卒業研究論文

Double-Base Chains

Fast Computation of Tate Pairing with Double-Base Chains

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 BKLS Algorithm, Duursma and Lee Algorithm が知られている. 本研究では 2 進展開,3 進展開を用いた Double-Base Chains に Miller Algorithm を適用した埋め込み次数 k = 4, 6, 8 のときの計算コストを求めた既存研究 [7] に対して, 埋め込み次数 k = 2 のときに適用した. また, 既存研究では計算コストまでしか求められていなかったが, 提案手法では k = 2, 4, 6 の楕円曲線に対して MAGMA 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　Duursma and Lee Algorithm　19

第 6 章　既存研究と提案手法　20
6.1　Double-Base Chains　20
6.2　Double-Base Chains を用いた Miller Algorithm　21
6.3　提案手法　23

第 7 章　実行結果　24
7.1　実行結果　24

第 8 章　考察と今後の課題　27

8.1 考察 27
8.2 今後の課題 28

謝辞 29

参考文献 30

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となる。このため,Miller のアルゴリズムの高速実装法の研究が盛んに行われている。これらの既存研究として BKLS Algorithm に加え,window 法と呼ばれれる, 楕円曲線上のスカラ倍演算を高速化させる手法と 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,BKLSAlgorithm に window 法を適用した。以下に本論文の構成を示す. 第 2 章で数学的準備, 第 3 章では楕円曲線について, 点の加算法や楕円離散対数問題について述べる. 第 4 章でペアリングについて, 第 5 章でペアリングの計算方法について述べる. 第 6 章で従来手法, 既存研究, 提案手法について, 第 7 章で実行結果, 第 8 章で考察と今後の課題を述べる.

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

環 R の中のイデアル 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. イデアル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 $\{f_1, \cdots, f_l\} \subset I$ によって生成されるなら, $I = \sum_{i=1}^{l} Rf_i$, または単に $I = (f_1, \cdots, f_l)$ 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, $a \equiv b \bmod n$

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,

$$[x]+[y]=[x+y]\ ,\quad [x]\cdot[y]=[xy] \tag{2.1}$$

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,

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

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,

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

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

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,

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

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，$\mathrm{char}(\mathbb{F})=0$ 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，$1+1+\cdots+1(p\text{ 回})$ 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，$\mathrm{char}(\mathbb{F})=p$ 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．元の数が 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 α 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 α が根である monic 既約多項式が一意に存在する．そして α 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，α の最小多項式と呼ばれる．もし，α 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 α のべきと $\mathbb{F}$ の元に関する任意の有理式は，べき $1,\alpha,\alpha^2,\cdots,\alpha^{d-1}$ の 1 次結合として表現できる．
したがって，これらの α のべきは $\mathbb{F}$ 上における $\mathbb{F}(\alpha)$ の基底をなす．よって，α を付加することにより得られる拡大次数は α 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 $\mathbb{Z}_p^* = \{1, 2, \cdots, p-1\}$ 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, $\{0, 1, \cdots, p-2\}$ 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, $r \in \{0, 1, \cdots, p-2\}$ 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，通常の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 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.　$P \neq Q, \quad x_1 \neq x_2 \longrightarrow$ 図 3.1
Case 2.　$P = Q \longrightarrow$ 図 3.2

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

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

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

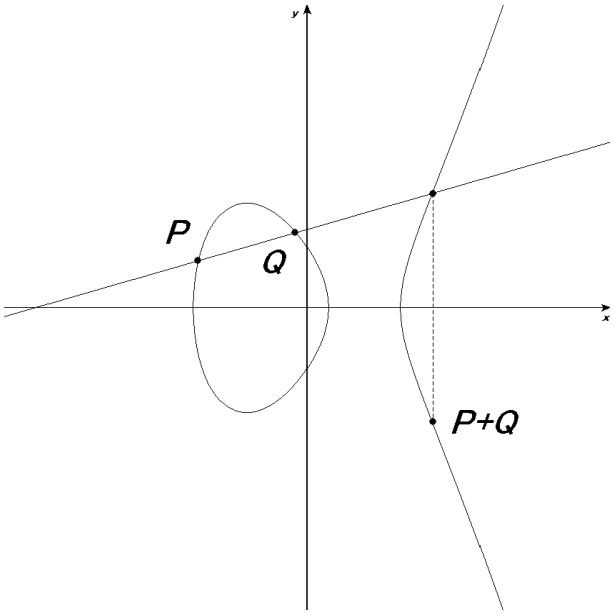

図 3.1 $P + Q$

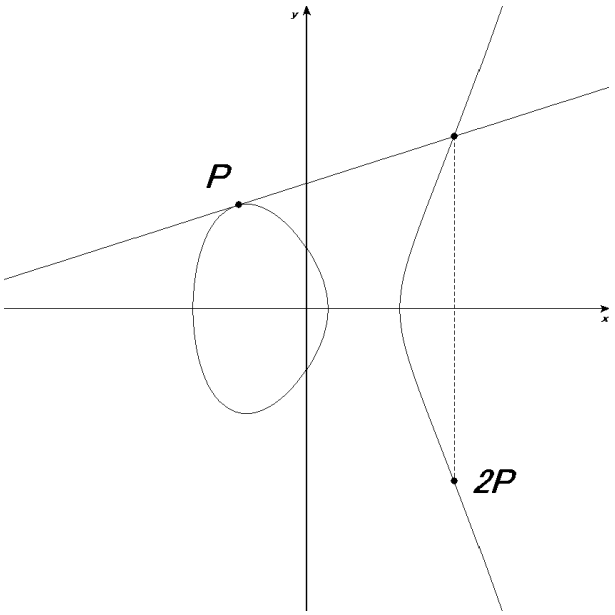

図 3.2 $2P$

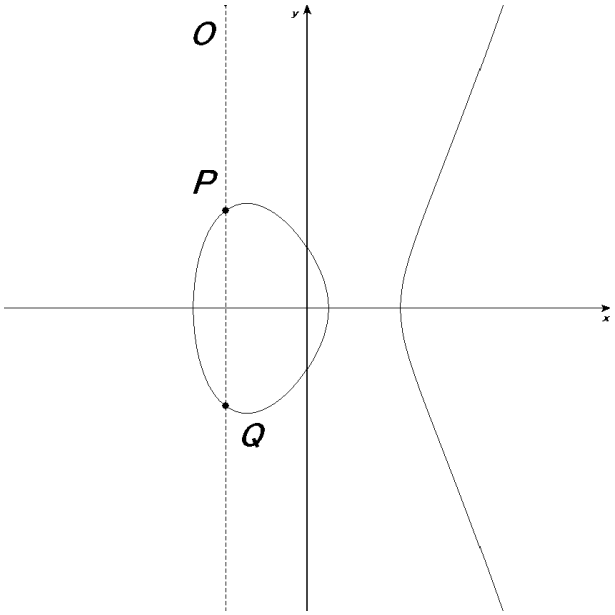

図 3.3 $P + Q = \mathcal{O}$

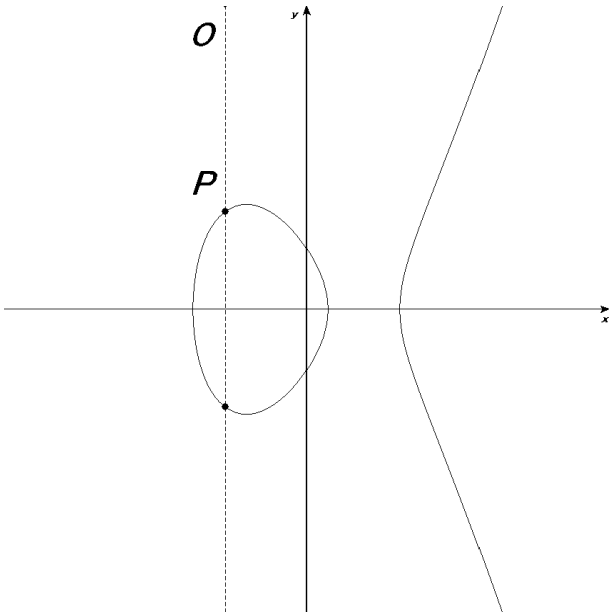

図 3.4 $P + \mathcal{O} = P$

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 離散対数問題 (DLP)

p を素数とし，g を $\mathbb{Z}_p^*$ の生成元とする．このとき任意の $y \in \mathbb{Z}_p^*$ に対し，

$$y \equiv g^x \pmod{p}$$

となる x が必ず存在する．このような x を y の離散対数という．y と g が与えられたとき，$y \equiv g^x \pmod{p}$ を満たす x，すなわち $x \equiv \log_g y \pmod{p}$ を求める問題を離散対数問題という．これは，$y \equiv g^x \pmod{p}$ を計算するのは容易だが，$x \equiv \log_g y \pmod{p}$ を求めることは非常に困難であることに基づいている．

3.3.2 楕円離散対数問題 (ECDLP)

任意の点 $P \in E(\mathbb{F}_q)$ に対して，$\langle P \rangle = \{\mathcal{O}, P, 2P, 3P, \ldots\}$ 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') 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 $Q \in G_2$ $(P \neq 0)$ 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 $\mathrm{Div}_{\overline{K}}(C)$ で表され，加法に関して群構造をなす．divisor D の台とは，$\mathrm{supp}(D) = \langle P \in E | n_P \neq 0 \rangle$ 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，$\mathrm{sum}(D) = \sum n_P P$ 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 $\mathrm{ord}_P(f)$ を数えることができる．$\mathrm{ord}_P(f)$ 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，$\mathrm{ord}_P(f)$ 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，

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

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

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

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，$\#E(K_0)$ で割り切られ，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 ϕ 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，α, β 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．

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

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

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

で定義される．μ_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，$\mathrm{div}(f) = n(T) - n(\mathcal{O})$ 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，$\mathrm{div}(g) = \sigma_{R \in E[n]}((T' + R) - (R))$ 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，$\mathrm{div}(g) = \sigma_{nT''=T}(T'') - \sigma_{nR=\mathcal{O}}(R)$ 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，$\mathrm{div}(f \circ n) = n(\sigma_R(T' + R)) - n(\sigma_R(R)) = \mathrm{div}(g^n)$ 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 $\phi : E \to E'$ 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 $\mathrm{div}(f_{n,P}) = n(P) - n(\mathcal{O})$ 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 $D \in \mathrm{Div}^0(E)$ を選択する．$\mathrm{supp}(\mathrm{div}(f_{n,P})) \cap \mathrm{supp}(D) = \emptyset$ 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．

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

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.

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

とする.

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

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 $\sigma \in \mathrm{Gal}(\overline{\mathbb{F}}_{q^k}/\mathbb{F}_q)$ 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[12] 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.

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

ここで f_h を

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

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

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

は次のように証明される.

Proof. 直線の divisor の公式

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

より，

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

となる．一方 f_h の定義から

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

が得られ，両者は等しくなる．
従って

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

これより，再帰的公式

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

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．

Input: $n,\ P \in E(\mathbb{F}_q)[n],\ Q \in E(\mathbb{F}_{q^k})$	
Output: $f \in \mathbb{F}_{q^k}$	
1:	$Q' \in_R E(\mathbb{F}_{q^k})$
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 do 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 $\log_2(n)$ の繰り返しなので，二重の操作は $\log_2(n)$ 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順同型写像である．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 ψ 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 ψ を用いて, $\hat{e} : G_1 \times G_1 \to G_3$ を

$$\hat{e}(P, Q) = \hat{e}(P, \psi(Q))$$

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

$$\hat{e}(P, P) = \hat{e}(P, \psi(P)) \neq 1$$

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 do 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 $n|q^k - 1,\ n \not| q^d - 1$ 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き, 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 $x \in \mathbb{F}_{q^{k/2}}$となるとき, 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 ψ 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 ψ_d を用いる.

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

表 5.2 BKLS Algorithm

さらに, $\mathbb{F}_{3^m}$ の有限体上で定義される supersingular curve $y^2 = x^3 - x + b,\ b \in \{-1, 1\}$ 上の 3 倍算は, $P = (x, y),\ 3P = (x_3, y_3)$ とすれば,

$$x_3 = (x^3)^3 - b$$

$$y_3 = -(y^3)^3$$

で計算可能である. 標数 3 の有限体における 3 乗算の計算コストは乗算と比較して小さく, 効率よく 3 倍算を求めることができる. 部分群の位数 n を 3 進展開し, 3 倍算に対し Miller fomula を求めることにより, Miller Algorithm の計算が可能となる. また distortion map ψ により, 分母消去の手法が適用可能である. 次にそのアルゴリズムを示す.

Input: $P,\ Q \in E(\mathbb{F}_{3^m})[n]$
Output: $f \in \mathbb{F}_{3^{6m}}$
$f \leftarrow 1,\ V \leftarrow P$
$n = \sum_{l-1}^{i=0} n_i 3^i,\ n_i \in \{-1, 0, 1\}$
for $j \leftarrow l-1$ down 0 do 0
$\quad f \leftarrow f^3 \cdot g_{V,\ -3V}(\psi(Q))$
$\quad V \leftarrow 3V$
if $n_j = 1$ then
$\quad f \leftarrow f \cdot g_{V,\ P}(\psi(Q))$
$\quad V \leftarrow V + P$
if $n_j = -1$ then
$\quad f \leftarrow f \cdot g_{V,\ -P}(\psi(Q))$
$\quad V \leftarrow V - P$
return f

表 5.3 BKLS Algorithm supersingular curve on $\mathbb{F}_{3^m}$

5.4 Duursma and Lee Algorithm

Duursma らは有限体 $\mathbb{F}_{p^m}(p \geq 3,\ \gcd(m,\ 2p) = 1)$ 上で定義された種数 $(p-1)/2$ の超特異代数曲線 $C : y^2 = x^p - x + d$ における Tate ペアリングの演算アルゴリズム [5] を提案した．特に $p = 3$ のとき，標数 3 の有限体における埋め込み次数 $k = 6$ を持つ supersingular curve に対する演算アルゴリズムとなる．Reduced Tate ペアリングの性質より，群位数 n を基底 p において Hamming Weight が 2 となる値 $p^{mp} + 1$ と置換，最終べき乗を Frobenius 写像と 1 回の逆元で計算可能な値 $(p^{2mp} - 1) = (p^{mp} + 1) = (p^{mp} - 1)$ と置換する．この手法により，Miller Algorithm の処理及び最終べき乗が著しく簡略化され，さらに mp 回のループ回数を m 回への削減も実現した．

超楕円曲線 $C,\ d = \pm 1,\ p \equiv 3(\mathrm{mod}4)$ を $C/\mathbb{F}_{p^m}$ とする．Duursma and Lee Algorithm では，点を通る直線 g を導出して Miller formula から導出せず，ある因子を持つ関数より値を計算する．$P \in C(\mathbb{F}_{p^m})$ とすると，因子 $(p^{pm} + 1)((P) - (\mathcal{O}))$ は主因子となる．すなわち，$\mathrm{div}(f_{p^{pm}+1,P}) = (p^{pm} + 1)(P) - ((p^{pm} + 1)P) - (p^{pm})(\mathcal{O})$ となる関数 $f_{p^{pm}+1,P}$ を求める．

次に標数 3 の有限体上で定義される supersingular curve における Tate ペアリングを求める Duursma and Lee Algorithm を示す．

Input: $E(\mathbb{F}_{3^m}) :\ y^2 = x^3 - x + d,\ P = (\alpha,\ \beta),\ Q \in E(\mathbb{F}_{3^m}[n])$
Output: $f_{3^{3m+1},\ P}(\psi(Q)) \in \mathbb{F}_{3^{6m}}$
1: $f \leftarrow 1,\ V \leftarrow P$
2: $r = \sum_{l-1}^{i=0} n_i 2^i,\ r_i \in \{0, 1\}$
3: for $j \leftarrow l - 1$ down 0 do 0
4: $\alpha \leftarrow \alpha^3,\ \beta \leftarrow \beta^3$
5: $g \leftarrow \beta y \hat{\sigma} - (\alpha + x - \rho + d)^{(p+1)/2}$
6: $f \leftarrow f \cdot g$
7: $x \leftarrow x^{1/3},\ y \leftarrow y^{1/3}$
8: return f

表 5.4 Duursma and Lee Algorithm

第 6 章

既存研究と提案手法

6.1 Double-Base Chains

楕円曲線 $E(\mathbb{F}_q)$ 上の k 倍点を求めるために 2 進展開, 3 進展開, NAF, window 法, フロベニウス展開などがある. Double-Base Number System は V. S. Dimitrov らによって提案された手法 [15] で, 整数 k は 2, 3 のべき乗を使って次のように表すことができる.

$$k = \sum_{i=1}^{m} s_i 2^{b_i} 3^{t_i},\ s_i \in_{\{} -1, 1\},\ b_i,\ t_i \geq 0.$$

このように k を展開する方法を DBNS 展開と呼び [15] , で指数計算の手法として提案された. この手法を楕円曲線 $E(\mathbb{F}_q)$ 上の k 倍点を求めるために適用する. この手法を Double-Base Chains [3] と呼び, 以下にそのアルゴリズムを示す.

Input: $k,\ b_{max},\ t_{max} > 0$	
Output: $k = \sum_{i=1}^{m} s_i 2^{b_i} 3^{t_i}$ となるような集合 $(s_i, b_i, t_i),\ b_1 \geq \cdots \geq b_m \geq 0,\ t_1 \geq \cdots \geq t_m \geq 0$	
1.	$s \leftarrow 1$
2.	while $k > 0$ do
3.	k に最も近似した値 $z = 2^b 3^t$ を定義する. $0 \leq b \leq b_{max},\ 0 \leq t \leq t_{max}$
4.	print (s, b, t)
5.	$b_{max} \leftarrow b,\ t_{max} \leftarrow t$
6.	if $k < z$ then
7.	$s \leftarrow -s$
8.	$k \leftarrow \lvert k - z \rvert$

表 6.1 DBNS representation algorithm(Double-Base Chains)

このとき, 2 進展開と 3 進展開の上限をそれぞれ $b_{max},\ t_{max}$ とし, m はループ回数を表す. これらの値は $b_{max} < \log_2(k) < n,\ t_{max} < \log_3(k) \approx 0.63n$ となる. このアルゴリズムの複雑性は Step 3 に依存する. しかし, これは事前計算した値のテーブルを使うことで容易に解決できる. さらにここで詳しくは述べないが, Double-Base Chains は暗号に対する攻撃手法の一つであるサイドチャネル攻撃に耐性がある.

6.2 Double-Base Chains を用いた Miller Algorithm

Miller Algorithm, Signed Miller Algorithm でそれぞれ部分群の位数 n に対して 2 進展開, 符号付き 2 進展開が使われていた. 次に C. Zhao らによって提案された Double-Base Chains を用いた Miller Algorithm[7] を次に示す.

Input: $n,\ P = (x_P, y_P) \in E(\mathbb{F}_q)[n],\ Q = (x_Q, y_Q) \in E(\mathbb{F}_{q^k})[n]$
Output: $e_n(P, Q)$

1. $T \leftarrow P,\ f_{-1} = \frac{1}{x_Q - x_P}$
2. if $s_1 = -1$ then 3. $f_1 \leftarrow f_{-1}$
4. $n = \sum_{i=1}^{m} s_i 2^{b_i} 3^{t_i},\ s_i \in \{-1, 1\},\ b_1 \geq b_2 \geq \cdots \geq b_m \geq 0,\ t_1 \geq t_2 \geq \cdots \geq t_m \geq 0$
5. for $i = 1, \cdots, m-1$ do
6. $u \leftarrow b_i - b_{i+1},\ v \leftarrow t_i - t_{i+1}$
7. if $u = 0$ then
8. for $j = 1, \cdots, v$ do
9. $f_1 \leftarrow f_1^3 \cdot \frac{g_{T,T}(Q) g_{T,2T}}{g_{2T}(Q) g_{3T}(Q)},\ T \leftarrow 3T$
10. else
11. if $v = 0$ then
12. for $j = 1, \cdots, u$ do
13. $f_1 \leftarrow f_1^2 \cdot \frac{g_{T,T}(Q)}{g_{2T}(Q)},\ T \leftarrow 2T$
14. else
15. for $j = 1, \cdots, u$ do
16. $f_1 \leftarrow f_1^2 \cdot \frac{g_{T,T}(Q)}{g_{2T}(Q)},\ T \leftarrow 2T$
17. for $j = 1, \cdots, v$ do
18. $f_1 \leftarrow f_1^3 \cdot \frac{g_{T,T}(Q) g_{T,2T}}{g_{2T}(Q) g_{3T}(Q)},\ T \leftarrow 3T$
19. if $s_{i+1} = 1$ then
20. $f_1 = f_1 \cdot \frac{g_{T,P}(Q)}{g_{T+P}(Q)},\ T \leftarrow T + P$
21. else
22. $f_1 = f_1 \cdot f_{-1} \cdot \frac{g_{T,-P}}{g_{T-P}},\ T \leftarrow T - P$
23. return f

表 6.2 Double-Base Chains を用いた Miller Algorithm

C. Zhao らは埋め込み次数 $k = 4, 6, 8$ のとき, 上記の手法と Miller Algorithm, Signed Miller Algorithm の計算コストを比較した. 計算コストを求めるにあたって C. Zhao らは [9] を参照した. 計算コストは $\mathbb{F}_q^*$ 上では乗算一回の演算時間を $1M$, 2 乗を $1S$, 除算を $1I$ とする. $\mathbb{F}_{q^k}^*$ では乗算一回の演算時間を $1M_k$, 2 乗を $1S_k$, 除算を $1I_k$ とし, $\mathbb{F}_q^*$と $\mathbb{F}_{q^k}^*$ の要素の乗算一回の演算時間を $1M_b$ とする. 同一の演算, スカラー倍の演算は時間がかからないものとする. また, $\mathbb{F}_q^*$, $\mathbb{F}_{q^k}^*$ 上の加算, 減算演算時間は乗算のコストに比べて小さいので無視する. さらに, $S = 0.8M,\ I = 10M,\ M_k = k^{1.6}M,\ S_k = 0.8k^{1.6}M,\ I_k = 10k^{1.6}M,\ M_b = kM$ と換算できる. 楕円曲線 $E(\mathbb{F}_q)$ 上では点の加算 (ECADD), 2 倍算 (ECDBL), 3 倍算 (ECTRL) がある. 減算は加算と同様のコストである. 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
ECADD	$1I + 2M + 1S$
ECDBL	$1I + 2M + 2S$
ECTRL	$1I + 7M + 4S$

表 6.3　楕円曲線における各演算の計算コスト

Tate ペアリングにおける演算部分のステップを加算 (TADD), 減算 (TSUB), 2 倍算 (TDBL), 3 倍算 (TTRL) に分ける. 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	$M_k + 2.5M_b + 1I + 3M + 1S$
TSUB	$M_k + 1I + (2k + 3)M + 1S$
TDBL	$M_k + S_k + 3.5M_b + 1I + 4M + 2S$
TTRL	$3M_k + S_k + 2M_b + 1I + 9M + 4S$

表 6.4　Tate ペアリングにおける各演算の計算コスト

n の bit 数を l とし, これらを用いて Miller Algorithm, Signed Miller Algorithm, Double-Base Chains を用いた Miller Algorithm の計算コストは次のように表される.

	計算コスト
Miller Algorithm	$(3l-2)M_k + (2l-2)S_k + (\frac{9l-6}{2}k + 12l - 7)$ $+(15l-12)\frac{s}{2}$
Signed Miller Algorithm	$(\frac{4}{3}l + 5)M_k + (l-1)S_k + I_k + 7M_{\frac{k}{2}} + I_{\frac{k}{2}}$ $+(\frac{17l-14}{4}k + (5l-4)) + (\frac{7}{3}l - 2)s + (\frac{4}{3}l - 1)I$
Double-Base Chains を用いた Miller Algorithm	$(b_{max} + 3t_{max} + m + 6)M_k + (b_{max} + t_{max} + 1)S_k$ $+(\frac{7}{2}b_{max} + 2t_{max} + \frac{5}{4}m)M_b + (b_{max} + t_{max} + m)I$ $+(4b_{max} + 9t_{max} + (k+3)m) + (2b_{max} + 4t_{max} + m)s$ $+I_k + 7M_{\frac{k}{2}} + I_{\frac{k}{2}}$

表 6.5　各手法の計算コスト

次に C. Zhao らによって示されたこの手法の $k = 4, 6, 8$ のときの計算コストと従来法を $b_{max} = 76,\ t_{max} = 53,\ m = 38$ のときの計算コストで比較したものを表にした. 表の計算コストは乗算に換算している.

b_{max}	t_{max}	m	cost($k = 4$)	cost($k = 6$)	cost($k = 8$)
57	65	45	8287	12744	17222
76	53	38	8350	12554	17085
95	41	37	8395	12552	17052
103	36	39	8493	12676	17186

表 6.6　$k = 4,\ 6,\ 8$ における Double-Base Chains を用いた Miller Algorithm の計算コスト

Algorithm	cost($k = 4$)	cost($k = 6$)	cost($k = 8$)
Miller Algorihtm	12328	20353	28379
Signed Miller Algorithm	9196	13685	18121
Double-Base Chains を用いた Miller Algorithm	8350	12554	17085

表 6.7　$k = 4,\ 6,\ 8$ における各手法の計算コスト

6.3　提案手法

埋め込み次数 $k = 2$ について, Miller Algorithm, Signed Miller Algorithm, Double-Base Chains を用いた Miller Algorithm の計算コストを算出する. そして提案手法である $k = 2$ に加えて既存研究である $k = 4,\ 6$ の場合を MAGMA で実装し, 計算時間を比較した. さらに考察では結果を踏まえた上で, Double-Base Chains を用いた Miller Algorithm の新たな高速化手法をいくつか提案する.

第 7 章

実行結果

7.1 実行結果

7.1.1 実装環境

プロセッサ: Intel(R) Xeon(TM) CPU 3.40GHz
メモリ: 2046MB RAM
OS: Microsoft Windows XP Professional
プログラミング言語: Magma ver.2.14-6

7.1.2 実装条件

部分群の位数 n はランダムに選ぶ.
計測は演算部分だけに対して 1000 回行い, その平均を取った.
楕円曲線は埋め込み次数 $k = 2,\ 4,\ 6$ で ordinary curve を使用した.
実験で使用した曲線は以下の通り. $k = 4$ の曲線は [14], $k = 6$ の曲線は [13] の例題から引用した.

1. $k = 2$ のとき, $y^2 = x^3 + x$

2. $k = 4$ のとき $y^2 = x^3 - 3x + b,\ b = 29934005285652524841310254358516467837540725144 02$

3. $k = 6$ のとき $y^2 = x^3 - 3x + b,\ b = 10677826069392299816489746483691451748795469887 30$

7.1.3 実行結果

$\log n \approx 80, 120, 160$ のときの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.

82bit($b_{max} = 50,\ t_{max} = 20,\ m = 13$)	計算コスト	計算時間
Miller Algorithm	3329	12.9ms
Signed Miller Algorithm	2906	11.3ms
Double-Base Chains を用いた Miller Algorithm	2487	9.6ms

表 7.1　$k = 2,\ \log n \approx 80$

120bit($b_{max} = 68,\ t_{max} = 33,\ m = 13$)	計算コスト	計算時間
Miller Algorithm	4885	21.1ms
Signed Miller Algorithm	4242	18.6ms
Double-Base Chains を用いた Miller Algorithm	3483	15.5ms

表 7.2　$k = 2,\ \log n \approx 120$

159bit($b_{max} = 97,\ t_{max} = 39,\ m = 17$)	計算コスト	計算時間
Miller Algorithm	6481	32.9ms
Signed Miller Algorithm	5614	29.2ms
Double-Base Chains を用いた Miller Algorithm	4565	25.1ms

表 7.3　$k = 2,\ \log n \approx 160$

161bit($b_{max} = 72,\ t_{max} = 56,\ m = 24$)	計算コスト	計算時間
Miller Algorithm	6563	43.8ms
Signed Miller Algorithm	5684	42.5ms
Double-Base Chains を用いた Miller Algorithm	4697	34.9ms

表 7.4　$k = 2,\ \log n \approx 160$

162bit($b_{max} = 113,\ t_{max} = 31,\ m = 45$)	計算コスト	計算時間
Miller Algorithm	12623	66.2ms
Signed Miller Algorithm	9273	64.0ms
Double-Base Chains を用いた Miller Algorithm	8796	59.0ms

表 7.5　$k = 4,\ \log n \approx 160$

159bit($b_{max}=80,\ t_{max}=49,\ m=36$)	計算コスト	計算時間
Miller Algorithm	19795	78.4ms
Signed Miller Algorithm	13289	77.8ms
Double-Base Chains を用いた Miller Algorithm	12205	64.8ms

表 7.6　$k=6,\ \log n \approx 160$

提案手法は従来手法に比べて表 7.1 から 82bit では 34%, 18%, 表 7.2 から 120bit では 36%, 20%, 表 7.3 から 159bit では 31%, 16%, 表 7.4 から 161bit では 25%, 22% の高速化を実現した. さらに既存研究を実装した結果、表 7.5 から $k=4$ で 160bit のとき, 12%, 8%, 表 7.6 から $k=6$ で 160bit のとき, 21%, 20% の高速化を実現した.

第 8 章

考察と今後の課題

8.1 考察

実行結果の表を二つに分類して考察する.

(1) 表 7.1, 7.2, 7.3 について考える.

計算コストから Miller Algorithm, Signed Miller Algorithm に比べて表 7.1 から 82bit では 34%, 17%, 表 7.2 から 120bit では 40%, 22%, 表 7.3 から 159bit では 42%, 23% の高速化が予想できるが, 実際に計算時間を比較すると計算コストに比べて多少遅かった. これは Double-Base Chains を用いる場合, TDBL, TTRL を行うが, 第 6 章で表にしたように TTRL のほうが計算コストが高い. 具体的に示すと TTRL $\approx$ 1.32TDBL となる. したがって, t_{max} の値は少なくとも $b_{max} > 1.3t_{max}$ であることが望ましいと思われる. 加えて各ステップで t_m の値が減らない場合, 計算が 2 倍算に比べて遅い 3 倍算が残ってしまうので, t_m より先に b_m が 0 になる場合, 計算コストと比較するとき, 計算時間が遅くなると予想される. 2 倍算に関しても遅くなると予想される. Miller Algorithm は left-to-right 法で 2 進展開を行っているが, Double-Base Chains を用いる場合ではそれができないので t_m が 0 になるときも, b_m がある程度少なくなることが望ましいと思われる.

(2) 表 7.4, 7.5, 7.6 について考える.

計算コストから表 7.4 から $k = 2$ のとき, 40%, 21%, 表 7.5 から $k = 4$ のとき, 43%, 5%, 表 7.6 から $k = 6$ のとき, 62%, 9% の高速化を予想しているが, Miller Algorithm で計算時間を比較すると計算コストを下回り, Sigend Miller Algorithm では計算コストを上回った. 主な原因としては前述の (1) が考えられる. 加えて Double-Base Chains を用いた Miller Algorithm 全体のループ回数とループ部分の演算の複雑性が原因だと思われる. [3] では n の値を Double-Base Chain に適した最適な値を選ぶと仮定している. [7] では計算コストを求めるとき, [3] の結果を参照し, Miller Algorithm と同様に, 計算コストでもループ回数 m が考慮されている. しかし, ループ部分の計算の複雑性から全体のループ回数と合わせるとどれだけ負担になるかについて言及していないので, m の値が大きいと従来手法と比較するとき, ループ部分の演算の複雑性により, 計算コストと計算時間にかなりの差が出てしまうことが予想される. 加えて (1) でも述べたが, Double-Base Chain は left-to-right 法が出来ないので、計算速度が遅くなった. 実際 $k = 4$, 6 のときの Hamming Weight は計算コストとほぼ同じであったことから, このことが原因であることが推定される.

Signed Miller Algorithm について $k = 6$ で高速化が可能になったのは, 符号付き 2 進展開をした場合, ループ回数があまり減らなかったためである. 計算コストでは {-1,1} の数は $n/3$ 個としているが, それ以上の個数が出てしまったため, 計算時間に差が生じた.

8.2 今後の課題

考察から, 計算速度をさらに速くするためには各ステップの b_m, t_m(特に t_m) の値の減少の度合いを少なくすることとループ回数 m を小さくすることが言える. 現在, Double-Base Chains の高速化手法として木構造を用いた手法が提案されている. この手法は楕円曲線 $E(\mathbb{F}_q)$ 上の点の加算で使われていて, m をより小さくすることが可能である. したがって, この手法を適用すれば, さらに Double-Base Chains を用いた Miller Algorithm を高速化することが可能である.

[3] でも述べられているが, Double-Base Chains に最適な値という定義が明確ではないので, これを明確にすることで高速化することが可能である. 加えてそのような曲線がどれだけ存在するのか調べる必要がある.

今回は ordinary curve に対して Miller Algorithm に Double-Base Chains を用いたが, BKLS Algorithm に拡張することやペアリングの手法である Ate ペアリングへの適用することも可能である.

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，村木氏, 森山氏を始めとする諸先輩方に深く感謝いたします. また, 楕円曲線の生成に関して助言を頂いた中央大学 趙研究室の佐藤氏, 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 cryptosystmes*,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. Dimitrov, L. Imbert, and P.K.Mishra: *Efficient and secure elliptic curve point multiplication using double-base chains*. Advances in Cryptology-Asiacrypt'2005,LNCS 3788, pp.59-78. Springer-Verlag, 2005.

[4] D. Freeman, M. Scott, and E. Teske: *A taxonomy of pairing-friendly elliptic curves*. Preprint 2006, Available from http://eprint.iacr.org/2006/372.

[5] I. Duusma and H. S. Lee. *Tate pairing implementation for hyperelliptic curves* $y^2 = x^p - x + d$, Advances in Cryptology-Asiacrypt 2003, LNCS 2894, pp.111-123, 2003.

[6] S. Matsuda, N. Kanayama, F. Hess, and E. Okamoto. *Optimised versions of the Ate and twisted Ate pairings*. Appear to the 11th IMA International Conference on Cryptography and Coding, Available at http://eprint.iacr.org/2007/013.pdf

[7] C. Zhao, F. Zhang and J. Huang: *Efficient Tate Pairing Computation Using Double-Base Chains*. Science in China Series F: Information Sciences, Science in China Press, 18.06.2008, vol. 51, no. 8, pp. 1096-1105

[8] T. Kobayashi, K. Aoki, H. Imai: *Efficient algorithms for Tate pairing*. IEICE Trans. Fundamentals, VOL.E89-A, NO.1 Jan 2006

[9] 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

[10] Lawrence C. Washington: *ELLIPTIC CURVES NUMBER THEORY AND CRYPTOGRPHY SECOND EDITION*. CHAPMAN/HALL&CRC

[11] V. Miller: *Short Programs for Functions on Curves*. Unpubliched manuscript, 1986.

[12] V. S. Miller: *The Weil Pairing, and Its Efficient Calculation*. Journal of CRYPTOLOGY, vol.17, No. 4, pp.235-261, September, 2004.

[13] M. Scott and P.S.L.M. Barreto: *Generating more MNT elliptic curves*. Designs, Codes and Cryptography, 38:209.217, 2006.

[14] K. Karabina: *On prime-order elliptic curves with embedding degrees 3, 4 and 6*. M.Math. thesis, Univ. of Waterloo, Dept. of Combinatorics and Optimization, 2006.

[15] V. S. Dimitrov, G. A. Jullien, and W. C. Miller: *Theory and applications of the double-base number system*. IEEE Transactions on Computers, 48(10):1098.1106, Oct. 1999.