Computer Security Symposium 2014
22 - 24 October 2014

2 次曲線を用いたペアリング暗号演算

永井 善孝 †　　白勢 政明 ‡

† 公立はこだて未来大学
041-8655 北海道函館市亀田中野町 116-2
g2113019@fun.ac.jp

‡ 公立はこだて未来大学
041-8655 北海道函館市亀田中野町 116-2
shirase@fun.ac.jp

ペアリング演算は，楕円曲線加算/2 倍算公式，Miller のアルゴリズムによって計算される．加算/2 倍算公式は，楕円曲線 $E : y^2 = x^3 + Ax + B$ とある直線 L との交点から導かれる．最近では，楕円曲線 E と 2 次曲線 $C : y = ax^2 + bx + c$ との交点から導かれる 2 倍-加算公式が提案されており，Miller のアルゴリズムに適用出来ることが示されている．本稿は，ペアリング演算において，楕円曲線上のある点の x 座標を 0 にする座標変換や 2 倍-加算公式を Miller のアルゴリズムに適用することで計算コストを削減出来ることを示す．実際に，128 ビットの Ate ペアリングの計算において約 15～17% コスト削減となった．

Pairing cryptography arithmetic using quadratic curves

Yoshitaka Nagai†　　Masaaki Shirase‡

†Future University Hakodate
116-2, Kamedanakano-cho, Hakodate Hokkaido, 041-8655, JAPAN
g2113019@fun.ac.jp

‡Future University Hakodate
116-2, Kamedanakano-cho, Hakodate Hokkaido, 041-8655, JAPAN
shirase@fun.ac.jp

Abstract Pairing arithmetic is calculated using addition and duplication formulas on elliptic curve and Miller's algorithm. Addition and duplication formulas are derived from intersection of an elliptic curve $E : y^2 = x^3 + Ax + B$ with a line L. Duplication-addition formula for computing $2Q+P$, which derived from intersection of the elliptic curve E and a quadratic curve $C : y = ax^2+bx+c$, was recently proposed and can be applied to Miller's algorithm. This paper shows that the cost of pairing arithmetic is reduced using duplication-addition formula and coordinate transform so that x coordinate of Q is 0. In fact, the cost of Ate pairing arithmetic of 128-bit is reduced by from 15 to 17%.

1　はじめに

近年，最も普及している公開鍵暗号の RSA 暗号と同等の安全性をより短い鍵長で実現できる楕円曲線暗号の普及が進んでいる．実際に ETC やディジタルテレビ，Blu-ray Disk 等の認証や著作権保護といった身近な場面で楕円曲線暗号が広く利用されるようになっている．更に，楕円曲線上のペアリングと呼ばれる双線形写像を用いた暗号プロトコルの研究も近年盛んになっており，そのような暗号プロトコルとして ID ベース暗号 [2]，タイムリリース暗号 [3]，属性ベース暗号 [6] 等がある．

ペアリング計算の高速化手法としては，Millerのアルゴリズムのループ回数が削減されたペアリングを構成する方法があり，Ateペアリング[7]やR-ateペアリング[8]，Xateペアリング[10]等が提案されている．

また[13]では，2次曲線を用いた2倍-加算公式がMillerのアルゴリズムに適用できることが示唆された．しかしながら，この手法がペアリング計算の速度に貢献するかどうかは検証されていなかった．

本稿では，楕円曲線上のある点のx座標を0にする座標変換と共に2次曲線を使用した2倍-加算公式を適用によりMillerのアルゴリズムを改良し，それによりアフィン座標におけるペアリング計算のコストが削減されることを示す．また，pairing-friendly楕円曲線であるBN曲線[1]を使用したAteペアリング演算において，従来法と提案法による計算コストの比較を行う．

注意

本稿でアフィン座標を用いた理由はMillerのアルゴリズム改良の容易さが第一の理由であるが，$\mathbb{F}_p$での逆元計算コストが乗算コストの5.7倍以下となる計算機環境[1]での楕円曲線スカラー倍ではアフィン座標の採用が最良であることが知られており[9]，ペアリング計算においてもアフィン座標の使用は考察に値するためである．

2　数学的準備

2.1節では楕円曲線及び楕円曲線の加法，因子を説明する．2.2節と2.3節では，ペアリング，Millerのアルゴリズムをそれぞれ説明する．

2.1　楕円曲線

$\mathbb{F}_p$を素体$(p \geq 5)$とする．Weierstrass形式と呼ばれる3次式

$$\begin{aligned} &E : y^2 = x^3 + Ax + B, \\ &\quad (A, B \in \mathbb{F}_p,\ 4A^3 + 27B^2 \neq 0) \end{aligned} \tag{1}$$

で定義される曲線を($\mathbb{F}_p$上)楕円曲線Eという．また，座標変換等によって楕円曲線Eが

$$E' : y^2 = x^3 + A'x^2 + B'x + C', A', B', C' \in \mathbb{F}_p$$

のような形に変わったとしてもE'は楕円曲線である．Eの$\mathbb{F}_p$有理点の集合$E(\mathbb{F}_p)$は

$$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p : (x, y) \text{は}(1)\text{を満たす}\} \cup \{\mathcal{O}\}$$

と定義される．ここで$\mathcal{O}$は無限遠点である．

2.1.1　楕円曲線の加法

$E(\mathbb{F}_p)$の重要な性質として,任意の2点$P, Q \in E(\mathbb{F}_p)$に対して,第3の点$P + Q \in E(\mathbb{F}_p)$が定義でき,この$+$に対して$\mathcal{O}$を零元とする群をなすことがある[11]．その位数を$\#E(\mathbb{F}_p)$と表記する．$P = (x_1, y_1)$, $Q = (x_2, y_2) \in E(\mathbb{F}_p)$に対して，$P + Q = (x_3, y_3)$は以下の公式を使って計算できる．

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned}$$

ここで,λはPとQを結ぶ直線の傾きであり次のように定義される．

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if} \quad P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if} \quad P = Q \end{cases}$$

上記の公式において,$P \neq Q$の場合は加算公式,$P = Q$の場合は2倍算公式と呼ばれる．$P = (x_1, y_1)$に対して,Pの逆元は$-P = (x_1, -y_1)$である．

この加算の繰り返しにより,楕円曲線上の点Pと整数nに対してスカラー倍

$$nP = \underbrace{P + P + \cdots + P}_{n\text{個}}$$

が定義される．実際にスカラー倍を求めるには,バイナリ法(Algorithm 1)やその改良を用いることが一般的である．

2.1.2　楕円曲線の因子

Eを楕円曲線とする．形式和

$$D = \sum_{P \in E} n_P(P)$$

[1]著者の計算機環境はこの条件を満たしていた．(CPUはCore 2 Duo U9300 1.2GHz，有限体実装にGMP[5]を使用．)

Algorithm 1: バイナリ法 [4]

Input: $P \in E(\mathbb{F}_p), n = (n_{s-1}, \cdots, n_0)_2$
Output: nP

$Q \leftarrow \mathcal{O}$ **and** $i \leftarrow s-1$
while $i \geq 0$
  if $n_i = 0$ **then** $Q \leftarrow 2Q$ (加算)
  else $Q \leftarrow 2Q + P$ (2 倍算と加算)
  $i \leftarrow i - 1$
return Q

を E の因子という．ここで，$n_P \in \mathbb{Z}$ で，有限個の $P \in E$ を除いて $n_P = 0$ である．E の因子 D の全体の集合を $Div(E)$ と表す．x, y を変数とする有理式 $f(x,y)$ が $P \in E$ で n_P 位の零を持ち，$Q \in E$ で m_Q 位の極を持つとする．このとき，f に付随する因子 $div(f)$ を

$$div(f) = \sum_{P \in E} n_P(P) - \sum_{Q \in E} m_Q(Q)$$

と定義する．因子 D がある有理式に付随する因子の時，D を主因子という．任意の主因子 $div(f), div(g)$ に対して

$$\begin{aligned} div(f \cdot g) &= div(f) + div(g) && (2) \\ div(f^{-1}) &= -div(f) && (3) \end{aligned}$$

が成り立つ．

命題 1[11]
E を楕円曲線，$D = \sum n_P(P) \in Div(E)$ を E の因子とすると，

$$D \text{ が主因子} \Leftrightarrow \sum n_P = 0 \text{ かつ } \underbrace{\sum n_P P}_{\text{楕円曲線の加算}} = \mathcal{O}$$

である．

系 2
$f(x,y)$ が多項式ならば，$n_P \in \mathbb{N}$ に対して

$$div(f) = \sum n_P(P) - \left(\sum n_P\right)(\mathcal{O})$$

である．

系 3
$P \in E, m \in \mathbb{N}$ に対して

$$m(P) - (mP) - (m-1)(\mathcal{O})$$

は主因子である．

2.1.3 2 倍-加算 [13]

楕円曲線

$$E : y^2 = x^3 + Ax + B$$

と 2 次曲線

$$C : y = ax^2 + bx + c$$

が (アフィン部分において)4 点 P, Q, R, S で交わる時 $div(C) = (P) + (Q) + (R) + (S) - 4(\mathcal{O})$ となり，命題 1 より

$$P + Q + R + S = \mathcal{O}$$

が成り立つ．$R = Q$ の場合を考えると

$$P + 2Q = -S$$

となり，C の式から 2 倍-加算公式を導くことができる．楕円曲線 E と 2 次曲線 C が $P_1 = (x_1, y_1)$ を通り，$P_2 = (x_2, y_2)$ で接すると，$P_1 + 2P_2 = (x_4, y_4)$ は以下のようにして得られる．
(i) 以下のような a, b, c を変数とする連立方程式を構成し，それを解く．

$$\begin{cases} y_1 &= ax_1^2 + bx_1 + c \\ y_2 &= ax_2^2 + bx_2 + c \\ \dfrac{3x_2^2 + A}{2y_2} &= 2ax_2 + b \end{cases}$$

(ii) $x_1, x_2, y_1, y_2, a, b, c$ を使用して以下のように x_4, y_4 を計算する．

$$\begin{cases} x_4 &= \dfrac{1 - 2ab}{a^2} - x_1 - 2x_2 \\ y_4 &= -x_4(ax_4 + b) - c \end{cases}$$

表記 4
ここで，いくつかの多項式や関数の表記を定義する．

(i) $P, Q \in E$ に対する $l_{P,Q}(x,y)$:
$l_{P,Q}(x,y) = 0$ が

$$\begin{cases} P \text{ と } Q \text{ を通る直線} & P \neq Q \text{ の時} \\ P \text{ での } E \text{ の接線} & P = Q \text{ の時} \end{cases}$$

を表す．

(ii) $P \in E$ に対する $v_P(x,y)$:
$v_P(x,y) = 0$ が P を通る x 軸に対する垂線を表す．

(iii) $P, Q \in E$ に対する $C_{P,2\times Q}(x,y)$:
$C_{P,2\times Q}(x,y) = 0$ が P を通り Q で E と接する 2 次曲線を表す．

(iv) $P \in E, m \in \mathbb{N}$ に対する $f_{m,P}$:

$f_{m,P}$ は

$$div(f_{m,P}) = m(P) - (mP) - (m-1)(\mathcal{O})$$

を満たす．(系 3 よりこのような $f_{m,P}$ は存在する．)

系 5

表記 4 の (i),(ii),(iii) の $l_{P,Q}, v_P, C_{P,2\times Q}$ に対して，

$$\begin{aligned} div(l_{P,Q}) &= (P)+(Q)+(-P-Q)-3(\mathcal{O}) \\ div(v_P) &= (P)+(-P)-2(\mathcal{O}) \\ div(C_{P,2\times Q}) &= 2(Q)+(P)+(-2Q-P)-4(\mathcal{O}) \end{aligned}$$

となる．

2.2 ペアリング

l を素数，$\mathbb{G}_1, \mathbb{G}_2$ を位数 l の加群，$\mathbb{G}_3$ を位数 l の乗法群とする．写像

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$$

が双線形性

$$\begin{aligned} e(P_1 + P_2, Q) &= e(P_1, Q) \cdot e(P_2, Q) \\ e(P, Q_1 + Q_2) &= e(P, Q_1) \cdot e(P, Q_2) \end{aligned}$$

を満たすとき，e をペアリングという．

2.2.1 Ate ペアリング [7]

E を $\mathbb{F}_p$ 上の楕円曲線とする．但し位数 $\#E(\mathbb{F}_p)$ は素数 l であるとする．$t = p+1-l$(t をトレースという)，k を $l|(p^k-1)$ を満たす最小の正整数 (k を埋め込み次数という)，π_p を E の p 乗 Frobenius 写像とする．すると，Ate ペアリングは群として $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3) = (E[l] \cap Ker(\pi_p - [p]), E(\mathbb{F}_p), \mathbb{F}^*_{p^k})$ を用い，

$$e(P, Q) = f_{t-1,P}(Q)^{(p^k-1)/l}$$

により定義される．

2.2.2 Barret-Naehrig(BN) 曲線

Barreto と Naehrig[1] は,

$$p = p(z) = 36z^4 + 36z^3 + 24z^2 + 6z + 1$$

(z は整数) で与えられる素数に対して，$b \in \mathbb{F}_p^*$ をランダムに選ぶと，$\mathbb{F}_p$ 上の楕円曲線 $E_b : y^2 = x^3 + b$ は 1/6 の確率で位数 n が $n = n(z) = 36z^4 + 36z^3 + 18z^2 + 6z + 1$ となり，E_b は埋め込み次数が 12 を持つことを示した．従って，$p(z)$ と $n(z)$ が共に素数となる $z \in \mathbb{Z}$ を選び，$\#E_b(\mathbb{F}_p) = n$ となる b を見つければ，埋め込み次数 12 の pairing-friendly 楕円曲線が得られる．この方法で構成される楕円曲線は Barreto-Naehrig(BN) 曲線と呼ばれ，最も効率的に構成できる pairing-friendly 楕円曲線の一つである．

2.3 Miller のアルゴリズム

ペアリング計算には，ある整数 n に対して $f_{n,P}(Q)$ を計算しなければならなく，この計算は Miller のアルゴリズムを用いて計算される．

E を楕円曲線とし，$P \in E$ とする．すると，表記 4 で与えられる関数に対して

$$\begin{aligned} & div(f_{m,P} \cdot l_{P,mP}/v_{(m+1)P}) \\ & \quad = (m+1)(P) - ((m+1)P) - m(\mathcal{O}) \\ & div(f_{m,P}^2 \cdot l_{mP,mP}/v_{2mP}) \\ & \quad = 2m(P) - (2mP) - (2m-1)(\mathcal{O}) \end{aligned}$$

が成り立つ (系 5,(2),(3) を参照)．言い換えると，

$$f_{m+1,P} = f_{m,P} \cdot l_{P,mP}/v_{(m+1)P} \quad (4)$$

$$f_{2m,P} = f_{m,P}^2 \cdot l_{mP,mP}/v_{2mP} \quad (5)$$

となる．(4) と (5) に $(x, y) = Q$ を代入しても等号が成り立つ．

$$\begin{aligned} f_{m+1,P}(Q) &= f_{m,P}(Q) \cdot l_{P,mP}(Q)/v_{(m+1)P}(Q) \\ f_{2m,P}(Q) &= f_{m,P}^2(Q) \cdot l_{mP,mP}(Q)/v_{2mP}(Q) \end{aligned}$$

従って，$f_{m,P}(Q)$ が与えられている場合，$f_{m+1,P}(Q)$ や $f_{2m,P}(Q)$ を計算できる．整数 n に対して $f_{n,P}(Q)$ は Algorithm 2 のようにして計算できる．

Algorithm 2: Miller のアルゴリズム

Input: $P \in \mathbb{G}_1, Q \in \mathbb{G}_2, n = (n_{s-1}, \cdots, n_0)_2$
　　但し，$n_{s-1} = 1$

Output: $f_{n,P}(Q)$

$V \leftarrow P, f \leftarrow 1$ **and** $i \leftarrow s-2$
while $i \geq 0$
　if $n_i = 0$
　　$f \leftarrow f^2 \cdot \dfrac{l_{V,V}(Q)}{v_{2V}(Q)}, V \leftarrow 2V$
　if $n_i = 1$
　　$f \leftarrow f^2 \cdot \dfrac{l_{V,V}(Q)}{v_{2V}(Q)}, V \leftarrow 2V$
　　$f \leftarrow f \cdot \dfrac{l_{V,P}(Q)}{v_{V+P}(Q)}, V \leftarrow V + P$
return f

2.3.1 2 倍-加算を適用した Miller のアルゴリズム [13]

Miller のアルゴリズム (Algorithm 2) のステップ 6 と 7 を 2.1.3 節の 2 倍-加算を使用することでひとつの処理にまとめることができる．系 2 と系 5 より

$$div(f_{m,P}) = m(P) - (mP) - (m-1)(\mathcal{O}) \quad (6)$$

$$div(f_{2m+1,P}) = (2m+1)(P) - ((2m+1)P) - (2m)(\mathcal{O}) \quad (7)$$

$$div(C_{mP,2\times P}) = 2(mP) + (P) + ((-2m-1)P) - 4(\mathcal{O}) \quad (8)$$

$$div(v_{(2m+1)P}) = ((2m+1)P) + ((-2m-1)P) - 2(\mathcal{O}) \quad (9)$$

が成り立つ．ここで，

$$\begin{aligned}&(8) - ((7) - 2\times(6))\\&= ((2m+1)P) + ((-2m-1)P) - 2(\mathcal{O})\\&= (9)\end{aligned}$$

となる．$(8) - ((7) - 2\times(6)) = (9)$ より，

$$\begin{aligned}&div(f_{2m+1,P})\\&= 2div(f_{m,P})) + div(C_{mP,2\times P}) - div(v_{(2m+1)P})\end{aligned}$$

となる．よって，

$$f_{2m+1,P} = f_{m,P}^2 \cdot C_{mP,2\times P}/v_{(2m+1)P}$$

が成り立つ．計算は Algorithm 3 のようになる．

Algorithm 3: Miller のアルゴリズム (2 倍-加算を使用)

Input: $P \in \mathbb{G}_1, Q \in \mathbb{G}_2, n = (n_{s-1}, \cdots, n_0)_2$ 但し，$n_{s-1} = 1$

Output: $f_{n,P}(Q)$

1. $V \leftarrow P, f \leftarrow 1$ **and** $i \leftarrow s-2$
2. **while** $i \geq 0$
3. **if** $n_i = 0$
4. $f \leftarrow f^2 \cdot \dfrac{l_{V,V}(Q)}{v_{2V}(Q)}, V \leftarrow 2V$
5. **if** $n_i = 1$
6. $f \leftarrow f^2 \cdot \dfrac{C_{V,2\times P}(Q)}{v_{2V+P}(Q)}, V \leftarrow 2V + P$
7. **return** f

3 提案手法

本節では，2 倍-加算を使用した Miller のアルゴリズム (Algorithm 3) において，座標変換を以下の 2 つの計算に施すことで従来 (Algorithm 2) より計算コストを削減できることを示す.

- 2 倍-加算 $(2V+P)$ で V の x 座標を 0 にする.
- $C_{V,2\times P}(Q)$ で代入する Q の x 座標を 0 にする.

3.1 $2V + P$ の計算

楕円曲線 $E: y^2 = x^3 + b$ 上の点 $P = (x_1, y_1), V = (x_2, y_2)$ の x_2 を 0 とする座標変換を行う．変換後の楕円曲線を E' とすると E' は

$$\begin{aligned}E': y^2 &= (x+x_2)^3 + b\\&= x^3 + (3x_2)x^2 + (3x_2^2)x + x_2^3 + b\end{aligned}$$

となる．変換後の点 P', V' はそれぞれ

$$\begin{aligned}P' &= (x_1', y_1) &= (x_1 - x_2, y_1)\\V' &= (0, y_2) &= (x_2 - x_2, y_2)\end{aligned}$$

となる．P' を通り V' で E' と接する 2 次曲線 $C_1: y = ax^2 + bx + c$ を構成すると，2.1.3 節の 2 倍-加算より以下の連立方程式が成り立つ．

$$\begin{cases} c = y_2 \\ b = \dfrac{3x_2^2}{2y_2} \\ a = \dfrac{y_1 - bx_1' - c}{x_1'^2} \end{cases}$$

連立方程式を解き，a, b, c, x_1', y_1, y_2 から 2 倍-加算 $2V' + P' = (x_4', y_4)$ は

$$\begin{cases} x_4' = \dfrac{1-2ab}{a^2} - x_1' \\ y_4 = -x_4'(ax_4' + b) - c \end{cases}$$

となる．x_2 平行移動した分を戻すことで $2V + P$ を求めることができる．よって $2V + P = (x_4, y_4)$ は

$$\begin{aligned}2V + P &= (x_4' + x_2, y_4)\\&= (x_4, y_4)\end{aligned}$$

となる．計算はアルゴリズム 4 のように行われる．1 回の $\mathbb{F}_p$ 乗算，2 乗算，逆元計算のコストをそれぞれ M，S，I で表すと，アルゴリズム 4 の計算コストは 7M+3S+3I である．

3.2 $C_{V,2\times P}(Q)$ の計算

3.1 節の 2 倍-加算で求めた 2 次曲線 $C_1: y = ax^2 + bx + c$ は x_2 平行移動した点 V' と P' を通る 2 次曲線なので，$C_{V,2\times P}(Q)$ の計算では x_2 平行移動した分を戻した 2 次曲線の式を使用する必要がある．ここでは更に代入する $Q = (x_Q, y_Q)$ の x 座標を 0 とする座標変換を同時に行う．変

Algorithm 4: 2 倍-加算

Input: $P=(x_1,y_1), V=(x_2,y_2)$
Output: $2V+P=(x_4,y_4)$

x_1'	$\leftarrow$	x_1-x_2	
W_1	$\leftarrow$	$(2y_2)^{-1}$	$(1I)$
W_2	$\leftarrow$	$3x_2^2$	$(1S)$
b	$\leftarrow$	W_1W_2	$(1M)$
c	$\leftarrow$	y_2	
W_3	$\leftarrow$	y_1-bx_1'	$(1M)$
W_4	$\leftarrow$	W_3-c	
W_5	$\leftarrow$	$x_1'^2$	$(1S)$
W_6	$\leftarrow$	W_5^{-1}	$(1I)$
a	$\leftarrow$	W_4W_6	$(1M)$
W_7	$\leftarrow$	a^2	$(1S)$
W_8	$\leftarrow$	W_7^{-1}	$(1I)$
W_9	$\leftarrow$	$1-2ab$	$(1M)$
x_4'	$\leftarrow$	$W_8W_9-x_1'$	$(1M)$
W_{10}	$\leftarrow$	$ax_4'+b$	$(1M)$
y_4	$\leftarrow$	$-x_4'W_{10}-c$	$(1M)$
x_4	$\leftarrow$	$x_4'+x_2$	

total cost: $7M+3S+3I$

換後の 2 次曲線を C_2 とすると C_2 は

$$\begin{aligned} C_2: y &= a(x-(x_2-x_Q))^2 \\ &\quad +b(x-(x_2-x_Q))+c \\ &= ax^2+(b-2a(x_2-x_Q))x \\ &\quad +(x_2-x_Q)(a(x_2-x_Q)-b)+c \end{aligned}$$

となる．この式に $Q'=(0,y_Q)$ を代入することで以下のように $C_{V,2\times P}(Q)$ は求めることができる．

$$C_{V,2\times P}(Q)=(x_2-x_Q)(a(x_2-x_Q)-b)+c-y_Q$$

4 コスト評価

この節では従来法と提案法による，BN 曲線での Ate ペアリング計算のコスト比較を行う．使用した BN 曲線のパラメータは以下の通りである．

楕円曲線 $E: y^2=x^3+2$
$z=7503760301170064939$
素数 $p=p(z)$
$=114134860152870813815265111402949341344$
$\quad 5521014660164156665915997572615 19697899$
位数$\#E(\mathbb{F}_p)=n(z)$
$=114134860152870813815265111402949341344$
$\quad 2142629540719204845311039850926572555 73$
トレース $t=p+1-\#E(\mathbb{F}_p)$

4.1 拡大体構成

この節では，筆者が実装した拡大体の構成，コスト評価のための拡大体のいくつかの表記を記す．拡大体の構成は以下のようになっている [12].

$$\begin{cases} \mathbb{F}_{p^2} &= \mathbb{F}_p[X]/(X^2+1) \\ \mathbb{F}_{p^6} &= \mathbb{F}_{p^2}[Y]/(Y^3-X-1) \\ \mathbb{F}_{p^{12}} &= \mathbb{F}_{p^6}[Z]/(Z^2-Y) \end{cases}$$

上述の拡大体構成において，12 次拡大の元は

$$\begin{aligned} &a_0+a_1X+a_2Y+a_3XY+a_4Y^2+a_5XY^2 \\ &+a_6Z+a_7XZ+a_8YZ+a_9XYZ+a_{10}Y^2Z+a_{11}XY^2Z \\ &(a_0,\cdots,a_{11}\in\mathbb{F}_p) \end{aligned}$$

と表される．2.2.1 節での群 $\mathbb{G}_1, \mathbb{G}_2$ の元はそれぞれ以下のようになっている．

- $P=(x_P,y_P)\in\mathbb{G}_1$
  $x_P=\alpha Y+\beta XY,\quad y_P=\gamma YZ+\delta XYZ$
  $(\alpha,\beta,\gamma,\delta\in\mathbb{F}_p)$

- $Q=(x_Q,y_Q)\in\mathbb{G}_2$
  $x_Q=\epsilon,\quad y_Q=\zeta$
  $(\epsilon,\zeta\in\mathbb{F}_p)$

ここでは，12 次拡大体の元に対して，α 個の $\mathbb{F}_p$ 係数が 0 でないとき，α-dense ということにする[2]．例えば，上記の x_P や y_P は 2-dense である．

4.2 計算式及び計算コスト

ここで，従来法と提案法の Miller のアルゴリズムの各ステップに必要となる計算コストをまとめておく．$V=(x_V,y_V), P=(x_P,y_P)\in\mathbb{G}_1, Q=(x_Q,y_Q)\in\mathbb{G}_2$ とする．また，除算処理を最後に一度だけ行うようにするために f を $f=\dfrac{f_n}{f_d}$ のように分子と分母に分けて計算を行う．計算コストには加算のコストは含めず，乗算，2 乗算，逆元計算でコスト評価を行う．

$\mathbb{F}_{p^{12}}$ の乗算と 2 乗算，逆元計算のような各演算は元の dense 数によって実装法を変えることで計算コストを削減できる．

[2] ある体 $\mathbb{F}$ に対して、$\mathbb{F}$ の拡大体の元の表現において β 個の $\mathbb{F}$ 係数が 0 であるとき，β-sparse ということがあるが，本稿では後の便宜上 dense を用いる．

以下では各演算に対して次のような記号を用いる．

$$\begin{array}{lcl}\times_\alpha & : & \text{コストが } \alpha\text{M の } \mathbb{F}_{p^{12}} \text{ 乗算} \\ *_\alpha & : & \text{コストが } \alpha\text{M の } \mathbb{F}_{p^{12}}\ 2 \text{ 乗算} \\ I_\alpha(\cdot) & : & \alpha\text{-dense の逆元計算} \\ & & I_2(\cdot) \text{ のコスト} = 3.6M + I \\ & & I_6(\cdot) \text{ のコスト} = 108.6M + I \end{array}$$

• アルゴリズム 2 のステップ 4

(i) $\dfrac{f_n}{f_d} \leftarrow \dfrac{f_n^2 \cdot l_{V,V}(Q)}{f_d^2 \cdot v_{2V}(Q)}$

○ 分子の計算

$f_n *_{54.4} f_n \times_{54} (2y_V \times_5 (y_Q - y_V) - 3x_V *_{2.4} x_V \times_5 (x_Q - x_V))$

○ 分母の計算

$f_d *_{17.6} f_d \times_{19} (2y_V \times_5 (x_Q - x_{2V}))$

(ii) $V \leftarrow 2V$

$$\begin{array}{lcl} x_{2V} & = & \lambda *_{2.4} \lambda - 2x_V \\ y_{2V} & = & \lambda \times_3 (x_V - x_{2V}) - y_V \\ \lambda & = & 3x_V *_{2.4} x_V \times_3 I_2(2y_V) \end{array}$$

(i) と (ii) の合計コスト

$176.8M + I$

• アルゴリズム 2 のステップ 6

ステップ 4 と同様

• アルゴリズム 2 のステップ 7

(iii) $\dfrac{f_n}{f_d} \leftarrow \dfrac{f_n \cdot l_{V,P}(Q)}{f_d \cdot v_{V+P}(Q)}$

○ 分子の計算

$f_n \times_{49} ((x_P - x_V) \times_5 (y_Q - y_V) - (y_P - y_V) \times_5 (x_Q - x_V))$

○ 分母の計算

$f_d \times_{19} ((x_Q - x_{V+P}) \times_5 (x_P - x_V))$

(iv) $V \leftarrow V + P$

$$\begin{array}{lcl} x_{V+P} & = & \lambda *_{2.4} \lambda - x_V - x_P \\ y_{V+P} & = & \lambda \times_3 (x_V - x_{V+P}) - y_V \\ \lambda & = & (y_P - y_V) \times_3 I_2(x_P - x_V) \end{array}$$

(iii) と (iv) の合計コスト

$95M + I$

• アルゴリズム 3 のステップ 4

アルゴリズム 2 のステップ 4 と同様

• アルゴリズム 3 のステップ 6

(v) $\dfrac{f_n}{f_d} \leftarrow \dfrac{f_n^2 \cdot C_{V,2\times P}(Q)}{f_d^2 \cdot v_{2V+P}(Q)}$

○ 分子の計算

$f_n *_{54.4} f_n \times_{50} ((x_V - x_Q) \times_8 (a \times_5 (x_V - x_Q) - b) + c - y_Q)$

○ 分母の計算

$f_d *_{17.6} f_d \times_{13} (x_Q - x_{2V+P})$

(vi) $V \leftarrow 2V + P$

V の x 座標を 0 とする座標変換を行うと

$$\begin{array}{lclcl} P' & = & (x'_P, y_P) & = & (x_P - x_V, y_P) \\ V' & = & (0, y_V) & = & (x_V - x_V, y_V) \end{array}$$

となる．次に以下の a, b, c を求める．

$$\left\{\begin{array}{lcl} c & = & y_V \\ b & = & 3x_V *_{2.4} x_V \times_3 I_2(2y_V) \\ a & = & (y_P - b \times_3 x'_P - c) \times_3 I_2(x'_P *_{2.4} x'_P) \end{array}\right.$$

a, b, c, x'_P, y_P, y_V から以下の計算を行う．

$$\begin{array}{lcl} x'_{2V+P} & = & (1 - 2a \times_3 b) \times_3 I_2(a *_{2.4} a) - x'_P \\ y_{2V+P} & = & -x'_{2V+P} \times_3 (a \times_3 x'_{2V+P} + b) - c \end{array}$$

平行移動した分を戻す処理を行い，$2V + P$ を求める．

$$\begin{aligned} 2V + P &= (x'_{2V+P} + x_V, y_{2V+P}) \\ &= (x_{2V+P}, y_{2V+P}) \end{aligned}$$

(v) と (vi) の合計コスト

$187M + 3I$

(vii) 最後の除算処理 $f \leftarrow \dfrac{f_n}{f_d}$

$f_n \times_{44} I_6(f_d)$

合計コスト: $152.6M + I$

4.3 Ate ペアリング計算コスト比較

Ate ペアリングの従来 (Algorithm 2) の計算コストと 3.1 の 2 倍-加算と 3.2 の計算をアルゴリズム 3 に適用した場合の計算コスト比較を行った．$|r|$ を r のビット長，$HW(r)$ をハミング重みとすると，従来法と提案法のペアリング計算 1 回にかかるコスト計算式は以下のようになる．

従来法:
$(|r| - HW(r)) \cdot ((\mathrm{i}){+}(\mathrm{ii}))$
$+HW(r) \cdot ((\mathrm{i}){+}(\mathrm{ii}){+}(\mathrm{iii}){+}(\mathrm{iv})) + (\mathrm{vii})$
提案法:
$(|r| - HW(r)) \cdot ((\mathrm{i}){+}(\mathrm{ii}))$
$+HW(r) \cdot ((\mathrm{v}){+}(\mathrm{vi})) + (\mathrm{vii})$

1回の$\mathbb{F}_p$乗算，逆元計算のコストをそれぞれM，Iで表すと，従来法と提案法のコスト計算式は以下のようになる．

従来法:
$(|r| - HW(r)) \cdot (176.8M + I)$
$+HW(r) \cdot (271.8M + 2I) + 152.6M + I$
提案法:
$(|r| - HW(r)) \cdot (176.8M + I)$
$+HW(r) \cdot (187M + 3I) + 152.6M + I$

逆元計算Iは計算環境によってコストが異なるため，ここでは5M，7M，10Mの3つの場合でそれぞれコスト評価を行った．$r = 128$ビット，$HW(r) = 0.5 \times r$のビット数とした場合の従来法と提案法のコスト比較を表1に示す．どの場合のおいても提案法は，従来法に比べ約15%程コストを削減という結果となった．

表 1: 従来法と提案法のコスト比較

	従来法	提案法	削減率
$I = 5M$	$29828M$	$24720.8M$	17.1%
$I = 7M$	$30214M$	$25234.8M$	16.4%
$I = 10M$	$30793M$	$26005.8M$	15.5%

5 まとめと今後の課題

本稿は，アフィン座標におけるAteペアリング演算において，演算で使用する楕円曲線上のある点のx座標を0にする座標変換や2次曲線を使用した2倍-加算公式をMillerのアルゴリズムに適用することで計算コストを削減し，ペアリング演算が高速化されることを示した．今後の課題として，射影座標やJacobian座標において提案手法がどのような影響を与えるのか調査したい．

謝辞

本研究はJSPS科研費25330156の助成を受けたものです．

参考文献

[1] P. Barreto and M. Naehrig, “Pairing-friendly elliptic curves of prime order,” SAC 2005, LNCS 3897, pp.319-331, 2006

[2] D. Boneh and M. Franklin, “Identity based encryption from the Weil pairing,” SIAM Journal of Computing, Vol. 32, No. 3, pp. 586-615, 2003.

[3] K. Chalkias and G. Stephanides, “Timed release cryptography from bilinear pairings using hash chains, ”CMS 2006, LNCS 4237, pp. 130-140, 2006.

[4] H. Cohen, G. Frey, R. Avanzi, C. Doche, and T. Lange, *Handbook of elliptic and hyperelliptic curve cryptography*, 2nd edition, Chapman and Hall/CRC, 2011.

[5] GMP, `https://gmplib.org/`

[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based Encryption for Fine-grained Access Control of Encrypted Data,” CCS 2010, pp.89-98, 2006.

[7] F. Hess, N. Smart, and F. Vercauteren, “The Eta Pairing Revisited,” IEEE Transactions on Information Theory, vol.52, pp.4595-4602, 2006

[8] E. Lee, H. Lee, and C. Park, “Efficient and generalized pairing computation on abelien varieties,” IEEE Transactions of Information Theory, Vol.55, No.4, pp.1793- 1803, 2009.

[9] 宮地充子, 代数学から学ぶ暗号理論, 日本評論社, 2012.

[10] Y. Nogami, M. Akane, Y. Sakemi, H. Kato, and Y. Morikawa, “Integer variable A-based Ate pairing,” Pairing 2008, LNCS 5209, pp.178-191, 2008.

[11] J. H. Silverman, *The arithmetic of elliptic curves*, GTM106, Springer, 1986.

[12] M. Shirase, “Universal construction of a 12th degree extension field for asymmetric pairing,” IEICE Transactions Vol. 94-A, No. 1, pp. 156-164, 2011.

[13] 白勢政明, “2次曲線を用いた楕円曲線演算,” SCIS2014.