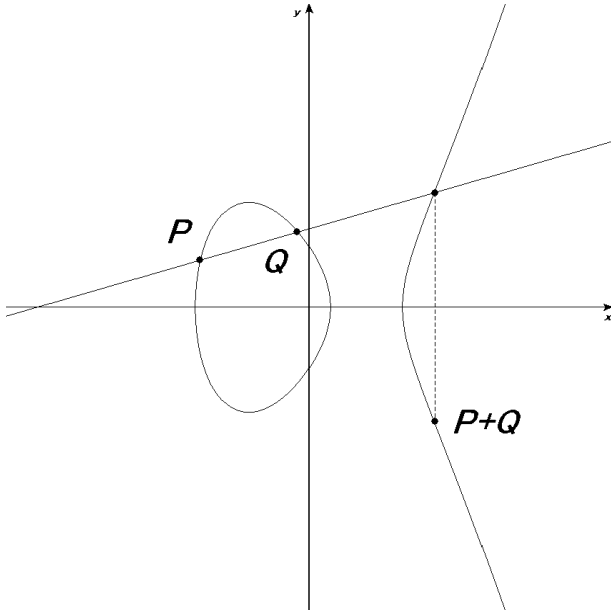


図 3.1  $P + Q$

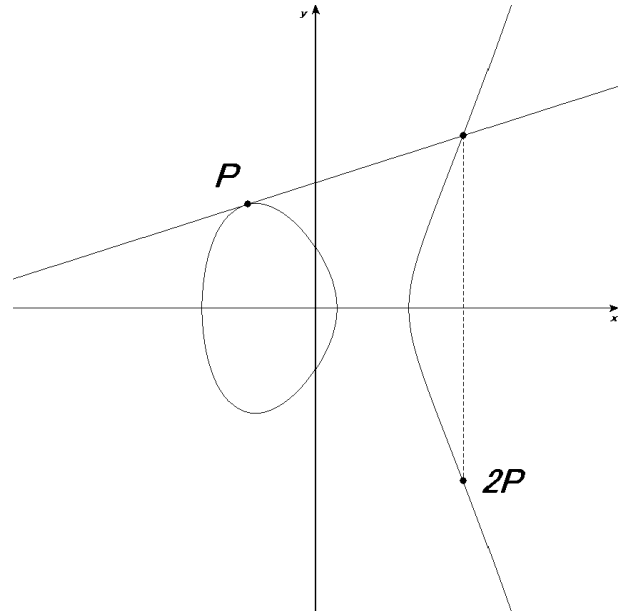


図 3.2  $2P$

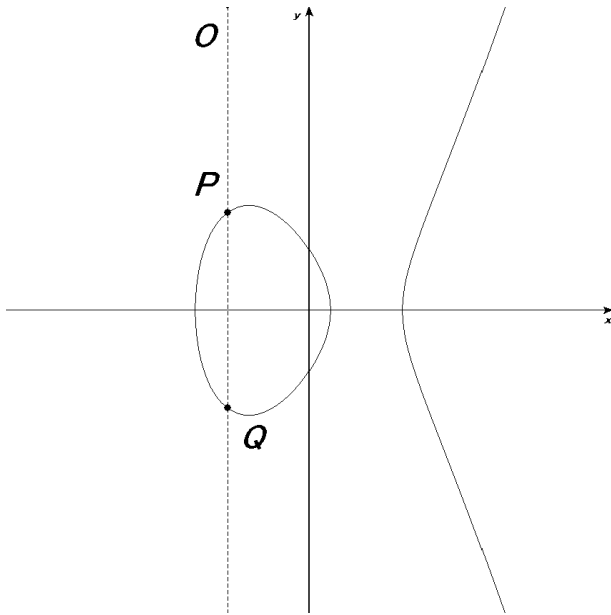


図 3.3  $P + Q = \mathcal{O}$

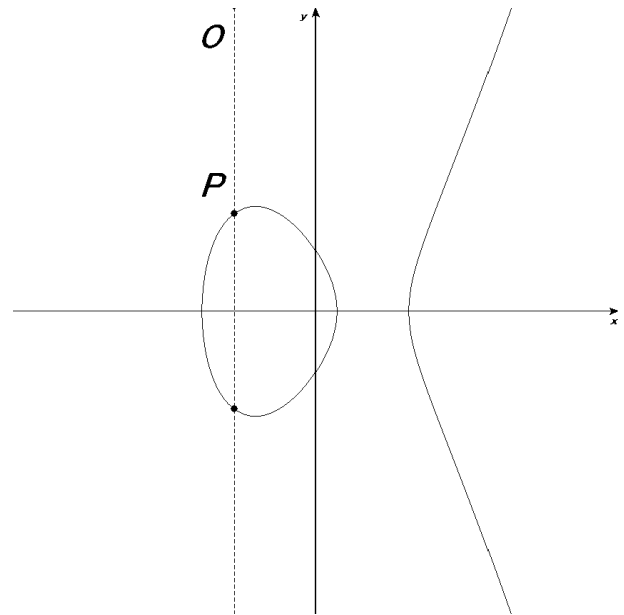


図 3.4  $P + \mathcal{O} = P$

ここで,  $P + Q$  を効率的に計算できるよう公式を与える. 楕円曲線を  $y^2 = x^3 + ax^2 + bx + c$  とし, 各点を

$$P_1 = (x_1, y_1), \quad P_2 = (x_2, y_2), \quad P * Q = (x_3, y_3), \quad P + Q = (x_3, -y_3)$$

とする. このとき,  $P_1$  と  $P_2$  を結ぶ直線の方程式は

$$y = \lambda x + \nu, \quad \lambda = \frac{y_2 - y_1}{x_2 - x_1}, \quad \nu = y_1 - \lambda x_1$$

となり，これを楕円曲線の方程式に代入して

$$x_3 = \lambda^2 - a - x_1 - x_2, \quad y_3 = \lambda x_3 + \nu$$

となる．また，点  $P$  と点  $P$  の加算は，2 倍算と定義され， $\lambda = \frac{f'(x)}{2y}$  を用いて 2 倍点の  $x$  座標は

$$2 \text{ 倍点の } x \text{ 座標} = \frac{x^4 - 2bx^2 - 8cx + b^2 - 4ac}{4x^3 + 4ax^2 + 4bx + 4c}$$

と表される．

### 3.3 楕円離散対数問題

#### 3.3.1 楕円離散対数問題 (ECDLP)

任意の点  $P \in E(\mathbb{F}_q)$  に対して， $\langle P \rangle = \{\mathcal{O}, P, 2P, 3P, \dots\}$  は有限巡回群となる．この巡回群の位数を  $n$  とすると任意の  $Q \in \langle P \rangle$  に対して，

$$xP = Q \quad x \in (\mathbb{Z}_n)$$

となる  $x$  がただ一つ存在する． $P$  と  $Q$  が与えられたとき， $x$  を求める問題を楕円離散対数問題という．これは  $x$  と  $P$  から  $xP = Q$  となる  $Q$  を求めるのは簡単だが， $P$  と  $Q$  から  $x$  を求めるのは非常に困難であることに基づいている．

## 第 4 章

# ペアリング

### 4.1 ペアリングの定義

$n$  を整数とする.  $G_1, G_2$  を単位元  $0$  の加法アーベル群とする.  $G_1, G_2$  は位数  $n$  を持つ.  $G_3$  は単位元  $1$  の乗法に関する位数  $n$  の巡回群とする. ペアリングというのは以下の関数である.

$$e : G_1 \times G_2 \longrightarrow G_3$$

全てのペアリングは以下の 2 つの性質を満たす.

・双線形性 全ての  $P, P' \in G_1$  と  $Q, Q' \in G_2$  に対して,

$$e(P+P', Q) = e(P, Q) + e(P', Q),$$

$$e(P, Q+Q') = e(P, Q) + e(P, Q') \text{ が成り立つ.}$$

・非退縮性

全ての  $P \in G_1$  ( $P \neq 0$ ) に対して  $e(P, Q) \neq 1$  となるような  $Q \in G_2$  が存在する.

全ての  $Q \in G_2$  ( $Q \neq 0$ ) に対して  $e(P, Q) \neq 1$  となるような  $P \in G_1$  が存在する.

### 4.2 ペアリングと関係する様々な定義

#### 4.2.1 divisor の定義

$C$  を体  $K$  上の楕円曲線とし,  $C(\overline{K})$  を体  $K$  上の代数閉包上で定義される全ての有理点の集合とする.  $C$  上の divisor とは, 次のような形式和で表される.

$$D = \sum_{P \in C(\overline{K})} n_P(P)$$

このとき,  $n_P \in \mathbb{Z}$  は有限であり,  $D$  は  $n_P = 0$  となるようなものを除いたものとする.  $C$  上の divisor の集合は  $\text{Div}_{\overline{K}}(C)$  で表され, 加法に関して群構造をなす. divisor  $D$  の台とは,  $\text{supp}(D) = \langle P \in C(\overline{K}) | n_P \neq 0 \rangle$  であるとする. divisor  $D$  の次数とは  $\deg(D) = \sum_P n_P$  であるとする. divisor  $D$  の和とは,  $\text{sum}(D) = \sum_P n_P P$  であるとする.

もし, 直線  $f$  が  $C$  上で零でない関数だとすると, 点  $P$  における  $f$  の重複度  $\text{ord}_P(f)$  を数えることができる.  $\text{ord}_P(f)$  は  $f(P) = 0$  のとき正であり,  $f$  が点  $P$  で極ならば負である. また,  $\text{ord}_P(f)$  が 1 ならば  $f = 0$  と  $E$  が交差し, 2 ならば  $f = 0$  が  $E$  に接し  $3P \neq \mathcal{O}$  となり, 3 ならば  $f = 0$  が  $E$  に接し  $3P = \mathcal{O}$  となる. 零でない関数  $f$  の divisor は  $(f)$  と書き,

$$\sum_{P \in C(\overline{K})} \text{ord}_P(f)(P)$$

である。これにより、直線  $f, g$  の divisor の計算は  $(fg) = (f) + (g)$  となり  $(f/g) = (f) - (g)$  となることがわかる。また、 $C$  の principal divisor とはある関数  $f$  に対して  $(f)$  と等しい divisor のことである。このとき、 $\deg((f)) = 0$  となる。点  $(P, Q)$  を通る直線  $l_{P,Q}$  を考えたとき、この直線の divisor を求める公式は

$$\operatorname{div}(l_{P,Q}) = (P) + (Q) + (-(P+Q)) - 3(\mathcal{O})$$

で表される。

#### 4.2.2 埋め込み次数の定義

$K_0 = \mathbb{F}_q$  を有限体とする。  $E$  を  $K_0$  上で定義された楕円曲線とし、 $\#E(K_0)$  で割り切れ、 $q$  と素な整数を  $n$  とする。体  $K = K_0(\mu_n)$  はある拡大体  $\mathbb{F}_{q^k}$  とする。  $k$  は埋め込み次数や安定乗数と呼ばれ、 $(q^k - 1)$  が  $n$  を割り切るような最小の正整数である。  $k$  というのは、 $q$  と  $n$  の関数  $k(q, n)$  である。  $k$  は  $n$  を法とした  $q$  の位数なので、 $k$  は  $\phi_{Eul}(n)$  (オイラーの  $\phi$  関数) で割り切られる。

任意の体  $K$  と任意の楕円曲線  $E$  に関して、もし  $n$  を  $\#E(\mathbb{F}_q)$  の大きな divisor とすると、埋め込み次数  $k$  は大抵の場合とても大きく (bit も  $n$  と同じ数だけある)、そのため体  $\mathbb{F}_{q^k}$  上の計算は指数的に複雑になる。

### 4.3 Weil の定理

ある楕円曲線  $E/\mathbb{F}_q$  に対して、曲線上の有理点の集合  $E(\mathbb{F}_q)$  を考える。このとき、 $E(\mathbb{F}_{q^m})$  の位数  $\#E(\mathbb{F}_{q^m})$  は  $E(\mathbb{F}_q)$  の位数  $\#E(\mathbb{F}_q)$  を用いて次のように求められる。

$$\begin{aligned}\#E(\mathbb{F}_{q^m}) &= q^m + 1 - t_{[m]} \\ t_{[m]} &= \alpha^m + \beta^m\end{aligned}$$

ただし、 $t$  を  $E(\mathbb{F}_q)$  のトレース  $t = q + 1 - \#E(\mathbb{F}_q)$  とする。このとき、 $|t| \leq 2\sqrt{q}$  が成り立つ。また、 $\alpha, \beta$  は  $\alpha\beta = q, \alpha + \beta = t$  を満たす複素数である。  $t_{[m]}$  は  $E(\mathbb{F}_{q^m})$  のトレースとする。  $t_{[m]}$  は  $E(\mathbb{F}_q)$  のトレースを用いて次式で与えられる。

$$t_{[m]} = \sum_{i=0}^{\lfloor m/2 \rfloor} \frac{m}{m-i} \alpha^{m-i} \beta^i (-q)^i t^{m-2i}$$

ここで、 $\lfloor m/2 \rfloor$  は  $m/2$  以下の最大の整数を意味する。Weil の定理を用いることで、定義体を拡大体とした場合の楕円曲線の位数を求めることができ、 $\#E(\mathbb{F}_q)$  は  $\#E(\mathbb{F}_{q^m})$  を割り切ることがわかる。

### 4.4 Weil ペアリング

$E$  を  $K_0$  上で定義された楕円曲線とし、 $n$  を  $K_0$  の標数と互いに素な整数とする。  $n$  で割り切れる位数の  $E(\overline{K})$  のすべての点の座標で生成された  $K_0$  の拡大体を  $K = K_0(E[n])$  で定義する。Weil ペアリングというのは、写像

$$e_n : E[n] \times E[n] \rightarrow \mu_n \subseteq K^*$$

で定義される。  $\mu_n$  は  $\overline{K}$  の単位元の  $n$  乗根とする。

$T \in E[n]$  とする。このとき、 $\operatorname{div}(f) = n(T) - n(\mathcal{O})$  なる関数  $f$  が存在する。  $nT' = T$  となるような  $T' \in E[n^2]$  を選ぶと、 $\operatorname{div}(g) = \sigma_{R \in E[n]}((T' + R) - (R))$  なる関数  $g$  が存在する。このとき、点  $R \in E[n]$  には  $n^2$  となる要素が含まれており、その場合は  $\sigma(T' + R)$  と  $\sigma(R)$  は打ち消される。  $g$  は  $T'$  の値によらないので、2つの違う  $T'$  を取ってきてもそれは  $R$  による。よって、 $\operatorname{div}(g) = \sigma_{nT''=T}(T'') - \sigma_{nR=\mathcal{O}}(R)$  と表される。

$f \circ n$  をある点を  $n$  倍した後に関数  $f$  を適応させるような関数とする.  $R \in E[n]$  であるような  $P = T' + R$  を選んだとき,  $nP = T$  である. このとき,  $\text{div}(f \circ n) = n(\sigma_R(T' + R)) - n(\sigma_R(R)) = \text{div}(g^n)$  であり,  $f \circ n$  は  $g^n$  の定数倍である. 適当に  $f$  を倍算したものを考えれば,  $f \circ n = g^n$  である.  $S \in E[n]$  とし  $P \in E(\overline{K})$  とすると,  $g(P + S)^n = f(n(P + S)) = f(nP) = g(P)^n$  となる. よって  $g(P + S)/g(P) \in \mu_n$  となる.  $g(P + S)/g(P)$  は  $P$  と独立となる.

以上より, Weil ペアリングというのは,

$$e_n(S, T) = \frac{g(P + S)}{g(P)}$$

で定義される.

Weil ペアリングは以下の特性を満たす.

1. (双線形性) すべての  $P, P', Q, Q' \in E[n]$  に対して,

$$e_n(P + P', Q) = e_n(P, Q)e_n(P', Q)$$

かつ

$$e_n(P, Q + Q') = e_n(P, Q)e_n(P, Q')$$

2. (一意性) 全ての  $P \in E[n]$  に対して,  $e(P, P) = 1$
3. (交換性) 全ての  $P, Q \in E[n]$  に対して,  $e_n(P, Q) = e_n(Q, P)^{-1}$
4. (非退化) もし全ての  $Q \in E[n]$  において  $e_n(P, Q) = 1$  ならば,  $P = \mathcal{O}$
5. (適合性) もし  $P \in E[nm]$  かつ  $Q \in E[n]$  ならば,

$$e_{nm}(P, Q) = e_n([m]P, Q)$$

6. もし  $\phi: E \rightarrow E'$  が 2 つの  $\hat{\phi}$  をもつ同種写像であるならば,

$$e_n(\phi(P), Q) = e_n(P, \hat{\phi}(Q))$$

## 4.5 Tate ペアリング

$P \in E(\mathbb{F}_q)[n]$  に対して  $\text{div}(f_{n,P}) = n(P) - n(\mathcal{O})$  となる有理関数  $f_{n,P} \in E(\mathbb{F}_q)$  が存在する. すべての点を  $n$  倍して得られることとなる点の集合を  $nE(\mathbb{F}_{q^k})$  とすると, その剰余類全体の集合を  $E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k})$  と表現する. 剰余類の代表元を  $Q$  として,  $D \sim (Q) - (\mathcal{O})$  となる因子  $D \in \text{Div}^0(E)$  を選択する.  $\text{supp}(\text{div}(f_{n,P})) \cap \text{supp}(D) = \emptyset$  を満足するようにランダムに選んだ点  $R \in E(\mathbb{F}_{q^k})$  を利用して  $D = (Q + R) - (R)$  とおくと,  $f_{n,P}(D)$  を計算可能である. Tate ペアリングは次のように定義可能である.

$$\langle \cdot, \cdot \rangle_n : \begin{cases} E(\mathbb{F}_q)[n] \times E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k}) \rightarrow \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^n \\ (P, Q) \mapsto \langle P, Q \rangle_n = f_{n,P}(D) \end{cases}$$

Tate ペアリングの値は剰余類全体の集合  $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^n$  に属しており, 一意に定まらない. すなわち, 二つの元  $a, b \in \mathbb{F}_{q^k}^*$  が元  $c \in \mathbb{F}_{q^k}$  を用いて  $a = bc^n$  と表現可能なとき,  $a, b$  は合同となる. ペアリングを利用する方式では  $\mathbb{F}_{q^k}^*$  における一意に定まる値が必要がため,  $c^n$  を消去する必要がある.  $a^{q^k-1} = 1, a \in \mathbb{F}_{q^k}^*$  の性質を利用して, Tate ペアリングの値に最終べき乗  $(q^k - 1)/n$  を行うことにより, 一意な値を得ることが可能である. Tate ペアリングの値を最終べき乗した Reduced Tate ペアリングを次のように定義する.

$$P \in E(\mathbb{F}_q)[n], Q \in E(\mathbb{F}_{q^k}), \mu_n = \{x \in \mathbb{F}_{q^k}^* | x^n = 1\}$$

$$\tau \langle P, Q \rangle = \langle P, Q \rangle_n^{(q^k-1)/n} = f_{n,P}(Q)^{(q^k-1)/n} \in \mu_n$$



さらに, Reduced Tate ペアリングの重要な性質として  $N = hn$  に対して次の式が成立する.

$$\tau(P, Q) = \langle P, Q \rangle_n^{(q^k-1)/n}$$

これにより,  $n$  を Hamming Weight が小さい  $N = hn \in \mathbb{N}$  に変更することにより, Miller Algorithm におけるループ回数は部分群の位数  $n$  を用いる場合に比べ増加するが, アルゴリズムの総計算量の低減が期待できる. Tate ペアリングは以下のような性質を持つ.

1. (双線形性) 全ての  $P, P', Q, Q' \in E[n]$  に対して,

$$\langle P + P', Q \rangle_n = \langle P, Q \rangle_n \langle P', Q \rangle_n$$

かつ

$$\langle P, Q + Q' \rangle_n = \langle P, Q \rangle_n \langle P, Q' \rangle_n$$

2. (非退化)  $\mathbb{F}_{q^k}$  を有限体とする. 全ての  $P \in E(\mathbb{F}_{q^k})[n]$  ( $P \neq 0$ ) に対して,  $\langle P, Q \rangle_n \neq 1$  となるような  $Q \in E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k})$  が存在する. 同様にして, 全ての  $Q \in E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k})$  ( $Q \notin nE(\mathbb{F}_{q^k})$ ) に対して,  $\langle P, Q \rangle_n \neq 1$  となるような  $P \in E(\mathbb{F}_{q^k})[n]$  が存在する.
3. (ガロアの不変性) もし  $\sigma \in \text{Gal}(\overline{\mathbb{F}_{q^k}}/\mathbb{F}_q)$  とすると,  $\langle \sigma(P), \sigma(Q) \rangle_n = \sigma(\langle P, Q \rangle_n)$  である.

## 4.6 Miller Algorithm

Miller Algorithm[3] とは, Weil ペアリングを多項式時間で求めるアルゴリズムである. また, Miller Algorithm は Tate ペアリングの計算にも用いることができる. Tate ペアリングにおいて Miller Algorithm を考える. 定義式

$$f_P(D_Q)^{(q^k-1)/n}$$

から, 以下の条件を満たす有理関数  $f_P$  の計算に帰着される.

$$\text{div}(f_P) = n(P) - n(\mathcal{O})$$

ここで  $f_h$  を

$$\text{div}(f_h) = h(P) - (hP) - (h-1)(\mathcal{O})$$

を満たす有理関数とする. 点  $iP$  と  $jP$  を通る直線を  $l_{iP,jP}$  とし, 点  $(i+j)P$  を通る垂線を  $v_{(i+j)P}$  とすると, Miller Algorithm 中で再帰的に  $f_P = f_h$  を求めるための基本公式

$$\text{div}(f_{i+j}) - \text{div}(f_i) - \text{div}(f_j) = \text{div}\left(\frac{l_{iP,jP}}{v_{(i+j)P}}\right)$$

は次のように証明される.

*Proof.* 直線の divisor の公式

$$\text{div}(l_{P,Q}) = (P) + (Q) + (-(P+Q)) - 3(\mathcal{O})$$

より,

$$\begin{aligned} \text{div}\left(\frac{l_{iP,jP}}{v_{(i+j)P}}\right) &= \text{div}(l_{iP,jP}) - \text{div}(v_{(i+j)P}) \\ &= \{(iP) + (jP) + (-(i+j)P) - 3(\mathcal{O})\} - \{((i+j)P) + (-(i+j)P) - 2(\mathcal{O})\} \\ &= (iP) + (jP) - ((i+j)P) - (\mathcal{O}) \end{aligned}$$

となる．一方  $f_h$  の定義から

$$\begin{aligned}\operatorname{div}(f_{i+j}) - \operatorname{div}(f_i) - \operatorname{div}(f_j) &= \{(i+j)P - ((i+j)P) - (i+j-1)(\mathcal{O})\} \\ &\quad - \{i(P) - (iP) - (i-1)(\mathcal{O})\} - \{j(P) - (jP) - (j-1)(\mathcal{O})\} \\ &= (iP) + (jP) - ((i+j)P) - (\mathcal{O})\end{aligned}$$

が得られ，両者は等しくなる．

従って

$$\operatorname{div}(f_{i+j}) = \operatorname{div}(f_i) + \operatorname{div}(f_j) + \operatorname{div}\left(\frac{l_{iP,jP}}{v_{(i+j)P}}\right) = \operatorname{div}\left(f_i f_j \frac{l_{iP,jP}}{v_{(i+j)P}}\right)$$

これより，再帰的公式

$$f_{i+j} = \left(f_i f_j \frac{l_{iP,jP}}{v_{(i+j)P}}\right)$$

が得られた．

□

これを用いて Miller Algorithm は， $(f) = n(P) - n(\mathcal{O})$  となるような関数  $f$  を構成している．次に Miller Algorithm を示す． $l = \log_2(n)$  とし，点  $U, V \in E(\mathbb{F}_{q^k})$  を通る直線の方程式を  $g_{U,V}$  とする． $U = V$  の場合  $V$  を通る

|   |
|---|
| Input: $n, P \in E(\mathbb{F}_q)[n], Q \in E(\mathbb{F}_{q^k})$               |
| Output: $f \in \mathbb{F}_{q^k}$  |
| 1: $Q' \leftarrow_R E(\mathbb{F}_{q^k})$                                      |
| 2: $S = Q + Q' \in E(\mathbb{F}_{q^k})$                                       |
| 3: $V \leftarrow P, f \leftarrow 1$   |
| 4: $n = \sum_{i=0}^{l-1} n_i 2^i, n_i \in 0, 1$                               |
| 5: for $j \leftarrow l-1$ down to 0   |
| 6: $f \leftarrow f^2 \cdot \frac{g_{V,V}(S)g_{2V}(Q')}{g_{2V}(S)g_{V,V}(Q')}$ |
| 7: $V \leftarrow 2V$  |
| 8: if $n_j = 1$ then  |
| 9: $f \leftarrow f \cdot \frac{g_{V,P}(S)g_{V+P}(Q')}{g_{V+P}(S)g_{V,P}(Q')}$ |
| 10: $V \leftarrow V + P$  |
| 11: return $f$  |

表 4.1 Miller Algorithm

Miller Algorithm の主な反復は  $l$  の繰り返しなので，二重の操作は  $l$  時間で成し遂げられる．5 行目から 10 行目までの掛け算と足し算の演算は  $n$  の Hamming Weight 以下で計算される．それゆえ Miller Algorithm は多項式時間で動作する．

適当な点  $S$  を選ぶ 1 つの方法としては， $E(\mathbb{F}_{q^k})$  上の点をランダムに選ぶことである． $n$  が大きいとき，このアルゴリズムは  $S = [i]P$  を取ることによって容易に決定的になる．このとき， $i$  の 2 進展開は  $n$  の 2 進展開の一部ではないとする ( $P \in E(\mathbb{F}_q)$  かつ  $Q \in E(\mathbb{F}_q)$  ならば  $S = Q$  を取る)．

このアルゴリズムの変化量は，直線プログラムとして関数  $f$  に対して明確な表示で出力される (例えば，もしそれが因数分解された形式を保つならば，小さな多項式の累乗の積として多項式の保存を必要とする)．

## 4.7 distortion map

distortion map というのは、 $E(\mathbb{F}_q)$  上の点を  $E(\mathbb{F}_{q^k})$  上の点に写像するような非線形な自己準同型写像である。点  $P \in E(\mathbb{F}_q)$  が位数  $n$  の点を持つとし、 $k > 1$  とする。 $E(\mathbb{F}_{q^k})$  は位数  $n^2$  の点を持たないとすると、 $E$  上の準同型  $\psi$  によって  $\psi(P) \notin E(\mathbb{F}_{q^k})$  ならば  $e(P, \psi(P)) \neq 1$  となる。

ペアリングの非退化について、distortion map が用いられる場合、 $E$  の非線形自己準同型  $\psi$  を用いて、 $e : G_1 \times G_1 \rightarrow G_3$  を

$$e(P, Q) = e(P, \psi(Q))$$

と定義することにより、非退化において全ての  $P \in G_1$  において

$$e(P, P) = e(P, \psi(P)) \neq 1$$

である。これは暗号方式を用いる際にかなり重要である。一意的な値を得るためには、ペアリングの値が 1 とならない場合についてのみ考えなければいけないからである。

## 4.8 Supersingular curve

上記の distortion map を持つような  $\mathbb{F}_q$  上の楕円曲線  $E$  は supersingular となる。Supersingular とは、以下の条件を 1 つでも満たす場合をいう。

1.  $\#E(\mathbb{F}_q) \equiv 1 \pmod{p}$  ( $p \mid t$  で  $\#E(\mathbb{F}_q) = q + 1 - t$  と等価)
2.  $E$  が  $\overline{\mathbb{F}}_q$  上で位数  $p$  の点を取らない。
3.  $\overline{\mathbb{F}}_q$  上の  $E$  の自己準同型環が非可換である。

もし  $E$  が supersingular でない場合、ordinary であるという。Supersingular curve は限られており、それは埋め込み次数が 6 以下であるので、ペアリングを用いた暗号系に適している。

## 第 5 章

# ペアリング演算の高速化

### 5.1 Signed Miller Algorithm

第 4 章で述べた Miller Algorithm では部分群の位数  $n$  を 2 進展開していた. Signed Miller Algorithm は符号付き 2 進展開を用いる手法である. 次にそのアルゴリズムを示す.

|   |
|---|
| Input: $n, P = (x_P, y_P) \in E(\mathbb{F}_q)[n], Q = (x_Q, y_Q) \in E(\mathbb{F}_{q^k})$   |
| Output: $f \in \mathbb{F}_{q^k}$  |
| 1: $Q' \in_R E(\mathbb{F}_{q^k})$   |
| 2: $S = Q + Q' \in E(\mathbb{F}_{q^k})$   |
| 3: $V \leftarrow P, f \leftarrow 1, f_{-1} = \frac{1}{x_Q - x_P}$                           |
| 4: $n = \sum_{i=0}^{l-1} n_i 2^i, n_i \in \{-1, 0, 1\}, n_0 = 1$                            |
| 5: for $j \leftarrow l-1$ down to 0   |
| 6: $f \leftarrow f^2 \cdot \frac{g_{V,V}(S)g_{2V}(Q')}{g_{2V}(S)g_{V,V}(Q')}$               |
| 7: $V \leftarrow 2V$  |
| 8: if $n_j = 1$ then  |
| 9: $f \leftarrow f \cdot \frac{g_{V,P}(S)g_{V+P}(Q')}{g_{V+P}(S)g_{V,P}(Q')}$               |
| 10: $V \leftarrow V + P$  |
| 11: if $n_j = -1$ then  |
| 12: $f \leftarrow f \cdot f_{-1} \cdot \frac{g_{V,P}(S)g_{V-P}(Q')}{g_{V-P}(S)g_{V,P}(Q')}$ |
| 13: $V \leftarrow V - P$  |
| 14: return $f$  |

表 5.1 Signed Miller Algorithm

このように符号付きで展開する方法は後述のアルゴリズムでも同様に用いることができる.

## 5.2 高速化手法

supersingular curve における Reduced Tate ペアリングの高速化手法が Barreto らによっていくつか提案された [1]. それを次に列挙する.

### 5.2.1 部分体の要素による乗算

埋め込み次数  $k$  の因子を  $d$  とすると, ペアリングの値を変えずに非零な要素  $x \in F_{q^d}$  を  $f_{n,P}(Q)$  に乗じることが可能である.  $q^k - 1 = (q^d - 1) \sum_{i=0}^{k/d-1} q^{id}$  と因数分解化のであり,  $k > 1$  のとき  $n|q^k - 1$ ,  $n \nmid q^d - 1$  となるため,  $n | \sum_{i=0}^{k/d-1} q^{id}$  である. すなわち,  $(q^k - 1)/n$  は因子として  $q^d - 1$  を必ず含んでいる. よって,  $f_{n,P}(Q)$  は最終べき乗  $(q^k - 1)/n$  の処理を行うことにより, フェルマーの小定理より乗じた値  $x^{(q^k-1)/n} = 1$  となる.

### 5.2.2 点による因子の置換

ペアリングの値は, 引数として与える  $Q \in E(K)$  とランダムに選択される  $R \in E(K)$  から構成される因子  $D = (Q+R)-(R)$  を用いて計算されるが, 因子  $D$  を点  $Q$  と置換しても正しく計算可能で  $P \in E(K_0)[n]$ ,  $Q \in E(K)$  は線形独立な点とすると, 次の式が成立する.

$$e(P, Q) = f_{n,P}(Q)^{q^k-1}$$

### 5.2.3 分母消去

Supersingular Curve であれば,  $Q'_1 \in E(K_0)$  に distortion map を適用させることにより,  $Q_1 = \psi(Q'_1) \in E(K)$  を得る. Ordinary Curve であれば, twist  $E'$  上の点  $Q'_2 \in E(\mathbb{F}_{q^e})$  から  $Q'_2 = \psi(Q'_2) \in E(K)$  を得る. ここで,  $m = \gcd(k, d)$ ,  $e = k/m$  である. よって  $Q = (x, y) \in E(K)$  において  $x \in \mathbb{F}_{q^{k/2}}$  となると, Miller Algorithm の 2 倍算または加算ステップに出現する分母の要素  $g_{2V}, g_{V+P}$  は  $e(P, Q)$  の値を変更することなく省略可能である.

### 5.2.4 群位数の Hamming Weight

Miller Algorithm は部分群の位数  $n$  に対する 2 進展開法に基づき, ペアリングの値を計算しているため, 加算ステップの処理回数は  $n$  の Hamming Weight (2 進展開した値における 1 の総数) に依存する. ランダムに選択された  $n$  の平均的な Hamming Weight はビット長の半分程度になるが, 部分群の位数として  $n = 2^\alpha \pm 2^\beta \pm 1$  を選択することにより, ペアリング演算コストを大幅に抑えることが可能である.

### 5.2.5 最終べき乗の高速化

Pairing-friendly field  $K$  における最終べき乗  $(q^k - 1)/n$  を想定する.  $k$  次の円分多項式  $\Phi_k(p)$  とすれば,  $n|q^k - 1$  であるため, 次のように変形できる.

$$f^{(q^k-1)/n} = (f^{(q^k-1)/\Phi_k(q)})^{(\Phi_k(q)/n)}$$

最終べき乗のコストは  $(\Phi_k(q)/n)$  乗のコストと同等,  $(\Phi_k(q)/n)$  のビット長は  $(\Phi_k(q)/k)\log_2(q^k) - \log_2(n)$  となる.

### 5.3 BKLS Algorithm

前述した高速化手法を用いて, supersingular curve の distortion map  $\psi$  を利用して分母消去の手法を適用した BKLS Algorithm [2] を次に示す. ordinary curve の場合,  $Q' \in E'(K)$  として, distortion map ではなく twist の同型写像  $\psi_d$  を用いる.

|   |
|---|
| Input: $P, Q \in E(K_0)[n]$                         |
| Output: $f \in K$                                   |
| 1: $f \leftarrow 1, V \leftarrow P$                 |
| 2: $n = \sum_{i=0}^{l-1} n_i 2^i, n_i \in \{0, 1\}$ |
| 3: for $j \leftarrow l-1$ down 0 do 0               |
| 4: $f \leftarrow f^2 \cdot g_{V, V}(\psi(Q))$       |
| 5: $V \leftarrow 2V$                                |
| 6: if $n_j = 1$ then                                |
| 7: $f \leftarrow f \cdot g_{V, P}(\psi(Q))$         |
| 8: $V \leftarrow V + P$                             |
| 9: return $f$                                       |

表 5.2 BKLS Algorithm

### 5.4 Window Miller Algorithm

window Miller アルゴリズムは、オンライン事前演算 を用いる方法である [5]。

まず、 $i = 1, \dots, 2w - 1$  に対して  $2w$  個の楕円曲線上の点  $P_i = iP$  を求め、各  $P_i$  において  $i(P) - i(O) = (P_i) - (O) + \text{div}(f_i)$  が成り立つような有理式  $f_i$  に対し、 $f'_i = f'_i(Q + S)/f_i(S)$  となる  $f'_i \in \mathbb{F}_{q^k}$  を求める。この  $P_i, f'_i$  を用いることで、楕円加算および直線  $g_{P, -P_i}$ , 垂線  $g_{P_i, O}$  に関する演算を削減する。上記アルゴリズムの説明では、簡単にするために  $w|l$  を仮定している。Miller のアルゴリズムでは  $n$  のハミング重み  $wH(n)$  回の楕円加算を行っていたが、window Miller アルゴリズムでは、 $l/w$  回行うことになるため、適切な  $w$  を用いれば、楕円加算および直線  $g_{P, -P_i}$ , 垂線  $g_{P_i, O}$  の計算を削減することができる。

|  |
|--|
| Input: $n, P \in E(\mathbb{F}_q)[n], Q \in E(\mathbb{F}_{q^k}) S \in E(\mathbb{F}_{q^k})$<br>Output: $f \in \mathbb{F}_{q^k}$  |
| (online computation)<br>1: $P_1 = P, f'_1 = 1$<br>2: for $i \leftarrow 2$ up to do $2^w - 1$<br>3: $P_i \leftarrow P + P_{i-1}$<br>4: $f'_i \leftarrow f'_{i-1} \cdot \frac{g_{P, -P_i}(S)g_{\mathcal{O}, P_i}(Q+S)}{g_{P, -P_i}(S)g_{\mathcal{O}, P_i}(Q+S)}$<br>(main computation)<br>5: $V \leftarrow P, f \leftarrow 1$<br>6: $n = \sum_{i=0}^{l-1} n_i 2^i, n_i \in \{0, 1\}, n_0 = 1$<br>7: for $n-1 \leftarrow i$ down to 0 step w<br>8: step 8-1 から 8-2 を w 回繰り返す<br>8-1: $V \leftarrow 2V$<br>8-2: $f \leftarrow f^2 \cdot \frac{g_{V, -2V}(Q+S)g_{\mathcal{O}, 2V}(S)}{g_{V, -2V}(Q+S)g_{\mathcal{O}, 2V}(S)}$<br>9: $n' \leftarrow \sum_i^{j=i-w+1} n_j 2^{j-i+w-1}$<br>10: if $n' \neq 0$ then<br>10-1: $f \leftarrow f f'_m \cdot \frac{g_{V, P_m}(Q+S)g_{\mathcal{O}, V+P_m}(S)}{g_{V, P_m}(Q+S)g_{\mathcal{O}, V+P_m}(S)}$<br>10-2: $V \leftarrow V + P_{n'}$<br>11: return $f$ |

表 5.3 Window Miller Algorithm

## 第 6 章

# 提案手法

### 6.1 提案手法のアルゴリズム

BKLS Algorithm, Window Miller Algorithm を組みわせることで新しい高速化手法を提案する。以下にそのアルゴリズムを示す。

|  |
|--|
| Input: $n, P, Q \in E(\mathbb{F}_q)[n]$                      |
| Output: $f \in \mathbb{F}_{q^k}$                             |
| (online computation)   |
| 1: $P_1 = P, f'_1 = 1$                                       |
| 2: for $i \leftarrow 2$ up to $2^w - 1$                      |
| 3: $P_i \leftarrow P + P_{i-1}$                              |
| 4: $f'_i \leftarrow f'_{i-1} \cdot g_{P_i, P}(\psi(Q))$      |
| (main computation)   |
| 5: $V \leftarrow P, f \leftarrow 1$                          |
| 6: $n = \sum_{i=0}^{l-1} n_i 2^i, n_i \in \{0, 1\}, n_0 = 1$ |
| 7: for $n - 1 \leftarrow i$ down to 0 step w                 |
| 8: step 8-1 から 8-2 を w 回繰り返す                                 |
| 8-1: $V \leftarrow 2V$                                       |
| 8-2: $f \leftarrow f^2 \cdot g_{V, V}(\psi(Q))$              |
| 9: $n' \leftarrow \sum_i^{j=i-w+1} n_j 2^{j-i+w-1}$          |
| 10: if $n' \neq 0$ then                                      |
| 10-1: $f \leftarrow f f'_m \cdot g_{V, P_m}(\psi(Q))$        |
| 10-2: $V \leftarrow V + P_{n'}$                              |
| 11: return $f$   |

表 6.1 Window 方を用いた BKLS Algorithm



## 6.2 提案手法の計算量

Miller Algorithm, BKLS Algorithm, Window Miller Algorithm, 提案手法のアルゴリズムの計算量を比較する。

Miller Algorithm における演算部分のステップを加算 (TADD), 2 倍算 (TDBL), BKLS Algorithm における演算部分のステップを加算 (BKLS-TADD), 2 倍算 (BKLS-TDBL) に分ける. [4] によると、それぞれの計算コストは次のようになっている.

| 演算        | 計算コスト                          |
|-----------|--------------------------------|
| TADD      | $4M_k + (6k + 10)M + 4S$       |
| TDBL      | $4M_k + 2S_k + (6k + 7)M + 7S$ |
| BKLS-TADD | $M_k + (3k + 10)M + 3S$        |
| BKLS-TDBL | $M_k + S_k + (3k + 7)M + 6S$   |

表 6.2 各演算の計算コスト

ただし、 $\mathbb{F}_q$  上の乗算、自乗の演算量を  $M, S$ ,  $\mathbb{F}_{q^k}$  上の乗算、自乗の演算量を  $M_k, S_k$  であらわしている。 $k$  が比較的小さい値 (supersingular な楕円曲線の場合、 $k = 2, 3, 6$ ) であることを考えると、ほぼ  $M_k = k^2 M$  と考えられる。ここで、 $S = M$  とすると

| 演算        | 計算コスト               |
|-----------|---------------------|
| TADD      | $(4k^2 + 6k + 14)M$ |
| TDBL      | $(6k^2 + 6k + 14)M$ |
| BKLS-TADD | $(k^2 + 3k + 13)M$  |
| BKLS-TDBL | $(2k^2 + 3k + 13)M$ |

表 6.3 計算量 (近似)

$n$  を  $l$  ビットのランダムな数とし、これらを用いて Miller Algorithm, BKLS Algorithm, Window Miller Algorithm, 提案手法のアルゴリズムの計算量は次のように表される。

| Algorithm               | 計算コスト   |
|-------------------------|---|
| Miller Algorithm        | $l \times \text{TDBL} + (1/2) \times \text{TADD}$                     |
| Window Miller Algorithm | $l \times \text{TDBL} + (1/w + 2^w - 2) \times \text{TADD}$           |
| BKLS Algorithm          | $l \times \text{BKLS-TDBL} + (1/2) \times \text{BKLS-TADD}$           |
| 提案手法                    | $l \times \text{BKLS-TDBL} + (1/w + 2^w - 2) \times \text{BKLS-TADD}$ |

表 6.4 各手法の計算コスト

## 第 7 章

# 評価と考察

### 7.1 評価

$l \approx 80, 120, 160, k=2, 4$ それぞれの場合に応じて, 従来手法と提案手法の計算コストと計算時間を次の表に示す.

| $k = 2, l = 80, w = 4$  | 計算コスト |
|-------------------------|-------|
| Miller Algorithm        | 5980M |
| Window Miller Algorithm | 5248M |
| BKLS Algorithm          | 3080M |
| 提案手法                    | 2942M |

表 7.1 各手法の計算コスト

| $k = 6, l = 80, w = 4$  | 計算コスト  |
|-------------------------|--------|
| Miller Algorithm        | 29040M |
| Window Miller Algorithm | 27876M |
| BKLS Algorithm          | 10920M |
| 提案手法                    | 10518M |

表 7.2 各手法の計算コスト

| $k = 2, l = 120, w = 4$ | 計算コスト |
|-------------------------|-------|
| Miller Algorithm        | 8520M |
| Window Miller Algorithm | 7848M |
| BKLS Algorithm          | 4620M |
| 提案手法                    | 4250M |

表 7.3 各手法の計算コスト

| $k = 6, l = 120, w = 4$ | 計算コスト  |
|-------------------------|--------|
| Miller Algorithm        | 43560M |
| Window Miller Algorithm | 40456M |
| BKLS Algorithm          | 16380M |
| 提案手法                    | 15308M |

表 7.4 各手法の計算コスト

| $k = 2, l = 160, w = 4$ | 計算コスト  |
|-------------------------|--------|
| Miller Algorithm        | 11360M |
| Window Miller Algorithm | 10268M |
| BKLS Algorithm          | 6160M  |
| 提案手法                    | 5562M  |

表 7.5 各手法の計算コスト

| $k = 6, l = 160, w = 4$ | 計算コスト  |
|-------------------------|--------|
| Miller Algorithm        | 58080M |
| Window Miller Algorithm | 53036M |
| BKLS Algorithm          | 21840M |
| 提案手法                    | 20098M |

表 7.6 各手法の計算コスト

## 7.2 考察

提案手法は従来の BKLS Algorithm に比べて表 7.1, 表 7.2 から 80bit では 4.5%, 3.7%, 表 7.3, 表 7.4 から 120bit では 8.0%, 6.5%, 表 7.5, 表 7.6 から 160bit では 9.7%, 8% の高速化を実現できる。ペアリングで使用する 2 つの点の位数  $n$  が大きいほど、高速化できる割合は大きく、埋め込み次数  $k$  が大きいほど高速化できる割合は小さくなることから分かる。また、今回は  $w$  の値を動かさなかったが、 $w$  の値の設定によって異なる結果が出るのが予想される。

位数  $n$  が小さく、埋め込み字数  $k$  が大きいほど、今回提案した手法のメリットを享受できなくなる。提案した手法は、事前演算を必要とするため、場合によって使い分ける必要がある。

## 7.3 今後の課題

今後の課題としては、既存の Signed Miller Algorithm や、3 倍算の計算コストが乗算よりも小さいことを利用した標数 3 の有限体における Miller Algorithm に distortion map を適用した手法に今回の提案に使用した Window Miller Algorithm を組み合わせるなどが考えられる。また、今回の手法を適用するのに適した楕円曲線を探すなどが挙げられる。

# 謝辞

本研究を進めるにあたり，適切な御指導，御助言，御検討を頂いた中央大学 理工学部 趙 晋輝 教授並びに，諸先輩方に深く感謝いたします。また，日ごろの学生生活でお世話になった中央大学 趙研究室の皆様にも深く感謝いたします。

## 参考文献

- [1] P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott: *Efficient algorithms for pairing-based cryptosystems*, Advances in Cryptology-Crypto'2002, LNCS 2442, pp. 354-368. Springer-Verlag, 2002.
- [2] P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott: *Efficient implementation of pairing-based cryptosystem*, Journal of Cryptology, 17(4):321-334, 2004.
- [3] V. S. Miller: *The Weil Pairing, and Its Efficient Calculation*. Journal of CRYPTOLOGY, vol.17, No. 4, pp.235-261, September, 2004.
- [4] T. Izu and T. Takagi, "Efficient Computations of the Tate Pairing for the Large MOV Degrees" ICISC 2002, LNCS 2587, pp.283-297, Springer-Verlag, 2003.
- [5] 小林 鉄太郎, 斉藤 泰一, 今井 秀樹: "Pairing 演算の高速化 Fast Computation of Pairing" The 2004 Symposium on Cryptography and Information Security Sendai, Japan, Jan.27-30, 2004
- [6] T. Kobayashi, K. Aoki, H. Imai: *Efficient algorithms for Tate pairing*. IEICE Trans. Fundamentals, VOL.E89-A, NO.1 Jan 2006
- [7] T. Izu and T. Takagi: *Efficient computations of the Tate pairing for the large MOV degrees*. ICISC f02, LNCS 2587, pp.283-297. Springer-Verlag, 2003
- [8] 大川一樹 :Double-Base Chains を用いたペアリング 暗号における Tate ペアリングの高速化に関する考察