SCIS 2016 2016 Symposium on
Cryptography and Information Security
Kumamoto, Japan, Jan. 19 - 22, 2016
The Institute of Electronics,
Information and Communication Engineers

種数2の超楕円曲線に対するGaudry-Schostの位数計算法の高速化

Speeding up the Gaudry-Schost point counting algorithm for genus 2 hyperelliptic curves

松尾 和人 *
Kazuto Matsuo

あらまし 有限体上の種数2の超楕円曲線を用いた安全な超楕円曲線暗号の構成法として知られるGaudryとSchostのℓ進位数計算法では等分多項式の効率的な計算法が利用されている。本論文ではGaudryとSchostの等分多項式計算の高速化手法を提案する。また、現実的なサイズの位数計算に必要となる等分多項式計算が提案手法により2割弱程度高速化されることを実装実験によって示す。さらに、提案手法を適用したGaudryとSchostのℓ進位数計算法を用いて96bitと128bitの有限素体上の種数2の超楕円曲線の位数計算を行った結果を示す。

キーワード 超楕円曲線暗号、種数2の超楕円曲線、ℓ進位数計算、等分多項式

1 はじめに

奇標数pの有限体$\mathbb{F}_q$上の種数2の超楕円曲線

$$
\begin{aligned}
C : Y^2 &= F(X) \\
F(X) &= X^5 + f_3X^3 + \cdots + f_1X + f_0 \qquad (1)
\end{aligned}
$$

のJacobian群$\mathcal{J}_C(\mathbb{F}_q)$を用いた超楕円曲線暗号[15]は長い間研究されているが、素体をはじめとする大標数の有限体上の安全な曲線を豊富に得ることは未だに難しい。大標数の有限体上安全な超楕円曲線の構成法の一つにAlgorithm 1に示す位数計算法を用いる方法がある。

Algorithm 1 安全な超楕円曲線の構成

```
1: repeat
2:    repeat
3:       C をランダムに選択
4:       #J_C(F_q) を計算（位数計算法を利用）
5:    until #J_C(F_q): almost-prime
6: until J_C(F_q) 上の DLP は既知の解法で解けない
7: return C
```

この構成法に必要となる位数計算法にはSchoofアルゴリズム[23, 24]の拡張としてPila [21]によって提案されたℓ進位数計算法と一般の有限アーベル群に適用可能なsquare-root法の変形であるSutherlandアルゴリズム[26, 27, 28, 29]等がある。本論文ではより広範囲な曲線に対して適用可能なℓ進位数計算法を扱う。

* 神奈川大学理学部情報科学科, 神奈川県平塚市土屋 2946, Dept. of Information Sciences, Faculty of Science, Kanagawa Univ., 2946, Tsuchiya, Hiratsuka-shi, Kanagawa 259-1293, Japan.

超楕円曲線に対するℓ進位数計算法はこれまでに様々な改良[14, 1, 2, 8, 19, 9, 11]が行われてきたが、特にGaudryとHarley [8]は位数計算に利用可能な等分多項式を構成し、超楕円曲線に対するℓ進位数計算法を初めて実装した。また、GaudryとSchost [9, 11]は等分多項式計算の改良を行いより効率的な位数計算法を提案した。これまでにℓ進位数計算の実装評価も幾つか知られており、GaudryとSchost [11]の（特殊な曲線ではあるものの）254bit位数（最大素因数は250bit）の安全な種数2の超楕円曲線の生成がこれまでの最高記録である。

本論文ではℓ進位数計算を用いた安全な種数2の超楕円曲線の構成のためにGaudryとSchostが提案した等分多項式の計算の高速化手法を提案する。また、提案手法を適用したℓ進位数計算法を用いて96bitと128bitの有限素体上の種数2の超楕円曲線の位数計算を行う。

本論文は第2節で式(1)で与えた種数2の超楕円曲線のJacobian群$\mathcal{J}_C(\mathbb{F}_q)$を定義するとともにその位数$\#\mathcal{J}_C(\mathbb{F}_q)$の持つ性質を述べ、$\ell$進位数計算法を概説する。また、第3節ではGaudryとSchostのℓ進位数計算法を説明する。そして、第4節でGaudryとSchostが提案した等分多項式の計算の高速化手法を提案し、第5節では提案手法の効率について考察する。第6節では96bitと128bitの有限素体上の種数2の超楕円曲線の位数計算を行った結果について述べる。

本論文では$\mathbb{F}_q$上のd次多項式の乗算に必要な計算量を$M(d)$と表す。

2　準備

2.1　種数 2 の超楕円曲線の因子と Jacobian

式 (1) で与えられた C 上の有限個の点 P_i の型式和 $\mathcal{D}$ を下式で定義する。

$$\mathcal{D} = \sum_i m_i P_i - \left(\sum_i m_i\right) P_\infty, \tag{2}$$
$$P_i = (x_i, y_i) \in C \setminus \{P_\infty\},\ m_i > 0,\ \sum_i m_i \le 2.$$

ここで、P_∞ は無限遠点を表す。また、$i \neq j$ に対し $P_i \neq \tilde{P}_j$、$F(x_i) = 0$ となる P_i に対し $m_i \le 1$ とする。式 (2) の形で与えられる $\mathcal{D}$ を C の被約因子という。C の被約因子 $\mathcal{D}$ は以下の 4 タイプに分類される。

$$\mathcal{D} = \begin{cases} 0 & \\ P_1 - P_\infty & \text{(Type I)} \\ 2P_1 - 2P_\infty,\ y_1 \neq 0 & \text{(Type II)} \\ P_1 + P_2 - 2P_\infty,\ x_1 \neq x_2 & \text{(Type III)} \end{cases} \tag{3}$$

ここで、$P_1 = (x_1, y_1),\ P_2 = (x_2, y_2) \in C$ である。C の被約因子を Mumford 表現と呼ばれる $\overline{\mathbb{F}_q}$ 上の多項式の組

$$\mathcal{D} = (U, V) \in \overline{\mathbb{F}_q}[x]^2$$

で一意に表現可能である。ここで、Type I の因子は $U = X - x_1,\ V = y_1$ であり、Type II の因子に対し、$U = (X - x_1)^2,\ V(x_1) = y_1,\ U \mid V^2 - F,\ \deg V \le 1$、Type III の因子に対し、$U = (X - x_1)(X - x_2),\ V(x_i) = y_i$ (i=1,2), $U \mid V^2 - F,\ \deg V \le 1$ である。被約因子の集合を $\mathcal{J}_C$ と書き、C の Jacobian と呼ぶ。

式 (2) で与えた $\mathcal{J}_C$ の元 $\mathcal{D}$ に対し Frobenius 写像 ϕ を以下で定義する。

$$\phi : \mathcal{D} \in \mathcal{J}_C \mapsto \mathcal{D}^q \in \mathcal{J}_C \tag{4}$$

ここで、$\mathcal{D}^q = \sum_i m_i P_i^q - (\sum_i m_i) P_\infty \in \mathcal{J}_C,\ P_i^q = (x_i^q, y_i^q) \in C \setminus \{P_\infty\}$ とする。Frobenius 写像 ϕ で固定される $\mathcal{J}_C$ の元の集合

$$\mathcal{J}_C(\mathbb{F}_q) = \{\mathcal{D} \in \mathcal{J}_C \mid \mathcal{D} = \phi(\mathcal{D})\}$$

を Jacobian 群と呼ぶ。$\mathcal{D} \in \mathcal{J}_C(\mathbb{F}_q)$ とその Mumford 表現 $\mathcal{D} = (U, V)$ に現れる多項式 U, V が $\mathbb{F}_q$ 係数であることは同値である。$\mathcal{J}_C(\mathbb{F}_q)$ は有限可換群となり、この上の離散対数問題に基づく暗号系を構成可能である。

式 (4) で与えた ϕ の特性多項式は 4 次の整数係数多項式

$$\chi = X^4 - s_1 X^3 + s_2 X^2 - s_1 p X + p^2 \in \mathbb{Z}[X]$$

となる [16, Theorem 5.1]。すなわち、任意の $\mathcal{D} \in \mathcal{J}_C$ に対して

$$\phi^4(\mathcal{D}) - [s_1]\phi^3(\mathcal{D}) + [s_2]\phi^2(\mathcal{D}) - [s_1 q]\phi(\mathcal{D}) + [q^2]\mathcal{D} = 0 \tag{5}$$

を満足する $(s_1, s_2) \in \mathbb{Z}^2$ が C に対して一意に存在する。この (s_1, s_2) は

$$-\lfloor 4\sqrt{q} \rfloor \quad \le s_1 \le \quad \lfloor 4\sqrt{q} \rfloor, \tag{6}$$
$$\lceil 2\sqrt{q}|s_1| - 2q \rceil \quad \le s_2 \le \quad \left\lfloor \frac{1}{4}s_1^2 + 2q \right\rfloor \tag{7}$$

を満足する [22, 5, 20]。

Frobenius 写像の特性多項式 χ から $\#\mathcal{J}_C(\mathbb{F}_q) = \chi(1)$ によって位数 $\#\mathcal{J}_C(\mathbb{F}_q)$ を求めることができる [16, Theorem 5.1]。すなわち、式 (5) を満足する $(s_1, s_2) \in \mathbb{Z}^2$ を求めることで位数が得られる。

2.2　ℓ 進位数計算法のアウトライン

ℓ 進位数計算法は与えられた C に対して、Frobenius 写像の特性多項式 χ を求める方法である。しかし、χ の係数候補数 $\#\{(s_1, s_2)\} = O(q^{3/2})$ なので、暗号に利用される q に対して、式 (5) を満足する (s_1, s_2) を直接探索するのは難しい。そこで $\mathcal{J}_C$ のねじれ群 $\mathcal{J}_C[\ell]$ と中国式剰余定理 [12, Theorem 5.7] を利用して (s_1, s_2) を効率的に求めている。以下では ℓ 進位数計算法のアウトラインを示す。

$n \in \mathbb{N}$ に対して、$\mathcal{J}_C$ の n ねじれ群を

$$\mathcal{J}_C[n] = \{\mathcal{D} \in \mathcal{J}_C \mid [n]\mathcal{D} = 0\} \subset \mathcal{J}_C$$

と定義する。素数 ℓ に対する ℓ ねじれ群を考え、任意の $\mathcal{D} \in \mathcal{J}_C[\ell]$ に対して式 (5) を満足する $(s_1, s_2) \in \mathbb{F}_\ell^2$ を求めれば、$\chi_\ell \equiv \chi \bmod \ell$ が得られたこととなる。ここで、$\#\{(s_1, s_2) \in \mathbb{F}_\ell^2\} = \ell^2$ なので、ℓ が十分に小さければ、全数探索や baby-step giant-step 法によって $(s_1, s_2) \in \mathbb{F}_\ell^2$ を求められる。これを、式 (6), (7) で与えた (s_1, s_2) の取りうる範囲に対して十分な個数の異なる ℓ に対して行えば、得られた χ_ℓ から中国式剰余定理によって χ を得ることが可能である。Algorithm 2 に ℓ 進位数計算法の概要を示す。

Algorithm 2 ℓ 進位数計算法

Input: $\mathbb{F}_q$ 上の種数 2 の超楕円曲線 C
Output: $\mathcal{J}_C$ の Frobenius 写像 ϕ の特性多項式 $\chi \in \mathbb{Z}[X]$

1: 下式を満足する素数 $\ell_{\max}$ と $m \in \mathbb{Z}$ を計算：

$$m = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdots \ell_{\max} > 4q$$

2: **for** 素数 $\ell \in \{2, 3, 5, \ldots, \ell_{\max}\}$ **do**
3: 　$\chi_\ell \equiv \chi \bmod \ell$ を計算
4: 中国式剰余定理により、$\{\chi_\ell\}$ から $\chi \bmod m$ を計算
5: $\chi \bmod m$ から $\chi \in \mathbb{Z}[X]$ を計算

現在の一般のコンピュータの計算能力では、Algorithm 2 のみで χ を得るのは難しい。例えば、q が 80 ビットの場合 $\ell_{\max} = 67$ であるが、これまでに知られている結果では 256bit の q に対する $\ell_{\max} = 31$ が最大値である

[11]。そこで、実際には、計算可能な場合には $\chi \bmod \ell^k$ や $\chi \bmod p$ も計算し、より大きな法 m に対する $s_1 \bmod m$, $s_2 \bmod m$ を得たうえで、最終的に square-root 法や、より効率的な多次元 square-root 法 [19, 13, 10] を用いて $\chi \in \mathbb{Z}[X]$ を得ている。

3 Gaudry-Schost アルゴリズム

3.1 ℓ 等分点の計算

ℓ 進位数計算法では、任意の $\mathcal{D} \in \mathcal{J}_C[\ell] \subset \mathcal{J}_C$ に対して式 (5) を満足する $(s_1, s_2) \in \mathbb{F}_\ell^2$ を求める必要がある。そのため $\mathcal{J}_C$ の中から $[\ell]\mathcal{D} = 0$ を満足する $\mathcal{D}$ を発見する方法が必要となる。楕円曲線に対しては、$\mathcal{D} \in \mathcal{J}_C[\ell]$ の発見に利用可能な ℓ 等分多項式 [17, Chapter II] が知られている。Schoof はこの ℓ 等分多項式を利用して楕円曲線の位数計算法を構成した。一方、超楕円曲線に対する完全な ℓ 等分多項式は知られていなかったものの、式 (3) の Type I に対する ℓ 等分多項式は Cantor [4] によって得られていた。しかし、$\mathcal{J}_C[\ell]$ の被約因子の殆どは Type III であり Type I の被約因子は殆ど無いことから Cantor の ℓ 等分多項式をそのまま用いた位数計算は現実的ではない。

Gaudry と Harley [8] は Cantor の結果を利用して式 (3) の Type III の被約因子対する ℓ 等分多項式を得ることに成功し、これを用いて超楕円曲線の ℓ 進位数計算に初めて成功した。さらに、Gaudry と Schost [9, 11] は Gaudry と Harley の結果を改良し、ℓ 進位数計算法による安全な超楕円曲線の構成に成功した。本節では、Cantor の n 倍公式, Gaudry-Harley, Gaudry-Schost の ℓ 等分多項式とその導出法の概略を紹介する。

3.2 Cantor の n 倍公式と Gaudry-Harley の等分多項式

Cantor は $n \in \mathbb{N}$ と Type I の被約因子 $P = (X - x_P, y_P) \in \mathcal{J}_C(\overline{\mathbb{F}_q})$ に対して、P の n 倍公式

$$[n]P = \left(X^2 + \frac{d_1^{(n)}(x_P)}{d_2^{(n)}(x_P)} X + \frac{d_0^{(n)}(x_P)}{d_2^{(n)}(x_P)}, y_P \left(\frac{e_1^{(n)}(x_P)}{e_2^{(n)}(x_P)} X + \frac{e_0^{(n)}(x_P)}{e_2^{(n)}(x_P)} \right) \right) \tag{8}$$

を満足する多項式 $d_0^{(n)},\ d_1^{(n)},\ d_2^{(n)},\ e_0^{(n)},\ e_1^{(n)},\ e_2^{(n)} \in \mathbb{F}_q[X]$ を求めた。ここで、奇素数 ℓ に対して、$\deg d_0^{(\ell)} = 2\ell^2 - 1$, $\deg d_1^{(\ell)} = 2\ell^2 - 2$, $\deg d_2^{(\ell)} = 2\ell^2 - 3$, $\deg e_0^{(\ell)} = 3\ell^2 - 2$, $\deg e_1^{(\ell)} = 3\ell^2 - 3$, $\deg e_3^{(\ell)} = 3\ell^2 - 2$ である。また、$d_2^{(\ell)}(x_P) \neq 0$ である。以下では $d = \deg d_0^{(\ell)}$ とする。$d = O(\ell^2)$ である。

Gaudry と Harley は、式 (8) を用いて以下のように一般の ℓ 等分点を求めた。

$P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2) \in C(\overline{\mathbb{F}_q})$ に対して $\mathcal{D} = P_1 + P_2 - 2P_\infty \in \mathcal{J}_C[\ell]$ としたとき、$\mathcal{D}$ が ℓ 等分点であるとすると、$[\ell]\mathcal{D} = 0$ より

$$[\ell](P_1 - P_\infty) = -[\ell](P_2 - P_\infty)$$

を得る。いま、

$$\begin{aligned} E_1(X_1, X_2) &= \frac{d_1^{(\ell)}(X_1) d_2^{(\ell)}(X_2) - d_1^{(\ell)}(X_2) d_2^{(\ell)}(X_1)}{X_1 - X_2}, \\ E_2(X_1, X_2) &= \frac{d_0^{(\ell)}(X_1) d_2^{(\ell)}(X_2) - d_0^{(\ell)}(X_2) d_2^{(\ell)}(X_1)}{X_1 - X_2}, \\ E_3(X_1, X_2) &= \frac{e_0^{(\ell)}(X_1) e_1^{(\ell)}(X_2) + e_0^{(\ell)}(X_2) e_1^{(\ell)}(X_1)}{X_1 - X_2} \end{aligned} \tag{9}$$

と定め、

$$E_1(X_1, X_2) = E_2(X_1, X_2) = E_3(X_1, X_2) = 0 \tag{10}$$

の解を $(X_1, X_2) = (x_1, x_2)$ とする。そして、x_1, x_2 に対して、$y_1^2 = F(x_1)$, $y_2^2 = F(x_2)$ を満足する y_1, y_2 を定め、$P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, $\tilde{P}_2 = (x_2, -y_2) \in C$ とすれば、$\mathcal{D} = P_1 + P_2 - 2P_\infty$ か $\mathcal{D} = P_1 + \tilde{P}_2 - 2P_\infty$ のどちらか一方は ℓ 等分点になる。

Gaudry と Harley [8] は X_1 の多項式である ℓ 等分多項式 R_{GH} を、X_2 に関する終結式 Res_{X_2} によって、以下のように得た。

$$\begin{aligned} &R_1 = \mathrm{Res}_{X_2}(E_1, E_2),\ R_2 = \mathrm{Res}_{X_2}(E_1, E_3) \in \mathbb{F}_q[X_1] \\ &R_{GH} = \gcd(R_1, R_2) \in \mathbb{F}_q[X_1] \end{aligned} \tag{11}$$

R_{GH} の根は C の Type III の ℓ 等分点の X 座標を含むので、R_{GH} から $\mathcal{D} \in \mathcal{J}_C[\ell]$ を得ることができる。本来は Type III の $\mathcal{D}$ を構成する 2 点 P_1, P_2 のうちの 1 点の X 座標のみを根に持てば十分であり、その場合には $\deg R_{GH} = (\ell^4 - 1)/2$ となるが、実際には P_1, P_2 の X 座標を根とし、$d_2^{(\ell)}$ の根の一部も根とするため、$\deg R_{GH}$ は理想値の 2 倍以上になる。

3.3 Gaudry-Schost の等分多項式

Gaudry と Schost [9, 11] は式 (11) で与えた Gaudry-Harley の等分多項式 R_{GH} の改良を行った。改良の一つは、R_{GH} の根集合から「パラサイト」と呼ばれる $d_2^{(\ell)}$ の根を除去し R_{GH} の次数を下げることにある。さらに、E_t が X_1, X_2 の対称式であることに着目し、基本対称式変形により、Type III を構成する C の 2 点 P_1, P_2 のうちの 1 点の X 座標のみを根に持ち次数が $\deg R_{GS} = (\ell^4 - 1)/2$ である等分多項式 R_{GS} を得た。以下ではこの改良の概略を紹介する。

$\mathcal{D} = P_1 + P_2 - P_\infty = (U, V) \in \mathcal{J}_C[\ell]$ に対し $U = X^2 + U_1 X + U_0$ とすると、$U_0 = x_1 x_2$, $U_1 = -x_1 - x_2$ であり、P_1, P_2 の X 座標の対称式となる。そこで、$U_0 = X_1 X_2$, $U_1 = -X_1 - X_2$ と置き、$t = 1, 2, 3$ に対して $\mathfrak{E}_t \in \mathbb{F}_q[U_0, U_1]$ を

$$E_t(X_1, X_2) = \mathfrak{E}_t(X_1 X_2, -X_1 - X_2)$$

で定義した後に、$\mathfrak{E}_t$ に対して変数消去を行えば X_1, X_2 と違い、U_0, U_1 には対称性がないため得られる多項式

には R_{GH} に発生した不要な重複が発生しない。この $\mathfrak{E}_t$ を得るために、

$$
\begin{aligned}
A_{d_i}(X_1,X_2) &= (d_i^{(\ell)}(X_1)-d_i^{(\ell)}(X_2))/(X_1-X2)\\
B_{d_i}(X_1,X_2) &= (X_1 d_i^{(\ell)}(X_2)-X_2 d_i^{(\ell)}(X_1))/(X_1-X2)
\end{aligned}
$$

と置き、$\mathfrak{A}_{d_i},\mathfrak{B}_{d_i}\in\mathbb{F}_q[U_0,U_1]$ を

$$
\begin{aligned}
\mathfrak{A}_{d_i}(X_1X_2,-X_1-X_2) &= A_{d_i}(X_1,X_2),\\
\mathfrak{B}_{d_i}(X_1X_2,-X_1-X_2) &= B_{d_i}(X_1,X_2)
\end{aligned}
$$

と定義すると、

$$
\begin{aligned}
\mathfrak{E}_1(U_0,U_1) &= \mathfrak{A}_{d_1}(U_0,U_1)\mathfrak{B}_{d_2}(U_0,U_1)-\\
&\quad \mathfrak{A}_{d_2}(U_0,U_1)\mathfrak{B}_{d_1}(U_0,U_1),\\
\mathfrak{E}_2(U_0,U_1) &= \mathfrak{A}_{d_0}(U_0,U_1)\mathfrak{B}_{d_2}(U_0,U_1)-\\
&\quad \mathfrak{A}_{d_2}(U_0,U_1)\mathfrak{B}_{d_0}(U_0,U_1),\\
\mathfrak{E}_3(U_0,U_1) &= \mathfrak{A}_{e_0}(U_0,U_1)\mathfrak{B}_{e_1}(U_0,U_1)+\\
&\quad \mathfrak{A}_{e_1}(U_0,U_1)\mathfrak{B}_{e_0}(U_0,U_1) \qquad (12)
\end{aligned}
$$

が得られる。これらの式の終結式計算をパラサイトの除去とともに行うと、$\mathfrak{E}_1(u_0,u_1)=\mathfrak{E}_2(u_0,u_1)=\mathfrak{E}_3(u_0,u_1)=0$ を満足する u_1 を根に持つ等分多項式 $R_{GS}(U_1)\in\mathbb{F}_q[U_1]$ が得られる。この R_{GS} の根 u_1 は $\mathcal{D}\in\mathcal{J}_C[\ell]$ の Mumford 表現の第 1 成分 U の 1 次項の係数となる。

また、部分終結式を用いることで、U_0 を U_1 の多項式 $\mathfrak{U}_0(U_1) \bmod R_{GS}$ として表現可能である。R_{GS} の計算においては、

$$\mathfrak{R}(U_1)=\mathrm{res}_{U_0}(\mathfrak{E}_1(U_0,U_1),\mathfrak{E}_2(U_0,U_1)) \qquad (13)$$

を計算した後に、

$$R_{GS}=\gcd(\mathfrak{R}(U_1),\mathfrak{E}_3(\mathfrak{U}_0(U_1),U_1) \bmod \mathfrak{R})$$

として計算の効率化を図っている [1]。

3.4 Gaudry-Schost の等分多項式の計算

前節で述べた $\mathfrak{R}(U_1)\in\mathbb{F}_q[U_1]$ を得るには一般には計算が困難である対称式による変数変換が必要である。Gaudry と Schost は終結式の一般的な高速計算法に対称式変換を組み合わせることで効率的にこの計算を行っている。以下では $\mathfrak{R}(U_1)\in\mathbb{F}_q[U_1]$ の高速計算法の概略を述べる。

まず、$u_j\in\mathbb{F}_q$ を選択し、この u_j に対して

$$r_j=\mathrm{res}_{U_0}(\mathfrak{E}_1(U_0,u_j),\mathfrak{E}_2(U_0,u_j))\in\mathbb{F}_q \qquad (14)$$

を計算する。そのためには $\mathfrak{E}_1(U_0,u_j)$, $\mathfrak{E}_2(U_0,u_j)$ を計算する必要があるが、これについては追って説明する。$\mathfrak{E}_1(U_0,u_j)$, $\mathfrak{E}_2(U_0,u_j)$ から r_j を計算するために計算量が $O(M(d)\log d)$ の再帰的 GCD アルゴリズム [6, Chapter 11] を利用することが可能である。これを $O(d^2)$ 個の異なる u_j に対して計算し、得られた $O(d^2)$ 個の r_j から多項式補間を用いて $\mathfrak{R}(u_j)=r_j$ を満足する $\mathfrak{R}$ を計算可能である。r_j に必要な GCD の計算量は $O(d^2)$ 回全体で $O(d^2M(d)\log d)$ であり、多項式補間の計算量は $O(M(d^2)\log d)$ である。

以下では、残された課題である、与えられた $u_j\in\mathbb{F}_q$ に対する $\mathfrak{E}_1(U_0,u_j)$, $\mathfrak{E}_2(U_0,u_j)$ の計算を説明する。実際には式 (12) より与えられた $d_i^{(\ell)}$ に対する $\mathfrak{A}_{d_i}(U_0,u_j)$, $\mathfrak{B}_{d_i}(U_0,u_j)$ の計算法を与えれば十分である。まずはじめに、$i=0,1,2$ に対して $d_i^{(\ell)}$ から $K_i=d_i^{(\ell)}(X-u_j/2)\in\mathbb{F}_q[X]$ を求める。この計算は Aho-Steiglitz-Ullman アルゴリズム [3] により $O(M(d))$ で計算可能である。次に

$$K_i=K_{i,\mathrm{even}}(X^2)+XK_{i,\mathrm{odd}}(X^2)$$

を満足する $K_{i,\mathrm{even}}$, $K_{i,\mathrm{odd}}\in\mathbb{F}_q[X]$ を求める。すると、

$$
\begin{aligned}
d_i^{(\ell)} &\equiv K_{i,\mathrm{even}}(u_j^2/4-U_0)+\\
&\quad (X+u_j/2)K_{i,\mathrm{odd}}(u_j^2/4-U_0)\\
&\equiv \mathfrak{B}_{d_i}(U_0,u_j)X+\mathfrak{A}_{d_i}(U_0,u_j)\\
&\quad \bmod X^2+u_jX+U_0
\end{aligned}
$$

が成立する。したがって、$\mathfrak{A}_{d_i}(U_0,u_j)$, $\mathfrak{B}_{d_i}(U_0,u_j)$ が

$$
\begin{aligned}
\mathfrak{A}_{d_i}(U_0,u_j) &= K_{\mathrm{i,even}}(u_{j1}^2/4-U_0)+\\
&\quad u_jK_{i,\mathrm{odd}}(u_j^2/4-U_0)/2\in\mathbb{F}_q[U_0]\\
\mathfrak{B}_{d_i}(U_0,u_j) &= K_{i,\mathrm{odd}}(u_j^2/4-U_0)\in\mathbb{F}_q[U_0]
\end{aligned}
$$

を計算することで得られる。これは K_i の計算と同様に計算量 $O(M(d))$ の Aho-Steiglitz-Ullman アルゴリズムを 2 回適用することで計算可能である。

4 Gaudry-Schost の等分多項式計算の改良

本節では Gaudry-Schost の等分多項式 R_{GS} の計算法の改良を与える。具体的には 3.4 節で説明した $\mathfrak{R}(U_1)$ の計算のコスト削減を図る。

Gaudry-Schost は X_1, X_2 の基本対称式 $U_0=X_1X_2$, $U_1=-(X_1+X_2)$ を用いて、等分多項式の次数削減を実現した。提案手法では、Gaudry-Schost とは別の対称式を用いて計算効率を向上させる。

まず、

$$U_3=X_1-X_2 \qquad (15)$$

[1] さらに、$\mathcal{D}=(U,V)\in\mathcal{J}_C[\ell]$ の第 1 項の各係数を $U=U_1X+\mathfrak{U}_0(U_1) \bmod R_{GS}$ のように U_1 の多項式で表現可能である。同様に V の係数も U_1 の多項式による表現が可能であり、楕円曲線に対する位数計算法と同様に以降の計算を多項式演算によって行うことが可能である。

と置き、その 2 乗を U_2 とすると、

$$U_2 = U_3^2 = X_1^2 - 2X_1X_2 + X_2^2$$

より、U_2 は X_1, X_2 の対称式である。そこで、U_0 の代わりに U_2 を U_1 とともに用いて、Gaudry-Schost の方法で得られる等分多項式と同一の等分多項式 R_{GS} を計算することを考える。この手法では、X_1, X_2 を U_1, U_3 に変換した後に U_1, U_2 に変換することで線形変換を利用可能となり効率の向上が期待できる。

提案手法では、まず式 (9) の各 E_t に対して、

$$E_t(X_1, X_2) = G_t(X_1 - X_2, -X_1 - X_2)$$

を満足する $G_t \in \mathbb{F}_q[U_3, U_1]$ $(t = 1, 2)$ を考える。すると、

$$G_t(U_3, U_1) = E_t((U_3 - U_1)/2, -(U_3 + U_1)/2)$$

が成立する。したがって、

$$\begin{aligned} G_1(U_3, U_1) &= E_1((U_3 - U_1)/2, -(U_3 + U_1)/2) \\ &= (d_1^{(\ell)}((U_3 - U_1)/2)d_2^{(\ell)}((-U_3 - U_1)/2) - \\ &\quad d_1^{(\ell)}((-U_3 - U_1)/2)d_2^{(\ell)}((U_3 - U_1)/2))/U_3 \end{aligned} \tag{16}$$

$$\begin{aligned} G_2(U_3, U_1) &= E_2((U_3 - U_1)/2, -(U_3 + U_1)/2) \\ &= (d_0^{(\ell)}((U_3 - U_1)/2)d_2^{(\ell)}((-U_3 - U_1)/2) - \\ &\quad d_0^{(\ell)}((-U_3 - U_1)/2)d_2^{(\ell)}((U_3 - U_1)/2))/U_3 \end{aligned} \tag{17}$$

である。以上で得られた G_t は U_3 の奇数次の項を持たないので、G_t に対して

$$G_t(U_3, U_1) = H_t(U_3^2, U_1) \tag{18}$$

を満足する $H_t \in \mathbb{F}_q[U_2, U_1]$ を係数の移動によって計算可能である。この H_t から式 (13) と同一の多項式

$$\mathfrak{R} = \mathrm{res}_{U_2}(H_1, H_2)$$

が得られる。

実際の計算は Gaudry-Schost の計算と同様に、多数の $u_j \in \mathbb{F}_q$ に対して

$$r_j = \mathrm{res}_{U_2}(H_1(U_2, u_j), H_2(U_2, u_j)) \in \mathbb{F}_q$$

を計算し、得られた結果から多項式補間によって $\mathfrak{R}$ を得る。したがって、与えられた $u_j \in \mathbb{F}_q$ に対して $H_t(U_2, u_j)$ を計算する必要がある。そのために、まず式 (16), (17) より $G_t(U_3, u_j)$, $(t = 1, 2)$ を計算する。$G_t(U_3, u_j)$ の計算においては、まず $d_i^{(\ell)}$ から $d_i^{(\ell)}(U_3/2)$ を求める。この計算は各 i について $O(d)$ で計算可能である。次に、$d_i^{(\ell)}(U_3/2)$ から $d_i^{(\ell)}(U_3/2 - u_j/2)$ を求める。この計算は Aho-Steiglitz-Ullman アルゴリズムによって、各 i について $O(M(d))$ で計算可能である。さらに、$d_2^{(\ell)}(U_3/2 - u_j/2)$ から $d_2^{(\ell)}(-U_3/2 - u_j/2)$ を求める。この計算は $O(d)$ で計算可能である。そして、

$$\begin{aligned} g_1(U_3) &= d_1^{(\ell)}(U_3/2 - u_j/2)d_2^{(\ell)}(-U_3/2 - u_j/2) \\ g_2(U_3) &= d_0^{(\ell)}(U_3/2 - u_j/2)d_2^{(\ell)}(-U_3/2 - u_j/2) \end{aligned}$$

を計算する。この計算は $O(M(d))$ の乗算 2 回で計算可能である。これらの g_t から各 G_t が

$$G_t(U_3, u_j) = (g_t(U_3) - g_t(-U_3))/U_3 \tag{19}$$

と計算される。これは $O(d)$ の多項式係数演算、$O(d)$ の多項式減算及び $O(d)$ の多項式係数シフトにより実現される。最後に式 (18) より H_t を計算する。これは $O(d)$ の計算量である。

5　効率の比較

本節では、4 節で与えた提案手法を適用した場合の等分多項式計算の計算効率を 3.3, 3.4 節で説明した Gaudry と Schost のオリジナルと比較する。

5.1　漸近計算量比較

表 1 に Gaudry-Schost の方法による式 (14) の r_j の計算の主要部の計算コスト、表 2 に提案手法による計算コストを示す。表 1, 2 において「回数」「次数」「計算量」はそれぞれ当該計算の必要回数、計算対象の多項式の次数、計算アルゴリズムの漸近計算量を示す。2 つの手法において最終的な再帰的 GCD 計算は同一であるのでその他の部分の計算コストを比較する。また、計算量 $O(M(d))$ を必要とする計算は $O(d)$ の計算と比較してコストが大きいので、計算量 $O(M(d))$ を必要とする計算のみを比較する。双方において漸近計算量 $O(M(d))$ の計算は多項式乗算及び Aho-Steiglitz-Ullman アルゴリズムである。Aho-Steiglitz-Ullman アルゴリズムは $O(M(d))$ の多項式乗算 1 回と数回の $O(d)$ の計算によって構成されている。したがって、計算量 $O(M(d))$ のアルゴリズムの計算コストを同一と見積もっても実際の計算コストと大きく異なることはないと思われる。また、d が十分に大きいと仮定し FFT 乗算の計算量を適用し、$M(d) = O(d \log d)$ とする。このとき、次数が 1/2 の多項式に対して計算コストも 1/2 程度であると期待できる。d 次多項式に $O(M(d))$ の計算を 1 回適用したときのコストを T_M と書くと、Gaudry-Schost の方法は再帰的 GCD 計算以外に $8T_M$ 程度、提案手法は $5T_M$ 程度を必要とする。したがって、等分多項式 R_{GS} の計算において、再帰的 GCD 計算以外の部分は提案手法により 40%程度の効率向上が期待される。

表 1: Gaudry-Schost の等分多項式計算における r_j の計算コスト

	回数	次数	計算量
K_i	3	約 d	$O(M(d))$
$K_{i,even}(u_j^2 - U_o)$, $K_{i,odd}(u_j^2 - U_o)$	6	約 $d/2$	$O(M(d))$
$\mathfrak{A}_{d_i}(U_0, u_j)$, $\mathfrak{B}_{d_i}(U_0, u_j)$	6	約 $d/2$	$O(d)$
$\mathfrak{E}_1(U_0, u_j)$, $\mathfrak{E}_1(U_0, u_j)$	4	約 $d/2$	$O(M(d))$
r_j	1	約 d	$O(M(d) \log d)$

表 2: 提案手法を用いた場合の r_j の計算コスト

	次数	計算量	回数
$d_i^{(\ell)}(U_3/2)$	3	約 d	$O(d)$
$d_i^{(\ell)}(U_3/2 - u_j/2)$	3	約 d	$O(M(d))$
$d_2^{(\ell)}(-U_3/2 - u_j/2)$	1	約 d	$O(d)$
$g_t(U_3)$	2	約 d	$O(M(d))$
$G_t(U_3, u_j)$	3	約 d	$O(d)$
$H_t(U_2, u_j)$	3	約 $d/2$	$O(d)$
r_j	1	約 d	$O(M(d) \log d)$

5.2 実装比較

ここでは、Gaudry-Schost の方法と提案手法の実装比較を行う。これらの手法の効率の違いは定義体や曲線を変えても大きく変わらないと考えられるのでここでは固定した曲線に対する比較を行う。96bit 素数

$$p = 79228162514264337593543950319$$

位数の有限素体 $\mathbb{F}_p$ 上の種数 2 の超楕円曲線

$$\begin{aligned} C : Y^2 \quad = \quad & X^5 + X^3 + \\ & 25846834439077714983636874797X^2 + \\ & 65207965374085192377003255630X + \\ & 50545209844219400120745695149 \end{aligned} \quad (20)$$

に対し Gaudry-Shost の方法と提案手法を適用し効率を比較した、Gaudry-Schost の方法には Gaudry が作成した NTLJac2 [7] を利用し、提案手法も NTLJac2 を修正して作成した。

表 3 に Gaudry-Schost の方法よる等分多項式 R_{GS} の計算時間（「Gaudry-Schost」と表記）と提案手法を適用した場合の R_{GS} の計算時間（「提案手法」と表記）を示す。表 3 に示した時間は与えられた $d^{(\ell)_i}$, $e^{(\ell)_i}$ に対して R_{GS} を求めるのに必要とした時間である。計算には Intel Xeon E5-2637 v2 3.5GHz を利用した。表 3 から、今回測定した範囲では提案手法を適用することで 2 割弱の速度向上が期待できることが分かる。ℓ が大きくなるにつれて提案手法の速度向上率が低くなるが、計算量の主項であり「Gaudry-Schost」、「提案手法」で同一の計算を行う、再帰的 GCD による r_j の計算の比率が ℓ が大きくなるにつれて高くなることが理由であると考えられる。

表 3: Gaudry-Schost の方法と提案手法による等分多項式 R_{GS} の計算時間（秒）

ℓ	Gaudry-Schost	提案手法
3	0	0
5	2	1
7	19	13
11	362	243
13	1156	755
17	6117	4661
19	12277	9587
23	45229	35013
29	182952	152465
31	263220	217181
37	848864	690851
41	1648268	1383522
43	2140679	1797842

6 位数の計算

現実的なサイズである 96bit と 128bit の有限素体上の種数 2 の超楕円曲線の Jacobian 群の位数計算を行った。Cantor の等分多項式の計算と、R_{GS} を用いた χ_ℓ の係数 s_1, $s_2 \bmod \ell$ の計算には NTLJac2 を利用した。また、R_{GS} の計算には 5.2 節と同様に NTLJac2 を修正して作成した提案手法を利用した。最終的な位数を得るために利用する baby-step giant-step 法には標準的な方法または文献 [19] に示された方法を利用した。また、$\ell = 2^k$ に対する計算には、Gaudry-Schost [9] の方法の改良 [30] の Magma 2.19 [18] 上の実装を利用した。$\ell = 2^k$ 以外の部分の実装には NTL 6.2.1 [25] を利用した。Baby-step giant-step 法の計算には Intel Xeon E5-2643 3.3GHz を利用し、それ以外の計算には Intel Xeon E5-2637 v2 3.5GHz を利用した。

6.1 96bit 有限素体上の曲線

式 (20) で与えた曲線の Jacobian 群の位数 $\#\mathcal{J}_C(\mathbb{F}_p)$ を計算した。

$\ell = 2^{12}, 3, \ldots, 53$ に対する s_1, $s_2 \bmod \ell$ の値を表 4 に示す。表 4 より、中国式剰余定理を用いて

$$\begin{aligned} m &= 2^{12} \cdot 3 \cdot 5 \cdots 47 \cdot 53 \\ &= 66742596561285211607040 \\ s_1 &\equiv 66742596495976970392997 \bmod m \\ s_2 &\equiv 53341510483169790042839 \bmod m \end{aligned}$$

表 4: 各 ℓ に対する s_1, s_2 mod ℓ

ℓ	2^{12}	3	5	7	11	13	17	19	23
s_1	1445	2	2	5	8	4	2	6	8
s_2	727	2	4	6	7	2	12	5	1
ℓ	29	31	37	41	43	47	53		
s_1	11	15	17	7	29	33	16		
s_2	11	9	8	22	1	28	13		

が得られる。ここで、m は式 (6) で与えられた s_i の取りうる範囲（$\approx 8\sqrt{p}$）を越えているので、s_1 が決まり、

$$\begin{aligned} s1 &= 6674259649597697039299 7 - m \\ &= -65308241214043 \end{aligned}$$

である。そこで、最終的な位数計算に通常の baby-step gaiant-step 法を利用して、193bit 位数

$$\begin{aligned} \#\mathcal{J}_C(\mathbb{F}_p) &= 627710173538668593808773784759443676 5 \\ &\quad 857975033604063641161 \\ &= 101 \cdot 2239 \cdot 3257 \cdot 74573 \cdot 11742677 \cdot \\ &\quad 13629741044729363 \cdot \\ &\quad 714051617237248748609 \end{aligned}$$

が得られた [2]。表 5 に計算に要した時間を示す。表 5 において「Cantor」は $d_i^{(\ell)}$, $e_i^{(\ell)}$ の計算時間、「R_{GS}」は $d_i^{(\ell)}$, $e_i^{(\ell)}$ から R_{GS} の計算時間、「χ_ℓ」は R_{GS} 等から χ_ℓ を得るのに必要とした時間を示す。また、「BSGS」は baby-step giant-step 法による最終的な $\#\mathcal{J}_C(\mathbb{F}_p)$ の計算時間を示す。

表 5: 式 (20) で与えた C に対する位数計算時間（秒）

ℓ	合計	Cantor	R_{GS}	χ_ℓ
2^{12}	364157			
3	0	0	0	0
5	4	0	1	2.29
7	24	0	13	10.9
11	307	0	243	64.8
13	920	0	755	165
17	5144	0	4661	483
19	10585	0	9587	997
23	38653	0	35013	3640
29	163360	1	152465	10894
31	228383	1	217181	11201
37	724951	1	690851	34099
41	1439832	1	1383522	56308
43	1878604	2	1797842	80760
47	3605022	2	3441563	163457
53	7510045	3	7276150	233892
BSGS	0			
合計	15969991			

表 5 から χ_ℓ を $\ell = 53$ まで計算すると $\#\mathcal{J}_C(\mathbb{F}_p)$ の計算に 185 日程度を必要とすることとなる。しかし、χ_ℓ の計算を $\ell = 29$ までに留めれば、必要な計算時間は 7 日弱である。この場合にはり、得られた結果から最終的な位数を得るために文献 [19] に示された baby-step giant-step 法を用いる。この計算を行ったところ 2 時間程度で計算を終了した。全体でも 7 日程度で位数計算が可能である。

[2] $\#\mathcal{J}_C(\mathbb{F}_p)$ の最大素因子は 142bit であり、この $\mathcal{J}_C(\mathbb{F}_p)$ では安全な暗号系を実現できないことに注意されたい。

6.2 128bit 有限素体上の曲線

128bit 素数

$$p = 196596493255301158097403350447824412221$$

位数の有限体 $\mathbb{F}_p$ 上の種数 2 の超楕円曲線

$$\begin{aligned} C : Y^2 &= X^5 + X^3 + \\ &\quad 182627089351503598583196047137017513007X^2 + \\ &\quad 58954541787513604095962299916270760246X + \\ &\quad 83860818921064957145121360450373424 23 \end{aligned} \tag{21}$$

の Jacobian 群の位数 $\#\mathcal{J}_C(\mathbb{F}_p)$ を計算した。

$\ell = 2^{13}, 3, \ldots, 47$ に対する s_1, s_2 mod ℓ の値を表 6 に示す。表 6 より、中国式剰余定理を用いて

表 6: 各 ℓ に対する s_1, s_2 mod ℓ

ℓ	2^{13}	3	5	7	11	13	17	19	23
s_1	6201	1	4	3	9	11	14	8	16
s_2	777	0	2	1	4	6	12	6	22
ℓ	29	31	37	41	43	47			
s_1	21	27	28	16	18	6			
s_2	7	15	29	19	8	21			

$$\begin{aligned} m &= 2^{13} \cdot 3 \cdot 5 \cdots 43 \cdot 47 \\ &= 2518588549482460815360 \\ s1 &\equiv 293681430551248696 9 \bmod m \\ s2 &\equiv 1075336056833495286537 \bmod m \end{aligned}$$

が得られる。ここで、m は式 (6) で与えられた s_i の取りうる範囲を越えているので、s_1 が決まり、

$$s1 = 2936814305512486969$$

である。そこで、最終的な位数計算に通常の baby-step gaiant-step 法を利用して、255bit 位数

$$\begin{aligned} \#\mathcal{J}_C(\mathbb{F}_p) &= 386501811602816737461774161410541005 80 \\ &\quad 210713533173219258216000295321425113541 \\ &= 3 \cdot 37 \cdot 97 \cdot 5179 \cdot 59707 \cdot 2381571483596663 \\ &\quad \cdot 48744097951443328718186023386319875 41 \\ &\quad 860341816757 \end{aligned}$$

が得られた [3]。表 7 に計算に要した時間を示す。表 7 の表記は表 5 に倣う。

[3] $\#\mathcal{J}_C(\mathbb{F}_p)$ の最大素因子は 162bit であり、この $\mathcal{J}_C(\mathbb{F}_p)$ では 256bit セキュリティを実現できないことに注意されたい。

表 7: 式 (21) で与えた C に対する位数計算時間（秒）

ℓ	合計	Cantor	R_{GS}	χ_ℓ
2^{13}	622661			
3	0	0	0	0
5	4	0	2	2
7	27	0	14	13
11	342	0	274	68
13	1005	0	839	166
17	6146	0	5310	836
19	12119	0	10880	1238
23	44356	0	40661	3695
29	182845	1	172119	10725
31	261093	1	247380	13712
37	826594	1	795523	31069
41	1715228	2	1626906	88321
43	2210963	2	2097199	113762
47	4101734	2	3987595	114137
BSGS	6942			
合計	9992060			

表 7 から χ_ℓ を $\ell = 47$ まで計算すると $\#\mathcal{J}_C(\mathbb{F}_p)$ の計算に 116 日程度を必要とすることとなる。しかし、96bit 有限体上の場合と同様に χ_ℓ の計算を少なくし、baby-step giant-step 法による最終位数計算の負担を増やすことで、全体の計算量を削減可能である。例えば、χ_ℓ の計算を $\ell = 43$ までに留めれば、必要な計算時間は 68 日強である。この場合、文献 [19] に示された baby-step giant-step 法を用いて最終的な位数を得るため 20 時間弱を必要とした。したがって、全体で 69 日程度で位数計算が可能である。χ_ℓ の計算をより一層少なくすることも考えられるが、$\ell = 41$ までに留めた場合であっても、最終位数を求めるためのた baby-step giant-step 法に 512GB 以上の RAM を必要とするため、現実的には難しい。

謝辞　位数計算の一部に小崎俊二氏が作成したプログラムを利用しました。同氏に感謝致します。本研究は JSPS 科研費 25400055 の助成を受けたものです。

参考文献

[1] L. M. Adleman, J. DeMarrais, and M. D. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobian of large genus hyperelliptic curves over finite fields. *ANTS-I*, LNCS877, pp. 28–40. Springer, 1994.

[2] L. M. Adleman and M. D. Huang. Counting rational points on curves and Abelian varieties over finite fields. *ANTS-II*, LNCS1122, pp. 1–16. Springer, 1996.

[3] A. Aho, K. Steiglitz, and J. D. Ullman. Evaluating polynomials at fxed sets of points. *SIAM J. Comput.*, 4, No. 4, pp. 533–539, 1975.

[4] D. G. Cantor. On the analogue of the division polynomials for hyperelliptic curves. *J. für die reine und angewandte Mathematik*, 447, pp. 91–145, 1994.

[5] N. D. Elkies. Elliptic and modular curves over finite fields and related computational issues. *Computational perspectives on number theory*, pp. 21–76. AMS, 1995.

[6] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge U. P., 3rd edition, 2013.

[7] P. Gaudry. NTLJac2. http://www.loria.fr/ gaudry/NTLJac2/, 2004.

[8] P. Gaudry and R. Harley. Counting points on hyperelliptic curves over finite fields. *ANTS-IV*, LNCS1838, pp. 313–332. Springer, 2000.

[9] P. Gaudry and É. Schost. Construction of secure random curves of genus 2 over prime fields. *EUROCRYPT 2004*, LNCS3027, pp. 239–256. Springer, 2004.

[10] P. Gaudry and É. Schost. A low-memory parallel version of Matsuo, Chao and Tsujii's algorithm. *ANTS-VI*, LNCS3076, pp. 208–222. Springer, 2004.

[11] P. Gaudry and É. Schost. Genus 2 point counting over prime fields. *J. Symbolic Comput.*, 47, pp. 368–400, 2012.

[12] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for Computer Algebra*. Kluwer Academic Pub., 1992.

[13] F. A. Izadi and V. K. Murty. Counting points on an Abelian variety over a finite field. *INDOCRYPT 2003*, LNCS2904, pp. 323–333. Springer, 2003.

[14] W. Kampkötter. *Explizite Gleichungen für Jacobische Varietäten hyperelliptischer Kurven*. PhD thesis, GH Essen, 1991.

[15] N. Koblitz. Hyperelliptic curve cryptosystems. *J. Cryptology*, 1, No. 3, pp. 139–150, 1989.

[16] N. Koblitz. *Algebraic Aspects of Cryptography*, Vol. 3 of *Algorithms and Computation in Mathematics*. Springer, 1998.

[17] S. Lang. *Elliptic Curves Diophantine Analysis*. Springer, 1978.

[18] The Magma computational algebra system. `http://magma.maths.usyd.edu.au/magma/`.

[19] K. Matsuo, J. Chao, and S. Tsujii. An improved baby step giant step algorithm for point counting of hyperelliptic curves over finite fields. *ANTS-V*, LNCS2369, pp. 461–474. Springer, 2002.

[20] K. Matsuo, J. Chao, and S. Tsujii. Baby step giant step algorithms in point counting of hyperelliptic curves. *IEICE Trans.*, E86-A, No. 5, pp. 1127–1134, May 2003.

[21] J. Pila. Frobenius maps of Abelian varieties and finding roots of unity in finite fields. *Math. Comp.*, 55, pp. 745–763, 1990.

[22] H.-G. Rück. Abelian surfaces and Jacobian varieties over finite fields. *Compositio Math.*, 76, pp. 351–366, 1990.

[23] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod *p*. *Math. Comp.*, 44, pp. 483–494, 1985.

[24] R. Schoof. Counting points on elliptic curves over finite fields. *J. Théorie des Nombers de Bordeaux*, Vol. 7, pp. 219–254, 1995.

[25] V. Shoup. NTL: A library for doing number theory. `http://www.shoup.net/ntl/`, 1990.

[26] A. V. Sutherland. Gallery of Jacobians. `http://www-math.mit.edu/~drew/ZetaFunctions.html`, 2007.

[27] A. V. Sutherland. *Order computations in generic groups*. PhD thesis, Massachusetts Institute of Technology, 2007.

[28] A. V. Sutherland. A generic approach to searching for Jacobians. *Math. Comp.*, 78, pp. 485–507, 2009.

[29] 磯田遼, 松尾和人. Sutherland の位数計算法について. SCIS2015, 1F2-3, 2015.

[30] 小崎俊二, 松尾和人. 種数 2 の超楕円曲線の 2 冪ねじれ点計算の改良. 日本応用数理学会論文誌, 17, No. 4, pp. 577–593, 12 月 2007.

[31] 松尾和人. 超楕円曲線暗号と位数計算. 情報セキュリティ総合科学, 第 2 巻, pp. 43–61. 情報セキュリティ大学院大, 2010. `http://www.iisec.ac.jp/proc/vol0002/iisec_proc_002_p043.pdf`.