3次ツイストを利用した効率的なペアリングアルゴリズム

石井 将大[1,a)]　照屋 唯紀[2,b)]　安田 貴徳[3,c)]

概要：ペアリング演算の効率的な実装として，埋め込み次数とツイスト次数が偶数で高次の楕円曲線を利用した実装が多く報告されている．最近，数体篩法による有限体離散対数問題の困難性評価が見直された．その結果，埋め込み次数が奇数の 3 次ツイストが利用可能な楕円曲線が新しい効率的パラメータの候補として挙げられるようになった．本研究ではこの様な楕円曲線上のペアリング計算アルゴリズムに対し，先行研究において提案されている手法を整理し，逆元演算が不要な新しい符号付き 2 進数展開による Miller のアルゴリズムを提案する．この提案法により，効率的にペアリング演算を計算できるパラメータの空間が拡大されたと考えられる．そこで，実際に効率的なペアリング演算に適したパラメータの探索を実施した．その結果，ハミング重みがより小さいパラメータや，効率的な拡大体を構成できるようなパラメータを新たに発見した．これらのことから，3 次ツイストを利用した高速なペアリング演算の実装については，効率化および高速化の余地がまだ残されていると考えられる．

キーワード：楕円曲線，ペアリング，3 次ツイスト，Miller のアルゴリズム，符号付き 2 進数展開

Efficient Pairing Algorithms Using Twists of Degree 3

Masahiro Ishii[1,a)]　Tadanori Teruya[2,b)]　Takanori Yasuda[3,c)]

Abstract: An elliptic curve with efficient implementation of the pairing often has a twist of a higher even degree and an even embedding degree. The complexity assessment of the finite field discrete logarithm problem using recently proposed number field sieving has been renewed, and elliptic curves with twists of degree 3 and odd embedding degrees become new efficient candidates. In this paper, we revisit the methods proposed in the prior research for the pairing calculation algorithm over such the elliptic curves. In particular, we propose a new variant of signed-binary Miller's algorithm without the inversion operation. We also revisit the parameter search, which provides efficient pairing by using our proposed algorithm. As a result, we found new parameters, one has low Hamming weight, and another one has an efficiently computable extension field. In conclusion, there remains room to accelerate pairing calculation over elliptic curves with twists of degree 3 and odd embedding degrees.

Keywords: elliptic curve, pairing, twist of degree 3, Miller's algorithm, signed-binary expansion

1. はじめに

ペアリング暗号とは，楕円曲線上に定義されるペアリング演算を利用して構成される暗号技術の総称である．様々な機能を持った暗号方式を高い計算効率で実現可能なため，実用的な高機能暗号の基盤技術として注目を集めている．

ペアリング暗号の安全性は，使用する楕円曲線において定義される計算問題の困難性を根拠とする．そのうち代表的な問題は楕円曲線離散対数問題 (ECDLP) と，有限体離

1 東京工業大学 〒152-8550 東京都目黒区大岡山 2-12-1
Tokyo Institute of Technology. 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550 Japan
2 産業技術総合研究所 〒135-0064 東京都江東区青海 2-4-7
National Institute of Advanced Industrial Science and Technology. 2-4-7 Aomi, Koto-ku, Tokyo 135-0064 Japan
3 岡山理科大学 〒700-0005 岡山県岡山市北区理大町 1-1
Okayama University of Science. 1-1 Ridaicho, Kita-ku, Okayama-shi, Okayama, 700-005 Japan
a) mishii@gsic.titech.ac.jp
b) tadanori.teruya@aist.go.jp
c) tyasuda@bme.ous.ac.jp

© 2020 Information Processing Society of Japan

散対数問題 (FFDLP) である．現時点において，ECDLP については深刻な解法の改良や解読報告はなされていない．一方，FFDLP については解法である数体ふるい法の改良が提案された [21]．この改良に対応し，安全性評価の見直しや推奨楕円曲線リストが提案されている [4], [5], [15]．これら報告について興味深い点は，これまでは 6 次ツイストを使用した楕円曲線が効率的で最速のペアリング暗号の実装を与えると考えられていたが，安全性評価の見直しの結果，それ以外の楕円曲線が推奨として挙げられていることである．これら楕円曲線の中から最良の選択をするためには，安全性評価だけでなく最も効率的なアルゴリズムを明らかにする必要がある．新しく注目されるようになった楕円曲線のうち，埋め込み次数が奇数で 3 の倍数であり CM 判別式が 3 の楕円曲線は 3 次ツイストを持つ．本稿ではこの様な楕円曲線およびその曲線族におけるペアリング演算の効率的な計算方法について述べる．

1.1 ツイスト次数 3 における既存研究

3 次ツイストを持つ楕円曲線は 6 次ツイストと同様にペアリング演算の計算に有利な性質を持つ [19]．また，3 次ツイストを持つ楕円曲線も曲線族として特徴付けられる [13]．ここで曲線族とは，効率的なペアリング演算を持つ楕円曲線の定義体の位数やトレースを多項式で表現したものである．ペアリング演算の計算コストはトレースの値に依存するため，曲線族から楕円曲線を生成するために多項式に代入する整数値がペアリング演算の計算コストに大きな影響を与える．そのため，高速実装を行う際には計算効率が良い整数値を選ぶことが重要である．この整数値は単にパラメータとも呼ばれる．性能評価および実装報告 [5], [12], [15], [16], [27] では，整数値を通常の符号無し 2 進数展開した時，ハミング重みが小さいもののみが議論されている．その理由は，ペアリング演算の計算に Zhang と Lin [33] のアルゴリズムのみを考慮しており，この場合は符号付き 2 進数展開で得られた負のビットの数だけ逆元演算が必要となるため，非効率的であると見なされたためと考えられる．Nanjo ら [28] は，Teruya ら [31] と Aranha ら [2] が示した式を組み合わせた符号付き 2 進数展開アルゴリズムについて議論している．

1.2 貢献

本稿では，3 次ツイストにおける逆元演算が不要な符号付き 2 進数展開によるペアリング演算，特に optimal ate ペアリングのための Miller 関数値の計算方法を整理し，その構成方法を明確にする．そして，これまでのテクニックを応用した新しい Miller のアルゴリズムを提案する．漸近的に非効率とされる逆元演算が不要で，さらに演算回数を削減するようなパラメータの設定方法が明らかとなったことから，従来よりも効率的なパラメータが存在する空間が拡大されたと考えられる．実際に探索を行ったところ，有用と考えられる性質を持つ新しいパラメータをいくつか発見できた．この結果から，3 次ツイストには，従来よりもさらにペアリング演算を高速に計算できるパラメータの存在可能性が示唆される．

2. 数学的準備

この節では，本稿で扱う楕円曲線上に定義されるペアリング演算について簡単に解説する．詳細な解説は [11] がある．

2.1 双線型写像

位数が十分に大きな素数 r の 3 つの巡回群 $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathrm{T}$ が存在し，その上で定義される関数 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\mathrm{T}$ が次の 3 つの性質を満たす時，e を (暗号学的) 双線型写像またはペアリング演算と呼ぶ．

(1) 双線型: 任意の $a, b \in \mathbb{G}_1$ と任意の $c, d \in \mathbb{G}_2$ について，$e(a \cdot b, c) = e(a, c) \cdot e(b, c)$ および $e(a, c \cdot d) = e(a, c) \cdot e(a, d)$.

(2) 非退化: $e(a, b)$ が単位元となるのは，a が単位元または b が単位元の時.

(3) 効率性: e の漸近的計算時間は $\log r$ の多項式.

特に $\mathbb{G}_1 \neq \mathbb{G}_2$ で，この 2 つの間に効率的な準同型写像が存在しない時，e は Type 3 である．本稿では Type 3 の双線型写像のみを論じる．なお，正確には離散対数問題や Diffie–Hellman 問題，ペアリング逆問題などの困難性が要件として求められるが [11]，本稿では論じない．

2.2 楕円曲線

標数 $p > 3$ の有限体 $\mathbb{F}_p$ 上定義される楕円曲線 E とは，2 つの定数 $a, b \in \mathbb{F}_p$ (但し $4a^3 + 27b^2 \neq 0$) と 2 変数 X, Y の Weierstrass 方程式 $Y^2 = X^3 + aX + b$ のことを指す．また，この時 $\mathbb{F}_p$ を E の定義体と呼ぶ．拡大体 $\mathbb{F}_{p^\ell}$ 上の有理点群を $E(\mathbb{F}_{p^\ell})$ と書き，単位元の無限遠点は ∞ と書く．$P = (x, y) \in E(\mathbb{F}_{p^\ell})$ の逆元は $-P = (x, -y)$ である．$P \in E(\mathbb{F}_{p^\ell})$ の整数 n 倍を $[n]P$ と書く．また，ある正整数 n について，$E[n] := \ker([n])$ と書き，n-ねじれ点と呼ぶ．$\pi_{p^\ell} : (x, y) \mapsto (x^{p^\ell}, y^{p^\ell})$ を p^ℓ 乗フロベニウス準同型写像と呼ぶ．$t := p + 1 - \#E(\mathbb{F}_p)$ を (フロベニウス準同型写像の) トレースと呼ぶ．素数 r は $\gcd(r, p) = 1$ および $r \mid \#E(\mathbb{F}_p)$，そして $r^2 \nmid \#E(\mathbb{F}_p)$ が成り立つとする．有限体 $\mathbb{F}_p$ 上定義される楕円曲線 E の群位数 r に関する埋め込み次数 k とは，$r \mid (p^k - 1)$ が成り立つ最小の正整数のことである．

2.3 Miller のアルゴリズム・最終べきとペアリング

楕円曲線 E は通常曲線とし，群位数 r に関する埋め込み次数を $k > 1$ とする．$\mathbb{G}_1 := E[r] \cap \ker(\pi_p - [1])$

Algorithm 1: Miller のアルゴリズム [25]

Input: $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$.
Data: $s = \sum_{i=0}^{\mathtt{mb}} s_i 2^i$.
Output: $f = f_{s,Q}(P)$.

$(f, V) \leftarrow (1, Q)$;
for $i = \mathtt{mb} - 1$ **down to** 0 **do**
  $(f, V) \leftarrow \left(f^2 \cdot (\frac{\ell_{V,V}}{v_{[2]V}})(P),\ [2]V\right)$;
  if $s_i = 1$ **then** $(f, V) \leftarrow \left(f \cdot (\frac{\ell_{V,Q}}{v_{V+Q}})(P),\ V + Q\right)$;
end
return f;

および $\mathbb{G}_2 := E[r] \cap \ker(\pi_p - [p])$, $\mathbb{G}_{\mathrm{T}} := \mu_r$ とする．ここで，μ_r は 1 の r 乗根がなす $\mathbb{F}_{p^k}^{\times}$ の乗法部分群である．2 つの点 $P, Q \in E$ について，関数 $\ell_{P,Q}$ を，その因子が $(\ell_{P,Q}) = (P) + (Q) + (-(P+Q)) - 3(\infty)$ となる E の正規化された関数とする．$\ell_{P,Q}$ は P と Q を通る直線を表す．$v_P := \ell_{P,-P}$ とする．次数が整数 a および台が点 $P \in E$ の Miller 関数 $f_{a,P}$ とは，その因子が $(f_{a,P}) = a((P) - (\infty)) - (([a]P) - (\infty))$ となる E の正規化された関数のことを指す．任意の整数 a, b と任意の点 $P \in E$ について，$f_{0,P}$ と $f_{1,P}$ は定数関数であり，$(f_{-a,P}) = -(f_{a,P}) - (v_P)$, $(f_{a+b,P}) = (f_{a,P}) + (f_{b,P}) + (\ell_{[a]P,[b]P}) - (v_{[a+b]P})$ である．また，Miller は $\mathbb{F}_{p^k} \times E(\mathbb{F}_{p^k})$ の 2 つの要素 $(a, P_a) := (f_{a,P}(Q), [a]P)$, $(b, P_b) := (f_{b,P}(Q), [b]P)$ に対して Q を補助入力とする群演算 $(a, P_a) \star (b, P_b) := \left(a \cdot b \cdot (\ell_{P_a,P_b}/v_{P_a+P_b})(Q), P_a + P_b\right) = (f_{a+b,P}(Q), [a+b]P)$ を利用した 2 進数展開アルゴリズムを示した．これは Miller のアルゴリズム (または Miller ループ) と呼ばれる [25]．ある正整数 $s = \sum_{i=0}^{\mathtt{mb}} s_i 2^i$ に対して，$f_{s,Q}(P)$ を効率的に求める Miller のアルゴリズム [25] を Algorithm 1 に示す．"**Data**" は，入力に依存しない曲線固有のパラメータであることを示す．ここで記号 mb は，s の 2 進数展開における最上位ビットの位置を表し，$i = 0, \ldots, \mathtt{mb}$ について $s_i \in \{0, 1\}$ である．

$f_{r,P}(Q)^{(p^k-1)/r}$ および $f_{r,Q}(P)^{(p^k-1)/r}$ は (reduced) Tate ペアリング，$(-1)^r \cdot f_{r,P}(Q)/f_{r,Q}(P)$ は Weil ペアリングと呼ばれ，双線型写像の要件を満たす．また，$(p^k - 1)/r$ のべき乗は最終べきと呼ばれる．k 次円分多項式 Φ_k により $(p^k - 1)/r = ((p^k - 1)/\Phi_k(p)) \cdot (\Phi_k(p)/r)$ と分解できる．左の項 $(p^k - 1)/\Phi_k(p)$ は計算コストが小さく easy part と呼ばれる．右の項 $\Phi_k(p)/r$ は計算コストが大きく hard part と呼ばれる．また，easy part は cyclotomic subgroup と呼ばれる部分群に要素を写す．この部分群では，逆元演算や平方算などが効率的に計算できる [5]．Hard part の計算は固定された指数によるべき乗算となる．これに対して最小べき乗 (加算) 連鎖を求める方法が提案されている [14], [17], [29], [33]．本稿では easy part および hard part の効率的な計算方法は取り扱わない．

3. 関連研究

3.1 Miller のアルゴリズムの効率化

Hess ら [19] は，いくつかの条件の下で $f_{t-1,Q}(P)^{(p^k-1)/r}$ は双線型写像の要件を満たすことを示した．これは Ate ペアリングと呼ばれる．なお，Ate (および twisted Ate) ペアリングの構成に関係する理論を Weil ペアリングに適用することで高速化した双線型写像は Eil ペアリングと呼ばれ，その効率化については Aranha ら [3] および Kinoshita と Suzuki [22] が提案している．

Hess ら [19] はさらに，ツイストを利用した高速化も示している．埋め込み次数 k を割る d について，$m = k/d$ とする．$d = 2, 3, 4$ または 6 で，E の Weierstrass 方程式がある特別な形式の時，$\mathbb{F}_{p^k}$ 上の有理点群が E と同型になる d 次ツイスト E' が存在し，対応する同型写像 $\phi_d : E'(\mathbb{F}_{p^k}) \to E(\mathbb{F}_{p^k})$ は効率的に計算可能である．この ϕ_d により E と E' の自己準同型環は同型となり，$Q' \in E'[r] \cap E'(\mathbb{F}_{p^m})$ とする時，$f_{r,Q'}(P')^{(p^k-1)/r}$ も双線型写像となる [10]．そこで，特に断らない限り E と E' を区別しない．

Vercauteren [32] と Hess [18] は，最も短い Miller のアルゴリズムのループ長を持つ双線型写像を構成するフレームワークを示した．ある整数 s とある多項式 $h = \sum_{i=0}^{d} h_i X^i \in \mathbb{Z}[X]$ について $h(s) \equiv 0 \pmod{r}$ が成り立ち，$R \in E[r]$ に対する E の正規化された関数 $f_{s,h,R}$ を，その因子が $(f_{s,h,R}) = \sum_{i=0}^{d} h_i(([s^i]R) - (\infty))$ となる関数とする．これをペアリング関数と呼ぶ．s, h, R が一定の条件を満たす時，対応するペアリング関数を使用して双線型写像を構成できる [18], [32]．Miller 関数との関係は，$S_i := \sum_{j=i}^{d} [h_j s^j] R$ として，$(f_{s,h,R}) = \sum_{i=0}^{d} (f_{h_i,[s^i]R}) + \sum_{i=0}^{d-1} ((\ell_{[h_i s^i]R, S_{i+1}}) - (v_{S_i}))$．効率的なペアリング演算を構成するためには，条件を満たす係数が小さい h を探索すればよい．直観的に言えば，h の係数ベクトルの 1-ノルムが $r^{\varphi(k)}$ に近い時，そのようなペアリング関数を optimal と呼ぶ．ここで φ は Euler の totient 関数である．

3.2 分母消去

Barreto ら [7] と Lin ら [24]，Zhang と Lin [33]，Le と Tan [23] は，Miller のアルゴリズムに現れている垂線関数 v の評価値による逆元演算を，最終べきの easy part が持つ構造を利用して消去する方法を提案した．

Barreto ら [7] は，埋め込み次数が偶数の場合は，Miller のアルゴリズムに現れている v の評価値による逆元演算を全て消去可能であることを示した．

Lin ら [24] は，3 次ツイストを利用できる場合における分母消去を提案した．さらに Zhang と Lin [33]，Le と Tan [23] は Affine 座標系における改良を提案した．本稿では $\ell_{P_1,P_2}/v_{P_1+P_2}(Q)$ や $1/v_P(Q)$ の計算に対して分母消去を

適用した計算をそれぞれ $\ell'_{P_1,P_2}(Q), v'_P(Q)$ と書く．

3.3 符号付き 2 進数展開

Miller 関数 $f_{a,P}$ について，次数 a のハミング重みが小さい程，$f_{a,P}(Q)$ を求める Miller のアルゴリズムの計算コストは小さい．次数 a を符号付き 2 進数展開した $a = \sum_{i=0}^{\mathtt{mb}} a_i 2^i$ (ここで $a_i \in \{-1, 0, 1\}$) を考えることで，ハミング重み (非ゼロの $a_0, \ldots, a_{\mathtt{mb}}$ の個数) を小さくすることができる．但し，負のビットに対応する処理を Miller のアルゴリズムに追加しなければならない．

Beuchat ら [9] と Teruya ら [31] は，分母消去を利用した効率的な Miller のアルゴリズムの符号付き 2 進数展開を示した．Miller 関数の因子について，一般的に $(f_{a-1,P}) = (f_{a,P}) + (f_{-1,P}) + (\ell_{[a]P,-P}) - (v_{[a-1]P}) = (f_{a,P}) - (v_P) + (\ell_{[a]P,-P}) - (v_{[a-1]P})$ が成り立つことを利用する．これに分母消去 [7], [24] を適用し，逆元演算が存在しない効率的な Miller のアルゴリズムの符号付き 2 進数展開を得る．

Zhang と Lin [33] は，$(f_{a-1,P}) = (f_{a,P}) + (v_{[a]P}) - (\ell_{-[a]P,P})$ という式を示した．この式に対して分母消去を適用することはできないため，論文 [33] で述べられているように，負のビットの数だけ逆元演算の計算が必要となる．なお，論文 [5], [12], [15], [16], [27] は論文 [33] とその後続論文を参照し，論文 [9], [31] を参照していない．

また，符号付き 2 進数展開の特別な場合として，整数 a が全て負のビットで表現できる場合 (つまり $a = \sum_{i=0}^{w} a_i 2^i$ で $a_i \in \{0, -1\}$)，$f_{a,Q}(P)$ を計算する代わりに $f_{-a,Q}(P)$ を計算することで高速化をはかる方法が提案されている [2]．

3.4 効率的なペアリング演算に適した楕円曲線

効率的なペアリング演算を持つ楕円曲線を探索・生成する方法については Freeman ら [13] がまとめている．Kim と Barbulescu [20] および Kim と Jeong [21] による数体篩法の改良に対応した推奨楕円曲線リストが Barbulescu ら [5] と Guillevic [15] により提案されている．

本稿では，このような安全性評価の見直しにより注目されている曲線のうち，3 次ツイストが利用可能な曲線 BLS15 と BLS21 について簡単に紹介する．楕円曲線 BLS15 は，埋め込み次数 $k = 15$ で定義体の体位数 p と $E(\mathbb{F}_p)$ の部分群の位数 r，フロベニウス写像のトレース t がそれぞれ変数 x を使用して $p(x) = (x^{12} - 2x^{11} + x^{10} + x^7 - 2x^6 + x^5 + x^2 + x + 1)/3$, $r(x) = \Phi_{15}(x)$ および $t(x) = x + 1$ と多項式表現できる．楕円曲線 BLS21 は，埋め込み次数 $k = 21$ で $p(x) = x + (x-1)^2 \cdot (x^{14} + x^7 + 1)/3$, $r(x) = \Phi_{21}(x)$ および $t(x) = x + 1$ と多項式表現できる．

このように，変数 x により多項式表現された楕円曲線を曲線族と呼ぶ．変数 x に整数 z を代入し，いくつかの条件を満たすものがペアリング暗号として利用可能な有限体 $\mathbb{F}_p$ 上定義される楕円曲線 E の存在を示す．代表的な必要条件は p と r が共に素数となることである．楕円曲線を生成するために x に代入する整数 z は，その曲線族のパラメータと呼ばれる．

3.5 曲線族と効率的なパラメータ

曲線族を利用することで，ペアリング演算で計算すべき式が多項式表現として得られる．Miller 関数の次数と最終べきのべき乗指数について，これに対して定義体の体位数 p とパラメータ z を基数として展開し，その最小加算 (べき乗) 連鎖を探索することで，ペアリング演算を効率化しつつ，その計算コストを推定できる．例えば，BLS 曲線族で生成できる楕円曲線については，$f_{z,Q}(P)^{(p(z)^k-1)/r(z)}$ が optimal である．(最終べきのべき乗指数の詳細な記述は省略する．いくつかの BLS 曲線族については [17] で示されている．) このように，高速なペアリング演算を実装するためには，まず始めに曲線族が持つ多項式表現から推定できるアルゴリズムの計算コストを参考に候補を絞り，次に実際に高速なペアリング演算を持つ楕円曲線を与える整数 z を探索することが典型的な手順である．つまり，その時点において知られている最速のアルゴリズムが，曲線族の選択方法やパラメータの探索指針に重大な影響を与える．効率的なペアリング演算を実装するためのパラメータ探索における代表的な指針は次の通り．

- Miller 関数の次数のハミング重みが小さい．
- 最終べきのべき乗指数のハミング重みが小さい．
- $\mathbb{F}_{p^k}$ を既約二項式で構成できる．

また，より安全かつ高速なペアリング暗号を実現するために，特別な条件を満たすパラメータの探索指針・選択方法が提案されている [4], [6], [8]．但し，特別な条件を満たすパラメータに対する安全性解析も報告されている [30] ことに注意．

4. 主結果

この節では，本稿で提案する 3 次ツイストにおける効率的な符号付き 2 進数展開によるペアリング演算を計算するアルゴリズムについて述べる．提案するアルゴリズムでは，Miller ループ部分について，Aranha ら [2] の符号反転と，最終べきの easy part を利用した工夫が新たに施されている．

4.1 冗長符号付き 2 進数展開

本稿で提案するアルゴリズムを記述するために，次のように符号付き 2 進数展開を拡張する．

定義 1. ある非ゼロの整数 s に対して，次の条件を全て満たす有限個の数値列 $s_0, \ldots, s_{\mathtt{mb}}, \delta_{-1}, \delta_0, \ldots, \delta_{\mathtt{mb}-1}$ を，s の**冗長符号付き 2 進数展開** *(Redundant Signed-Binary Expansion, RSBE)* と呼ぶ:

- $i = 0, \ldots, \mathtt{mb}$ について $s_i \in \{-1, 0, 1\}$.
- $i = -1, \ldots, \mathtt{mb}-1$ について $\delta_i \in \{0, 1\}$.
- $s = \sum_{i=0}^{\mathtt{mb}} s'_i s_i 2^i$ が成り立ち，ここで $s'_i := (-1)^{\Delta_i}$ および $\Delta_i := \sum_{j=-1}^{i-1} \delta_j$.

この時，s の RSBE における最上位非ゼロビット (Most Significant Bit, MSB) の位置を mb とし，ハミング重みは $s_0, \ldots, s_{\mathtt{mb}}$ のうち非ゼロの個数とする．

ある非ゼロの整数 s の RSBE は複数存在することに注意．また，ある RSBE が $i = -1, \ldots, \mathtt{mb}-1$ について $\delta_i = 0$ の時，それは従来の符号付き 2 進数展開に等しい．

4.2 Miller ループと最終べきの Easy Part

4.2.1 アルゴリズムとその概要

本稿で提案するアルゴリズムを Algorithm 2 に示す．提案アルゴリズムは，ある RSBE に基づき，入力に対して ate ペアリング (Miller ループと最終べき) の計算を行う．Algorithm 2 の概要を，Teruya ら [31] および Nanjo ら [28]，そして Aranha ら [2] のアルゴリズムとの差分や関係と共に述べる:

- 第 6 行目に，$s_{\mathtt{mb}} = -1$ の場合の処理を Miller 関数の定義に従って追加．この変更は Nanjo ら [28] と同様である．
- 第 3-15 行目は，Aranha ら [2] と同様に，上位 2 ビット $s_{\mathtt{mb}}$ と $s_{\mathtt{mb}-1}$ の値に対する処理をアンロール．これにより $s_{\mathtt{mb}} = 1$ の時に $\mathbb{F}_{p^k}$ 平方算を 1 回減らすことができる．
- $\delta_0, \ldots, \delta_{\mathtt{mb}-1}$ の値によって，ある時点までに求めた Miller 関数の次数の符号と有理点のスカラー倍の値の符号を，第 9, 18 行目で適宜反転する．
- 第 27-33 行目で，最終べきを計算．この時，δ_{-1} の値による符号反転も行うが，第 9, 18 行目とは異なる方法で符号反転を計算する．

ここで，埋め込み次数を正整数 $d > 1$ によって $k = dm$ と書いた時，$p^k - 1 = p^{dm} - 1 = \left(\sum_{i=0}^{d-1} p^{im}\right)(p^m - 1)$ と分解できる．$d > 1$ と定義したので $m < k$ となり，埋め込み次数の定義より $r \nmid (p^m - 1)$．よって $r \mid \left(\sum_{i=0}^{d-1} p^{im}\right)$．また，$i = -1, \ldots, \mathtt{mb}-1$ について $\delta_i = 0$ の時，それは，ループの途中で求めた Miller 関数の次数の符号と有理点のスカラー倍の値の符号が反転しない，自然な意味での符号付き 2 進数展開の Miller のアルゴリズムの動作となる．

Miller 関数の公式 $(f_{-a,Q}) = -(f_{a,Q}) - (v_Q) = \left((f_{a,Q} \cdot v_Q)^{-1}\right)$ と次に述べる補題 1 について $d = 3$ とした式を組み合わせた手続きが第 9, 18 行目の処理である．この補題は Aranha らの符号反転 [2], Lemma 1 の一般化である．

補題 1. 埋め込み次数 k について，$k = dm$ と正整数 $d > 1$ で書いた時，$g, h \in \mathbb{F}_{p^k}^{\times}$ について，$\left(g^{-1} \cdot h\right)^{\frac{p^k-1}{r}} = \left(g^{\sum_{i=1}^{d-1} p^{im}} \cdot h\right)^{\frac{p^k-1}{r}}$ が成り立つ．

Algorithm 2: 提案アルゴリズム

Input: $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$.
Data: 定数 $p, r, k = dm$ $(d > 1)$, s の RSBE $s_0, \ldots, s_{\mathtt{mb}}, \delta_{-1}, \delta_0, \ldots, \delta_{\mathtt{mb}-1}$.
Output: $f = f_{s,Q}(P)^{(p^k-1)/r}$.

1 **if** $s_i = 1$ *for all* i **then** $t \leftarrow 1$;
2 **else** $t \leftarrow v'_Q(P)$; // E_v,
3 **if** $s_{\mathtt{mb}} = 1$ **then**
4 $\quad (f, V) \leftarrow (\ell'_{Q,Q}(P), [2]Q)$; // D
5 **else**
6 $\quad (f, V) \leftarrow (t^2 \cdot \ell'_{-Q,-Q}(P), [2](-Q))$; // $D + M_k + S_k$
7 **end**
8 **if** $\delta_{\mathtt{mb}-1} = 1$ **then**
9 $\quad (f, V) \leftarrow ((f \cdot v_V(P))^{p^{2m}+p^m}, -V)$; // $E_v + 2M_k + 2F_k$
10 **end**
11 **if** $s_{\mathtt{mb}-1} = 1$ **then**
12 $\quad (f, V) \leftarrow (f \cdot \ell'_{V,Q}(P), V + Q)$; // $A + M_k$
13 **else if** $s_{\mathtt{mb}-1} = -1$ **then**
14 $\quad (f, V) \leftarrow (f \cdot \ell'_{V,-Q}(P) \cdot t, V - Q)$; // $A + 2M_k$
15 **end**
16 **for** $i = \mathtt{mb} - 2$ **down to** 0 **do**
17 $\quad$ **if** $\delta_i = 1$ **then**
18 $\quad\quad (f, V) \leftarrow ((f \cdot v_V(P))^{p^{2m}+p^m}, -V)$; // $E_v + 2M_k + 2F_k$
19 $\quad$ **end**
20 $\quad (f, V) \leftarrow (f^2 \cdot \ell'_{V,V}(P), [2]V)$; // $D + M_k + S_k$
21 $\quad$ **if** $s_i = 1$ **then**
22 $\quad\quad (f, V) \leftarrow (f \cdot \ell'_{V,Q}(P), V + Q)$; // $A + M_k$
23 $\quad$ **else if** $s_i = -1$ **then**
24 $\quad\quad (f, V) \leftarrow (f \cdot \ell'_{V,-Q}(P) \cdot t, V - Q)$; // $A + 2M_k$
25 $\quad$ **end**
26 **end**
27 **if** $\delta_{-1} = 0$ **then**
28 $\quad f \leftarrow f^{p^m} \cdot f^{-1}$; // $I_k + M_k + F_k$
29 **else**
30 $\quad f \leftarrow f \cdot v_V(P)$; // $E_v + M_k$
31 $\quad f \leftarrow f \cdot (f^{p^m})^{-1}$; // $I_k + M_k + F_k$
32 **end**
33 $f \leftarrow f^{\left(\sum_{i=0}^{d-1} p^{im}\right)/r}$; // 最終べきの残りの計算
34 **return** f;

証明．一般に，変数 X と正整数 $n > 1$ について $X^n - 1 = \left(\sum_{i=0}^{n-1} X^i\right)(X - 1)$ である．また，$X^n - X = X(X^{n-1} - 1)$ となることに注意．証明したい補題に合わせて $n = d$ そして $X = p^m$ と置き，これら分解を利用する．$g \in \mathbb{F}_{p^k}^{\times}$ より $g^{p^k} = g$ である．埋め込み次数の定義から $r \nmid (p^m - 1)$ なので $r \mid \left(\sum_{i=0}^{d-1} p^{im}\right)$ である．これらから:

$$g^{-1\cdot\frac{q^k-1}{r}} = g^{-1\cdot(p^m-1)\cdot\frac{\sum_{i=0}^{d-1}p^{im}}{r}} = g^{(1-p^m)\cdot\frac{\sum_{i=0}^{d-1}p^{im}}{r}}$$
$$= g^{(p^{dm}-p^m)\cdot\frac{\sum_{i=0}^{d-1}p^{im}}{r}} = g^{(\sum_{i=1}^{d-1}p^{im})\cdot(p^m-1)\cdot\frac{\sum_{i=0}^{d-1}p^{im}}{r}}$$
$$= g^{(\sum_{i=1}^{d-1}p^{im})\cdot\frac{p^k-1}{r}},$$

が成り立つ． □

補題 1 によって得られる逆元演算を置き換えた式は，d が大きくなると計算コストが増加すると考えられる．よって，逆元演算と計算速度を比較し，高速な方を使用すべきである．

4.2.2 演算回数

Algorithm 2 の第 33 行目以外の各行末の式は，その行を計算するために実行する演算の回数を示す．M_k, S_k, I_k, F_k はそれぞれ $\mathbb{F}_{p^k}$ 上の乗算，平方算，逆元算，フロベニウス写像の計算を 1 回行うことを表し，E_v と $E_{v'}$ はそれぞれ $v_Q(P)$ と $v'_Q(P)$ の計算を 1 回行うこと，D と A はそれぞれ有理点群演算と関数値の評価を複合した $(\ell'_{Q,Q}(P), [2]Q)$，そして $(\ell'_{V,\pm Q}(P), V \pm Q)$ の計算を 1 回行うことを表す．第 33 行目の計算に必要な演算は，使用する楕円曲線の埋め込み次数 k の値と，$k = dm$ と分解した時の d と m の値によって定まる．本稿では 3 次ツイストを持つ楕円曲線を使用するため $d = 3$ としたが，使用する楕円曲線を限定しないため m および k が定まらず，その演算回数は明確に記述できない．そのため，第 33 行目の計算に必要な演算の回数を抽象化して R と記述する．L を命題とする時，$[L]$ は L が真ならば 1，偽ならば 0 とする．また，$S_\pm := \{i \mid s_i \neq 0 \wedge i \in \{0, \ldots, \mathtt{mb}-2\}\}$，$S_- := \{i \mid s_i = -1 \wedge i \in \{0, \ldots, \mathtt{mb}-2\}\}$，$\Delta' := \{i \mid \delta_i = 1 \wedge i \in \{0, \ldots, \mathtt{mb}-1\}\}$ とする．ある RSBE に対する Algorithm 2 の計算に必要な演算回数は次の式で推定できる:

$$\begin{aligned}
&\big([\neg(s_i \neq -1 \text{ for all } i \in \{0, \ldots, \mathtt{mb}\})] \cdot E_{v'}\big) \\
&+ \big(\#\Delta' \cdot (E_v + 2M_k + 2F_k)\big) \\
&+ \big([s_{\mathtt{mb}} \neq 0] \cdot D + [s_{\mathtt{mb}} = -1] \cdot (M_k + S_k)\big) \\
&+ \big([s_{\mathtt{mb}-1} \neq 0] \cdot (A + M_k) + [s_{\mathtt{mb}-1} = -1] \cdot M_k\big) \\
&+ \big((\mathtt{mb}-1) \cdot (D + M_k + S_k)\big) \\
&+ \big(\#S_\pm \cdot (A + M_k) + \#S_- \cdot M_k\big) \\
&\quad + \big(I_k + M_k + F_k + [\delta_{-1} = 1] \cdot (E_v + M_k)\big) + R. \quad (1)
\end{aligned}$$

式 (1) が示すように，$\#\Delta'$ と $\#S_-$ が大きい程，演算回数が増加するため，演算回数ができるだけ小さな RSBE を構成すべきである．また，本稿では疎乗算などの個々の演算の効率化を考慮していないことに注意．

4.3 新しいパラメータとその冗長符号付き 2 進数展開

この節では，BLS15 と BLS21 についてパラメータの探索を行った結果について述べる．探索方針は，パラメータの符号付き 2 進数展開のハミング重みが小さく，既約二項式で $\mathbb{F}_{p^k}$ を構成でき，達成する安全性が論文 [5], [16], [27] と同等なパラメータである．後者については厳密な評価は困難である．これまでの研究 [4], [5], [11], [15] から，安全性の評価については $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathrm{T}$ の離散対数問題の困難性が重要であり，その解読計算量は群位数 r や体位数 p の大きさが漸近的に支配的なパラメータであると考えられる．そこで本稿では，最上位ビットの位置を揃えることで同等の安全性を達成するものとする．具体的には，BLS15 の場合はパラメータの最上位ビットの位置を 32，BLS21 は 43 とする．また，この節では Algorithm 2 を効率的に計算するための RSBE についても考察する．

表 1 BLS15 について発見したパラメータ

$-2^{12} + 2^{21} - 2^{28} + 2^{32}$	$2^2 + 2^{17} - 2^{22} - 2^{32}$
$-2^3 - 2^{14} - 2^{20} - 2^{32}$	$2^6 + 2^{17} - 2^{22} - 2^{32}$

注意 1. §4.2.2 で簡単に述べたように，本稿では Algorithm 2 の計算に必要な個々の演算の効率化は考慮していない．例えば，座標系 [1] や疎乗算 [26] など，これまでに示されている様々な効率化テクニックを考慮していない．また，使用する楕円曲線を限定しておらず，部品となる個々の演算の詳細やその実際の演算速度を明らかにしていないため，“実際に” 最も高速なペアリング演算を与える RSBE を議論することは難しい．精密に最も高速にペアリング演算を計算できる RSBE を算出するためには，その部品となる個々の演算の効率化とその計算速度の実測が必要である．つまり，本稿で考察する効率的に Algorithm 2 を計算できる RSBE の構成は，あくまでもおおまかな見積もりであることに注意．

4.3.1 BLS15 について

BLS15 については 4 つのパラメータを発見した．これを表 1 に示す．これらパラメータは，実装報告 [5], [12], [27] で使用されているパラメータよりもハミング重みが 1 小さい．

式 (1) より，$\#S_-$ が 1 減少すると演算回数が M_k 減少するが，$\#\Delta'$ が 1 増加すると演算回数が $E_v + 2M_k + 2F_k$ 増加する．しかし，$\delta_{-1} = 1$ の場合は演算回数の増加が $E_v + M_k$ だけで済む．よって，表 1 の 4 つのパラメータのうち，符号付き 2 進数展開の非ゼロビットが全て負の $z = -2^3 - 2^{14} - 2^{20} - 2^{32}$ について，その RSBE を $\delta_{-1} = 1$，他の δ_i は全て 0，$s_3 = s_{14} = s_{20} = s_{32} = 1$ とし，他の s_i は全て 0 とすることで，128 ビット安全性の BLS15 における最も高速な ate ペアリングを構成できると考えられる．この RSBE を使用した場合の Algorithm 2 の計算に必要な演算の実行回数は，$3A + 32D + E_v + F_k + I_k + 36M_k + 31S_k + R$ となる．

4.3.2 BLS21 について

BLS21 については，$z = 2^{10} - 2^{29} - 2^{37} - 2^{43}$ を発見した．実装報告 [16] のパラメータとハミング重みが同じだ

が，実装報告 [16] で使用されているパラメータは既約二項式で $\mathbb{F}_{p^k}$ を構成するための必要条件を満たさない．発見したパラメータは必要条件を満たす．このパラメータについては，次の 2 つの RSBE が最も高速な ate ペアリングの候補として挙げることができる:

候補 1 z の RSBE を $\delta_{-1} = 1$，他の δ_i は全て 0，$s_{29} = s_{37} = s_{43} = 1$，そして $s_{10} = -1$ とし，他の s_i は全て 0 とする．この RSBE に対する Algorithm 2 の計算に必要な演算の実行回数は，$3A + 43D + E_v + E_{v'} + F_k + I_k + 48M_k + 42S_k + R$ である．

候補 2 z の RSBE を $\delta_{28} = 1$, 他の δ_i は全て 0, $s_{10} = s_{29} = s_{37} = s_{43} = 1$ とし, 他の s_i は全て 0 とする．この RSBE については, $3A+43D+E_v+3F_k+I_k+48M_k+42S_k+R$ が Algorithm 2 の計算に必要な演算の実行回数である．

演算回数について，候補 1 から候補 2 を引くと，$E_{v'} - 2F_k$ を得る．個々の演算の実際の計算時間をこの式に代入した時，負となるならば候補 1 が，正となるならば候補 2 が，256 ビット安全性の BLS21 における最も高速に Algorithm 2 を計算可能とする RSBE であると考えられる．

5. まとめ

本稿では符号付き 2 進数展開を拡張した冗長符号付き 2 進数展開 (RSBE) を定義し，その効率的なアルゴリズムを Algorithm 2 に示した．その結果，効率的にペアリング演算を計算可能なパラメータが存在する空間が従来よりも大きいことを示した．この結果に従い，BLS15 と BLS21 についてパラメータの探索を行った．そして § 4.3 に示すように，実際に高速実装に有利なパラメータを新しく発見した．このように，提案アルゴリズムによって 3 次ツイストのペアリング演算は従来よりも高速なパラメータの存在と，さらなる高速化の可能性が示された．さらに，効率的なパラメータ空間が拡大したことで，Barreto ら [6] や Barbulescu と Duquesne [4] が議論している特別な条件を満たす楕円曲線のパラメータや，SNARK 方式 [8] に適した効率的なパラメータが選択できる可能性の向上が期待できる．

今後の課題は，個々の演算の効率化，他の曲線族の効率的なパラメータの探索，最終べきの効率化および高速化，そして計算速度を実測し比較および評価することである．使用する楕円曲線が定まった時，最も高速なペアリング演算を与える RSBE を自動あるいは半自動的に構成する方法の確立も今後の課題である．また，twisted Ate ペアリングおよび Eil ペアリングの効率的な計算方法や，スカラー倍，べき乗など，他の様々な演算の効率化および高速実装も今後の課題である．

謝辞

本研究の一部は，JSPS 科研費 JP19H01109 の助成および JST CREST JPMJCR19F6 の支援を受けたものです．本研究について，林田大輝氏と早坂健一郎氏より有益なコメント・アドバイスを頂きました．この場を借りて深く御礼申し上げます．

参考文献

[1] Aranha, D. F., Barreto, P. S. L. M., Longa, P. and Ricardini, J. E.: The Realm of the Pairings, *SAC 2013*, pp. 3–25 (2013).

[2] Aranha, D. F., Karabina, K., Longa, P., Gebotys, C. H. and Hernandez, J. L.: Faster Explicit Formulas for Computing Pairings over Ordinary Curves, *EUROCRYPT 2011*, pp. 48–68 (2011).

[3] Aranha, D. F., Knapp, E., Menezes, A. and Rodríguez-Henríquez, F.: Parallelizing the Weil and Tate Pairings, *IMACC 2011*, pp. 275–295 (2011).

[4] Barbulescu, R. and Duquesne, S.: Updating Key Size Estimations for Pairings, *J. Cryptology*, Vol. 32, No. 4, pp. 1298–1336 (2019).

[5] Barbulescu, R., El Mrabet, N. and Ghammam, L.: A taxonomy of pairings, their security, their complexity, Cryptology ePrint Archive, Report 2019/485 (2019).

[6] Barreto, P. S. L. M., Costello, C., Misoczki, R., Naehrig, M., Pereira, G. C. C. F. and Zanon, G.: Subgroup Security in Pairing-Based Cryptography, *LATINCRYPT 2015*, pp. 245–265 (2015).

[7] Barreto, P. S. L. M., Lynn, B. and Scott, M.: Efficient Implementation of Pairing-Based Cryptosystems, *J. Cryptology*, Vol. 17, No. 4, pp. 321–334 (2004).

[8] Ben-Sasson, E., Chiesa, A., Genkin, D., Tromer, E. and Virza, M.: SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge, *CRYPTO 2013*, pp. 90–108 (2013).

[9] Beuchat, J., González-Díaz, J. E., Mitsunari, S., Okamoto, E., Rodríguez-Henríquez, F. and Teruya, T.: High-Speed Software Implementation of the Optimal Ate Pairing over Barreto–Naehrig Curves, *Pairing 2010*, pp. 21–39 (2010).

[10] Costello, C., Lange, T. and Naehrig, M.: Faster Pairing Computations on Curves with High-Degree Twists, *PKC 2010*, pp. 224–242 (2010).

[11] El Mrabet, N. and Joye, M.(eds.): *Guide to Pairing-Based Cryptography*, Chapman and Hall/CRC (2016).

[12] Fouotsa, E., El Mrabet, N. and Pecha, A.: Optimal Ate Pairing on Elliptic Curves with Embedding Degree 9, 15 and 27, `https://arxiv.org/abs/2002.11920` (2020).

[13] Freeman, D., Scott, M. and Teske, E.: A Taxonomy of Pairing-Friendly Elliptic Curves, *J. Cryptology*, Vol. 23, No. 2, pp. 224–280 (2010).

[14] Fuentes-Castañeda, L., Knapp, E. and Rodríguez-Henríquez, F.: Faster Hashing to $\mathbb{G}_2$, *SAC 2011*, pp. 412–430 (2011).

[15] Guillevic, A.: A Short-List of Pairing-Friendly Curves Resistant to Special TNFS at the 128-Bit Security Level, *PKC 2020*, pp. 535–564 (online), DOI: 10.1007/978-3-030-45388-6_19 (2020).

[16] 林田 大輝，早坂 健一郎：BLS-21 曲線を用いた効率的なペアリング計算，SCIS 2020, 4A1-5 (2020).

[17] Hayashida, D., Hayasaka, K. and Teruya, T.: Efficient Final Exponentiation via Cyclotomic Structure for Pairings over Families of Elliptic Curves, Cryptology ePrint Archive, Report 2020/875 (2020).

[18] Hess, F.: Pairing Lattices, *Pairing 2008*, pp. 18–38 (2008).

[19] Hess, F., Smart, N. P. and Vercauteren, F.: The Eta Pairing Revisited, *IEEE Trans. Information Theory*, Vol. 52, No. 10, pp. 4595–4602 (2006).
[20] Kim, T. and Barbulescu, R.: Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case, *CRYPTO 2016*, pp. 543–571 (2016).
[21] Kim, T. and Jeong, J.: Extended Tower Number Field Sieve with Application to Finite Fields of Arbitrary Composite Extension Degree, *PKC 2017*, pp. 388–408 (2017).
[22] Kinoshita, K. and Suzuki, K.: Accelerating beta Weil pairing with precomputation and multi-pairing techniques, *IWSEC 2020* (2020). To appear.
[23] Le, D. and Tan, C. H.: Speeding Up Ate Pairing Computation in Affine Coordinates, *ICISC 2012*, pp. 262–277 (2012).
[24] Lin, X., Zhao, C., Zhang, F. and Wang, Y.: Computing the Ate Pairing on Elliptic Curves with Embedding Degree $k = 9$, *IEICE Trans. Fundamentals*, Vol. 91-A, No. 9, pp. 2387–2393 (2008).
[25] Miller, V. S.: The Weil Pairing, and Its Efficient Calculation, *J. Cryptology*, Vol. 17, No. 4, pp. 235–261 (2004).
[26] Mori, Y., Akagi, S., Nogami, Y. and Shirase, M.: Pseudo 8-Sparse Multiplication for Efficient Ate-Based Pairing on Barreto-Naehrig Curve, *Pairing 2013*, pp. 186–198 (2013).
[27] Nanjo, Y., Kodera, Y., Matsumura, R., Shirase, M., Kusaka, T. and Nogami, Y.: Evaluation of Pairing on Elliptic Curves with Embedding Degree 15 with Type-II All-one Polynomial Extension Field of Degree 5, SCIS 2020, 4A1-3 (2020).
[28] Nanjo, Y., Shirase, M., Kusaka, T. and Nogami, Y.: A Technique for Fast Miller's Algorithm of Ate Pairings on Elliptic Curves with Embedding Degrees of Multiple of Three, *ITC-CSCC 2020*, pp. 283–287 (2020).
[29] Scott, M., Benger, N., Charlemagne, M., Perez, L. J. D. and Kachisa, E. J.: On the Final Exponentiation for Calculating Pairings on Ordinary Elliptic Curves, *Pairing 2009*, pp. 78–88 (2009).
[30] Teruya, T.: A Note on Subgroup Security in Pairing-Based Cryptography, *APKC 2018*, pp. 35–43 (2018).
[31] Teruya, T., Saito, K., Kanayama, N., Kawahara, Y., Kobayashi, T. and Okamoto, E.: Constructing Symmetric Pairings over Supersingular Elliptic Curves with Embedding Degree Three, *Pairing 2013*, pp. 97–112 (2013).
[32] Vercauteren, F.: Optimal pairings, *IEEE Trans. Inf. Theory*, Vol. 56, No. 1, pp. 455–461 (2010).
[33] Zhang, X. and Lin, D.: Analysis of Optimum Pairing Products at High Security Levels, *INDOCRYPT 2012*, pp. 412–430 (2012).