SCIS 2015 The 32nd Symposium on Cryptography and Information Security
Kokura, Japan, Jan. 20 - 23, 2015
The Institute of Electronics, Information and Communication Engineers

Sutherlandの位数計算法について
On the Sutherland Algorithm for Group Order Computing

磯田 遼 * 松尾 和人 *
Ryo Isoda Kazuto Matsuo

あらまし Sutherlandは一般の有限アーベル群の位数計算アルゴリズムを提案し、これを用いて超楕円曲線のJacobianの位数を計算した。Sutherlandのアルゴリズムは、位数がスムーズ性に関する一定の条件を満足したときのみ計算に成功するものである。また、この条件は計算量を決定するパラメータによって変化する。アルゴリズムを安全な超楕円曲線の生成に利用する場合には、ランダムに選択した曲線に対して位数計算を繰り返す必要があるが、パラメータの設定によってアルゴリズムの計算量と計算の成功確率が変化するため、効率的な曲線生成には適切なパラメータ設定が必要となる。Sutherlandはパラメータ設定について言及しているものの詳細な記述はない。そこで、本論文では安全な超楕円曲線を生成する場合の適切なパラメータ設定について議論した。また、80bit有限素体上の種数2の超楕円曲線に対する実装実験により議論の妥当性を検証するとともにアルゴリズムの有効性を確認した。

キーワード 位数計算, 有限アーベル群, 超楕円曲線, baby-step giant-step アルゴリズム, 超楕円曲線暗号

1 はじめに

安全な超楕円曲線暗号を構成するために必要なJacobianの位数計算は長い間研究課題となっている。小標数の有限体上の曲線に対する位数計算に対しては実用的なアルゴリズム[Ked01, Mes02, Ver03, LL06]が提案されている。素体をはじめとする大標数の有限体上の曲線に対しても安全な曲線の構成は可能ではあるものの、曲線を豊富に提供する迄には至っていない。大標数の有限体上の超楕円曲線に適用可能な位数計算アルゴリズムは楕円曲線に対するSchoofアルゴリズム[Sch85, Sch95]の拡張としてPila [Pil90]によって提案され、その後も様々な改良が行われてきた[Kam91, ADH94, AH96, GH00, MCT02, GS04, GS12]。また、実装評価も行われ、現在のところGaudryとSchostが（特殊な曲線ではあるものの）254bitの位数（最大素因数は250bit）の安全な種数2の超楕円曲線の生成に成功している[GS12]。これらのSchoof法とは別に、Sutherlandは一般の有限アーベル群に対して適用可能な位数計算アルゴリズム[Sut07b, Sut09]（以下では、Sutherlandアルゴリズムと呼ぶ。）を提案し、これを用いてJacobianが188bit位数（最大素因数は182bit）の安全な種数2の超楕円曲線の構成に成功している[Sut07a]。

Sutherland アルゴリズムは Shanks [Sha71] の baby-step giant-step アルゴリズムに基づいたアルゴリズムであり、Schoof法とは違い一般の有限アーベル群に適用可能である。しかし、baby-step giant-step アルゴリズムやSchoof法と異なり、計算量を決定するパラメータ B によって決まる“B-easy”と呼ばれるスムーズ条件を満足する位数のみを計算可能なアルゴリズムである。Sutherlandアルゴリズムの計算量は B に比例するが B が小さいと位数が B-easyである確率は低くなる。したがって、アルゴリズムの計算量とアルゴリズムの計算成功確率にトレードオフがあり、効率的に安全な超楕円曲線を得るためには適切な B を選択する必要がある。Sutherland [Sut09]はパラメータ B の設定について言及しているものの詳細な記述はない。そこで、本論文では安全な超楕円曲線を生成する場合の適切な B の設定について議論し、160bit及び256bit位数に対して最適な B を具体的に求めた。また、Jacobianが160bit位数を持つ種数2の超楕円曲線に対する実装実験によりその妥当性を検証するとともにアルゴリズムの有効性を確認した。

本論文では、まず第2節で B-easy数の定義示すとともにSutherlandアルゴリズムとその計算量を概説する。次に、第3節でSutherlandアルゴリズムを用いて位数計算に成功するまでに必要な計算量について議論し、BachとPeraltaの結果[BP96]を利用した場合の計算量を160bitと256bit位数について具体的に求める。そして、第4節で B-easy 数の分布を実験から求め、その結果を用いて

* 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.

Sutherland アルゴリズムが位数計算に成功するまでに必要な計算量を 160bit と 256bit 位数について再評価する。また、第 5 節では Sutherland アルゴリズムと一般の baby-step giant-step 法の効率を比較する。さらに、第 6 節では種数 2 の超楕円曲線に対する Sutherland アルゴリズムを実装しアルゴリズムの効果を確認する。

2 Sutherland の位数計算アルゴリズム

G を有限アーベル群とする。G の任意の元の位数の最小公倍数を $\lambda(G)$ と書き G の exponent と呼ぶ。すなわち、$\lambda(G) = \mathrm{lcm}\{|\alpha| \mid \alpha \in G\}$ である。Sutherland アルゴリズムはランダムに選択した $\alpha \in G$ に対して $|\alpha|$ を求めることを繰り返して $\lambda(G)$ を求め、$\lambda(G)$ から $|G|$ を求めるアルゴリズムである。$|\alpha|$ の計算には素因数分解における $p-1$ 法に似たアイディアを利用し、スムーズな位数の計算を効率化している。[Sut09, Definition 1] にしたがい、定義 1 に Sutherland アルゴリズムの計算量評価に必要となる “B-easy” と呼ばれるスムーズ数の定義を示す。

定義 1 (B-easy)**.**
与えられた $B \in \mathbb{N}$ に対して $\ell_1, \ell_2, \ldots, \ell_w$ を B 以下の素数の全てとする。即ち、

$$\ell_i < \ell_{i+1} \ (i = 1, 2, \ldots),\ \ell_w \leq B,\ \ell_{w+1} > B \tag{1}$$

とする。また、ℓ_i の B を越えない最大冪を q_i とする。即ち、

$$q_i = \ell_i^{d_i} \ s.t.\ d_i \geq 1,\ q_i \leq B,\ q_i \ell_i > B \tag{2}$$

とする。さらに

$$E = \prod_{1 \leq i \leq w} q_i \tag{3}$$

とする。$N/\gcd(N, E) \leq B^2$ を満足する $N \in \mathbb{N}$ を *“B-easy”* であるという。また、*B-easy* ではない N を *“B-hard”* であるという。

Sutherland アルゴリズムは $^{\forall}B \in \mathbb{N}$ に対して以下を満足する [Sut09, Proposition 2]。

1. アルゴリズムがリジェクトしたならば $|G|$ は B-hard である。
2. そうでないならば、高確率で $\lambda(G)$（と $|G|$、群構造）を出力する。
3. 計算量は $O(B) + O(\log^2 |G|)$ である。

したがって、Sutherland アルゴリズムを用いて位数が得られるまでの効率は B の設定によって変化する。特に、B が大きいほど位数を計算できる可能性が大きくなるが、計算量も大きくなり、B の設定には計算成功確率と計算量のトレードオフが存在する。

Algorithm 1 Group Order

Input: G, B
Output: $\lambda(G)$, $|G|$, G の群構造またはリジェクト

1: 式 (3) で与えられた E を計算
2: $\lambda(G)$ が B-easy であると仮定し $\lambda(G)$ を計算（リジェクト可）
3: $\lambda(G)$ の各素因子 p に対応する p-Sylow 群の群構造を計算し、それらから G の群構造と $|G|$ を得る。

大きな位数の Jacobian に対しては Step 3 は必要ないことが多い [Sut09, Sect. 4]。そこで以下では Step 2 を扱う。Algorithm 2 に Step 2 の計算を行う Group Exponent アルゴリズムを示す。

Algorithm 2 Group Exponent

Input: G, B, E, $c \in \mathbb{N}$
Output: $\lambda(G)$ または $\lambda(G)$ が B-hard のときリジェクト

1: $N \leftarrow 1$
2: $\alpha \in G$ をランダムに選択
3: $\gamma \leftarrow [N]\alpha$
4: $\beta \leftarrow [E]\gamma$
5: $N' \leftarrow |\beta|$ ただし $N' > B^2$ の場合は計算を中断し、$\lambda(G)$ が B-hard であるとする
6: $\delta \leftarrow [N']\gamma$
7: $N'' \leftarrow |\delta|$（$N'' \mid E$ を利用する）
8: $N \leftarrow NN'N''$
9: **for** $t = 1, \ldots, c$ **do**
10: 　$\alpha \in G$ をランダムに選択
11: 　$\gamma \leftarrow [N]\alpha$
12: 　$N' \leftarrow |\gamma|$（$N' \mid E$ を利用する／中断可）
13: 　**if** Step 12 の計算が中断された **then**
14: 　　**goto** Step 2
15: 　$N \leftarrow NN'$
16: **return** N ($= \lambda(G)$ w. high prob.)

Algorithm 2 の入力 c は Step 9 以下の繰り返し回数を制御する小さな定数であり、c が大きいほど出力が $\lambda(G)$ である確率が高くなる。

以下では [Sut09] にしたがって Algorithm 2 の計算量評価を示す。式 (1) より

$$w = O(B/\log B) \tag{4}$$

であり、(2) より

$$q_i = O(B) \tag{5}$$

であるので、

$$E = O(B^{B/\log B})$$

を得る。したがって、Algorithm 2 の Step 4 の計算量は $O(\log E) = O(B)$ となる。また、Step 3 は探索範囲を $|\beta| \le B^2$ に設定した baby-step giant-step アルゴリズムを用いるので、$|\beta| \le B^2$ の場合には計算量 $O(B)$ で $|\beta|$ が求まる。一方、$|\beta| > B^2$ のときには値が求まらないのでアルゴリズムを中断し、$\lambda(G)$ が B-hard であるとする。Algorithm 2 の Step 7 と 12 は Algorithm 3 に示す Linear Order アルゴリズムによって計算される。

Algorithm 3 Linear Order

Input: $E = q_1 \dots q_w$, $\alpha \in G$

Output: $|\alpha| \mid E$ のとき $|\alpha|$

1: $j \leftarrow 0$
2: $\alpha_0 \leftarrow \alpha$
3: **repeat**
4: $\quad j \leftarrow j + 1$
5: $\quad$ **if** $j > w$ **then**
6: $\qquad |\alpha| \nmid E$ として計算を中断
7: $\quad \alpha_j = [q_j]\alpha_{j-1}$
8: **until** $\alpha_j = 0$
9: $N \leftarrow 1$
10: **repeat**
11: $\quad i \leftarrow j - 1$
12: $\quad N \leftarrow |[N]\alpha_i|N$
13: $\quad [N]\alpha_j = 0$ を満足する最小の $j \in [0, i]$ を計算
14: **until** $j = 0$
15: **return** N

式 (4), (5) より Algorithm 3 の Step 3 から始まる repeat 文全体の計算量は $O((B/\log B)\log B) = O(B)$ である。以下では Algorithm 3 の Step 10 以降の repeat 文の計算量を評価する。Step 12 の $|[N]\alpha_i|$ の計算は $[N]\alpha_{i+1} = [N][q_{i+1}]\alpha_i = 0$ を利用し線形探索によって

$$\begin{aligned} O(d_{i+1}\log l_{i+1}) &= O((\log B/\log l_{i+1})\log l_{i+1}) \\ &= O(\log B) \end{aligned}$$

で計算可能である。また、Step 13 の $j \in [0, i]$ の計算には 2 分探索によって計算量 $O(\log^2 |G|)$ で計算可能である。repeat 文の繰り返し回数は $O(\log |G|)$ であり、$B < |G|$ であることを考慮すると、Step 10 以降の repeat 文全体の計算量は $O(\log^3 |G|)$ となる。したがって、Algorithm 3 の計算量は $O(B + \log^3 |G|)$ である。

以上より Algorithm 2 の計算量は $O(B + \log^3 |G|)$ であるが、以下では $\log^3 |G| = O(B)$ と仮定し、Algorithm 2 の計算量を $O(B)$ として議論を進める。

3 位数計算に成功するまでに必要な計算量

前節で述べたように Sutherland アルゴリズムの計算量は $O(B)$ である。しかし、Sutherland アルゴリズムは通常のアルゴリズムと違い B-hard な位数を計算できない。例えば安全な超楕円曲線を構成する際には、その Jacobian の位数が十分なサイズの素因数を持つまで曲線を替えて繰り返し位数計算を行うので、Sutherland アルゴリズムのような位数が計算できない場合があるアルゴリズムであってもそれほど問題とはならない。しかし、位数計算の成功確率を知らなければ安全な曲線が得られるまでの計算量の見積りができない。Sutherland [Sut09] は Bach と Peralta の結果 [BP96] を利用してアルゴリズムの成功確率を論じている。そこで本節では、まず、Algorithm 2 の評価に必要となる [BP96] の結果の一部を紹介する。次に、[BP96] の結果を利用し、Sutherland アルゴリズムを繰り返し適用することよって 160bit 位数と 256bit 位数を得るまでに必要な計算量を評価する。

3.1 Bach と Peralta の結果 [BP96]

n を w 個の異なる素数 $\ell_1, \ell_2, \dots, \ell_w$ の積

$$n = \ell_1 \ell_2 \cdots \ell_{w-1} \ell_w$$

とする。ただし、$\ell_1 < \ell 2 < \cdots < \ell_{w-1} < \ell_3$ とする。x に対して最大素因数が y 以下で 2 番目に大きな素因数が z 以下の整数 $n \le x$ の個数を

$$\Psi(x, y, z) = \#\{n \le x \mid \ell_w \le y, \ell_{w-1} \le z\}$$

と置く。Bach と Peralta は、最大素因数が x^β 以下で、2 番目に大きな素因数が x^α 以下である整数 x の存在確率の極限値

$$G(\alpha, \beta) = \lim_{x \to \infty} \Psi(x, x^\beta, x^\alpha)/x, \ 0 < \alpha < \beta < 1$$

が存在することを示し、これを積分方程式として与えた。さらに

$$\sigma(u, v) = G(1/u, 1/v)$$

の具体的な数値を $2 \le u \le 20$, $2 \le v \le 10$ に対して求めた [BP96, Table 1]。また、$10 \le u \le 15$, $6 \le v \le 9$ 対するより詳細な結果を [BP96, Table 2] に示し、このテーブルの範囲に於いて、σ の指数関数近似を指数部の $u\log u$, $v\log v$ に関する線形近似によって

$$\begin{aligned} \sigma(u, v) &\approx e^{f(u,v)} \\ f(u, v) &= -0.933064u\log u - 0.280283v\log v \\ &\quad +4.55219 \end{aligned} \tag{6}$$

と求めた。近似には最小 2 乗法が利用された。この近似式は [BP96, Table 2] の範囲では誤差が 30%以内に収ま

るが、範囲外では誤差が急速に大きくなる [BP96]。

3.2 Bach と Peralta の結果を利用した計算量評価

Sutherland [Sut09] は Bach と Peralta の結果を利用することで、位数計算に成功するまでに必要な計算量を評価可能であることを示している。また、それによって、最適な B を決定可能であり、成功確率を考慮に入れた最適な計算量が評価可能であると述べている。以下では [Sut09] にしたがって位数計算に成功するまでに必要な計算量を評価する。

$|G|$ の最大値を N とする。$B = N^{1/u}$ と置き、N が B-easy である確率を $\sigma(u)$ と書くと

$$\sigma(u) = \sigma(u, u/2) = G(1/u, 2/u)$$

である。したがって、与えられた u に対して位数計算に成功するまでに必要な計算量が

$$O(B/\sigma(u)) = O(N^{1/u}/\sigma(u)) \tag{7}$$

と得られる。式 (7) を最小化する u を求めることで、N に対する B の最適値及び最小計算量を決定可能である。

3.3 Bach と Peralta の結果を利用した最小計算量評価

第 3.2 節にしたがって、160bit と 256bit の N に対して Sutherland アルゴリズムの最適計算量を決定する。

まず、式 (6) より $\sigma(u)$ の近似が

$$\sigma(u) \approx e^{f(u)} \tag{8}$$

$$f(u) = -1.0732055u\log u + 4.55219 \tag{9}$$

と得られる。しかし、この近似値は $12 \le u \le 15$ の外では精度が悪く、また実際の u の値はこの範囲より小さいところにあると考えられる。そこで [BP96, Tables 1, 2] から u のより広い範囲で精度の良い近似を求めた。具体的には f のより精度の良い近似関数を求めた。

まず、[BP96, Tables 1, 2] を変換して得た $4 \le u \le 20$ に対する $\sigma(u)$ の値を表 1 に示す。表 1 において "-" は [BP96, Tables 1, 2] に値が示されていないことを表す。

本研究では、表 1 の数値を利用し、指数 f が $u\log u$ の 3 次式である指数関数で $\sigma(u)$ を近似した。近似には最小 2 乗法を用いた。結果を式 (10) に示す。

$$\begin{aligned} f(u) = {} & 4.458\text{e-}4(u\log u)^3 - 6.213\text{e-}3(u\log u)^2 \\ & -8.122\text{e-}1u\log u + 2.447 \end{aligned} \tag{10}$$

式 (8) で与えられる $\sigma(u)$ にこの f を適用した場合、[BP96, Tables 1, 2] との誤差は 14%以内に収まることを確認した。

以下では、式 (7) より位数計算に成功するまでに必要な Algorithm 2 の合計計算量を

$$T = B/\sigma(u) = N^{1/u}/\sigma(u)$$

と置き、式 (10) を利用して T の最小値を求める。

まず、$N = e^{\gamma}$ と書くと、T を f を用いて

$$T \approx e^{\gamma/u}/e^{f(u)} = e^{\gamma/u - f(u)} \tag{11}$$

と書くことができる。したがって、$\gamma/u - f(u)$ を最小化する $u > 0$ を求めることで、最小計算量およびその計算量を実現するパラメータ B を得ることができる。

例えば、$N = 2^{160}$ に対しては $\gamma = 160\log 2$ であり、$u = 6.419$ のとき $\gamma/u - f(u)$ が最小化される。そして、そのときの計算量は

$$T \approx e^{25.33} \approx 2^{36.54}$$

となる。また、このとき

$$B = 2^{160/u} = 2^{24.93}$$

である。したがって、B を 25bit 程度に設定すれば、繰り返しも含めて 37bit 程度の計算量で 160bit の位数を持つ群を得ることが可能である。

同様に $N = 2^{256}$ のとき $\gamma = 256\log 2$ であり、$u = 7.73$ で $\gamma/u - f(u)$ が最小化される。この u に対し

$$T \approx e^{34.72} \approx 2^{50.10}$$

$$B = 2^{256/u} = 2^{33.12}$$

である。したがって、B を 34bit 程度に設定すれば、51bit 程度の計算量で 256bit の位数を持つ群を得ることが可能である。

4 B-easy に関する実験とその結果を利用した最小計算量評価

前節では、Sutherland [Sut09] にしたがい、Algorithm 2 を利用して所望のサイズの位数を持つ群を得るまでに必要な計算量を [BP96, Tables 1, 2] を利用して求めた。しかし、Algorithm 2 で計算可能な B-easy 数と [BP96] が扱っているスムーズな数は素数冪の部分にギャップがある。そこで本節では B-easy 数の存在確率を実験により確認し、その結果を用いて Algorithm 2 で位数を発見するまでに必要な計算量を再評価する。

4.1 実験

まず、区間 $[1, 2^{160} - 1]$ の整数乱数 10^6 個を生成し、10^6 個中の 2^n-easy 数の個数を $0 \le n \le 160$ に対し積算した。積算結果を表 2 に示す。次に表 2 から $\sigma(u)$ の値を求めた。結果を表 3 に示す。

実験には Sage 6.3 [Sag] を利用した。

表 1: Bach と Peralta の結果 [BP96, Tables 1,2] から得られる $\sigma(u)$ の値

u	4.0	5.0	6.0	7.0	8.0	9.0	10.0
$\sigma(u)$	9.639901e-2	-	1.092267e-3	-	3.662652e-6	-	5.382861e-9
u	11.0	12.0	13.0	14.0	15.0	16.0	17.0
$\sigma(u)$	-	4.254912e-12	9.875063e-14	2.048898e-15	3.836313e-17	6.534072e-19	-
u	18.0	19.0	20.0				
$\sigma(u)$	1.465048e-22	-	2.415504e-26				

表 2: 160bit 乱数 10^6 個に対する 2^n-easy 数の個数

n	19	20	21	22	23	24	25	26	27	28
個数	0	2	14	27	76	186	405	769	1351	2380
n	29	30	31	32	33	34	35	36	37	38
個数	3886	5993	8992	12945	18045	24579	32704	42367	53740	67209
n	39	40	41	42	43	44	45	46	47	48
個数	82345	99405	118439	139082	161312	184973	210121	236233	263631	291635
n	49	50	51	52	53	54	55	56	57	58
個数	320854	350666	381039	411643	443345	474592	505043	534683	562938	590300
n	59	60	61	62	63	64	65	66	67	68
個数	616792	642217	666816	690555	713641	735823	757149	777990	798097	817479
n	69	70	71	72	73	74	75	76	77	78
個数	836057	854188	871647	888351	904539	920277	935224	949912	964015	977589
n	79	80								
個数	989927	1000000								

4.2 最小計算量の再評価

Algorithm 2 によって位数が得られるまでに必要な計算量を表 3 を用いて再評価した。

そのために第 3.3 節と同様に $\sigma(u) \approx e^{f(u)}$ と置き、表 3 の数値を用いて f を $u \log u$ の 3 次式で近似した。式 (12) に得られた近似関数 f を示す。

$$\begin{aligned} f(u) &= 1.073\text{e-}3(u\log u)^3 - 5.014\text{e-}2(u\log u)^2 \\ &\quad -2.640\text{e-}1(u\log u) + 5.296\text{e-}1 \end{aligned} \tag{12}$$

この f を式 (11) に適用し $\gamma/u - f(u)$ を最小化する $u > 0$ を求めることで、第 3.3 節と同様に最小計算量およびその計算量を実現するパラメータ B を得た。

$N = 2^{160}$ に対しては $u = 6.277$ のとき $\gamma/u - f(u)$ が最小化され、計算量は

$$T \approx e^{25.20} \approx 2^{36.36}$$

となる。このとき

$$B = 2^{25.49}$$

である。したがって、B を 26bit 程度に設定することで 37bit 程度の計算量で 160bit の位数を持つ群を得ることが可能であるとの結論が得られる。

同様に $N = 2^{256}$ のとき $u = 7.506$ で $\gamma/u - f(u)$ が最小化され、この u に対し

$$\begin{aligned} T &\approx e^{34.87} \approx 2^{50.30} \\ B &= 2^{34.11} \end{aligned}$$

である。したがって、B を 35bit 程度に設定すれば、51bit 程度の計算量で 256bit の位数を持つ群を得ることが可能であると結論付けられる。

以上で得られた結果と第 3.3 節の結果を比較すると最小計算量の評価結果ほぼ変わっていない。したがって、B-easy な数と [BP96] が扱っているスムーズな数の間に存在する素数冪に関するギャップは Sutherland アルゴリズムの計算成功確率の評価にはほとんど影響を与えないことが分かった。

5 位数計算に成功するまでに必要な計算量に関する考察

本節では Sutherland アルゴリズムと通常の baby-step giant-step アルゴリズムの効率を比較する。まず、通常の baby-step giant-step アルゴリズムでは任意の有限アーベル群の位数を計算可能であるが、Sutherland アルゴリズムでは位数が B-easy でないと計算できず、用途が限定される。しかし、低種数の超楕円曲線は Jacobian の位数が得られればその 2 次ツイストの Jacobian の位数も得られるため、B-easy な位数が計算できれば、その Jacobian の位数が十分な大きさの素因数を持つ 2 次ツイストを得ることが可能である。したがって、安全な超楕円曲線の構成に Sutherland アルゴリズムを利用可能である。同様に離散対数問題が困難となる安全な有限アーベル群の構成において、B-easy な位数を持つ群から他の群への変換手法が存在する場合には、Sutherland アルゴリズムを利用できる可能性がある。そこで以下では、離

表 3: 実験から得られた $\sigma(u)$ の値

u	2.0	2.025	2.051	2.078	2.105	2.133	2.162	2.192	2.222	2.254
$\sigma(u)$	1.0	9.899e-1	9.776e-1	9.640e-1	9.499e-1	9.352e-1	9.203e-1	9.045e-1	8.884e-1	8.716e-1
u	2.286	2.319	2.353	2.388	2.424	2.462	2.50	2.540	2.581	2.623
$\sigma(u)$	8.542e-1	8.361e-1	8.175e-1	7.981e-1	7.780e-1	7.571e-1	7.358e-1	7.136e-1	6.906e-1	6.668e-1
u	2.667	2.712	2.759	2.807	2.857	2.909	2.963	3.019	3.077	3.137
$\sigma(u)$	6.422e-1	6.168e-1	5.903e-1	5.629e-1	5.347e-1	5.050e-1	4.746e-1	4.433e-1	4.116e-1	3.810e-1
u	3.200	3.265	3.333	3.404	3.478	3.556	3.636	3.721	3.810	3.902
$\sigma(u)$	3.507e-1	3.209e-1	2.916e-1	2.636e-1	2.362e-1	2.101e-1	1.850e-1	1.613e-1	1.391e-1	1.184e-1
u	4.0	4.103	4.211	4.324	4.444	4.571	4.706	4.848	5.0	5.161
$\sigma(u)$	9.941e-2	8.235e-2	6.721e-2	5.374e-2	4.237e-2	3.270e-2	2.458e-2	1.805e-2	1.295e-2	8.992e-3
u	5.333	5.517	5.714	5.926	6.154	6.4	6.667	6.957	7.273	7.619
$\sigma(u)$	5.993e-3	3.886e-3	2.380e-3	1.351e-3	7.690e-4	4.050e-4	1.860e-4	7.600e-5	2.700e-5	1.400e-5
u	8.0									
$\sigma(u)$	2.000e-6									

散対数問題に基づく暗号の構成を想定して、一般の有限アーベル群と種数 1, 2 の超楕円曲線の Jacobian の位数計算に対して効率を比較する。

一般の有限アーベル群に対して baby-step giant-step アルゴリズムの計算量は位数の最大値 N に対して $O(\sqrt{N})$ であり、$N \approx 2^{160}$ のとき約 80bit、$N \approx 2^{256}$ のとき約 128bit の計算量を必要とする。一方、第 4.2 節の評価では、Sutherland アルゴリズムによって位数を得るまでに必要な計算量は $N \approx 2^{160}$ に対して約 37bit、$N \approx 2^{256}$ に対して約 51bit の計算量であるので、通常の baby-step giant-step アルゴリズムより効率的である。

次に超楕円曲線に対する効率を比較する。

Hasse-Weil の定理から有限体 $\mathbb{F}_q$ 上の種数 g の超楕円曲線の Jacobian $\mathcal{J}(\mathbb{F}_q)$ の位数は

$$(\sqrt{q}-1)^{2g} \leq \#\mathcal{J}(\mathbb{F}_q) \leq (\sqrt{q}+1)^{2g}$$

を満足する。したがって、$\#\mathcal{J}(\mathbb{F}_q)$ の最大値を N とすると $\#\mathcal{J}(\mathbb{F}_q)$ が取り得る値の範囲は $O(N^{\frac{2g-1}{2g}})$ である。通常の baby-step giant-step アルゴリズムはこれを利用可能であり、計算量は位数の範囲の平方根になる。種数 1（楕円曲線）のときは 160bit 位数に対して位数の範囲は約 80bit なので計算量は約 40bit になる。同様に 256bit 位数に対して位数の範囲は約 128bit なので計算量は約 64bit になる。したがって、楕円曲線に関しては 256bit 位数に対しては baby-step giant-step アルゴリズムより Sutherland アルゴリズムの方が効率的であると考えられるが、160bit 位数に関しては今回の結果だけでは効率の優劣を判断できない。また、種数 2 の超楕円曲線に対しては 160bit 位数の位数範囲は約 120bit なので計算量は約 60bit になり、256bit 位数の位数範囲は約 192bit なので計算量は約 96bit になる。したがって、種数 2 の超楕円曲線に関しては 160bit 以上の安全な曲線を構成する際、baby-step giant-step アルゴリズムより Sutherland アルゴリズムを利用した方が効率的であると考えられる。

6 実装評価

Magma 2.19 [Mag] 上で Algorithm 1 を実装し 80bit の有限素体上の種数 2 の超楕円曲線の Jacobian の位数計算の実験を行った。

Algorithm 1 の Step 2 には Algorithm 2 を実装し利用した。また、Step 3 には [Sut07b, Algorithm 9.1] に記載されたアルゴリズムを実装し利用した。

Sutherland は primorial-steps アルゴリズム [Sut07b, Sut09] と呼ばれる baby-step giant-step アルゴリズムの改良アルゴリズムを提案し、Algorithm 2 の Step 5 にこのアルゴリズムを適用しているが、本実装では Step 5 に通常の baby-step giant-step アルゴリズムを利用した。

実験では 80bit 素数

$$p = 793716941781534054254869$$

に対して素体 $\mathbb{F}_p$ 上の種数 2 の超楕円曲線

$$Y^2 = X^5 + a_3X^3 + a_2X^2 + a_1X + a_0, \quad a_i \in \mathbb{F}_p$$

の Jacobian の位数を計算した。パラメータ B には最適設定であると考えられる 26bit 値を利用した。また、比較のために、28bit の B に対しても実験を行った。Algorithm 2 の Step 9-15 の繰り返し回数は $c = 12$ と設定した。

実験は Core i7-3820 3.6GHz 上の Linux 2.6 で行った。

6.1 26bit の B を用いた実験

第 4 節の評価で位数計算に成功するまでに必要な計算量が最小となった 26bit の B として $B = 48000000$ を選択し 3200 本のランダムな種数 2 の超楕円曲線に対して Algorithm 2 を適用した結果、2 本の曲線の位数計算に成功した。$B = 48000000$ に対して $u = \log_{2^{160}} B \approx 6.270$ より、式 (12) を用いて評価すると $\sigma(u) \approx 5.444\text{e-}4$ であり、位数計算に成功する曲線が 1800 本に 1 本程度存在すると見積もられるが、本実験の結果はこの見積りに矛

盾しない。

位数計算に成功した曲線は

$$
\begin{aligned}
C_1 : Y^2 &= X^5 + 420211455200083993764556X^3 \\
&\quad +626506411773987673110452X^2 \\
&\quad +194493675718514293670671X \\
&\quad +443161714765550528118516 \\
C_2 : Y^2 &= X^5 + 260870204168959237767444X^3 \\
&\quad +469808903396604422055006X^2 \\
&\quad +621452280191572156702031X \\
&\quad +160607371667356688871554
\end{aligned}
$$

である。C_1 の Jacobian の位数とその素因数分解は

$$
\begin{aligned}
\#\mathcal{J}(\mathbb{F}_p) &= 629986583671982734735967627463 7538 \\
&\quad 92623246649600 \\
&= 2^8 \cdot 5^2 \cdot 11 \cdot 23 \cdot 29 \cdot 34183 \cdot 159721 \\
&\quad \cdot 1061623 \cdot 1776701 \cdot 5916853 \\
&\quad \cdot 220183700641
\end{aligned}
$$

で与えられる。$\#\mathcal{J}(\mathbb{F}_p)$ は 159bit であった。また、C_1 の 2 次ツイストの Jacobian の位数は

$$
\begin{aligned}
\#\mathcal{J}_t(\mathbb{F}_p) &= 6299865836700795038076359189634959 \\
&\quad 31668075193780 \\
&= 2^2 \cdot 3 \cdot 5 \cdot 17 \cdot 6176339055589014743212 1 \\
&\quad 16852583293447726227 39
\end{aligned}
$$

となる。$\#\mathcal{J}_t(\mathbb{F}_p)$ の最大素因数は 149bit であり $\mathcal{J}_t$ は暗号利用には適さないと考えられる。

C_2 の Jacobian の位数は

$$
\begin{aligned}
\#\mathcal{J}(\mathbb{F}_p) &= 6299865836712413622667460934772904 \\
&\quad 99872825577136 \\
&= 2^4 \cdot 3^2 \cdot 83 \cdot 197297 \cdot 286001 \cdot 297757 \\
&\quad \cdot 912227 \cdot 3088957 \cdot 1113335142470003
\end{aligned}
$$

で与えられる。$\#\mathcal{J}(\mathbb{F}_p)$ は 159bit であった。また、C_2 の 2 次ツイストの Jacobian の位数は

$$
\begin{aligned}
\#\mathcal{J}_t(\mathbb{F}_p) &= 6299865836708208762768577117863329 \\
&\quad 11267179858676 \\
&= 2^2 \cdot 31 \cdot 5771291 \cdot 304839203 \\
&\quad \cdot 2887791443224753474965528975763
\end{aligned}
$$

で与えられる。$\#\mathcal{J}_t(\mathbb{F}_p)$ の最大素因数は 102bit であり $\mathcal{J}_t$ は暗号利用には適さない。

C_1 に対する計算では、Algorithm 2 の Step 1-8 は 1 回のみ実行された。Algorithm 2 の Step 8 の計算で $\#\mathcal{J}(\mathbb{F}_p)/20$ が求まり、Algorithm 2 の終了時点で $\#\mathcal{J}(\mathbb{F}_p)/4$ が求まった。そして、Algorithm 1 の Step 3 で $\#\mathcal{J}(\mathbb{F}_p)$ が求まった。

C_2 に対する計算においても、Algorithm 2 の Step 1-8 は 1 回のみ実行された。C_2 に対しては、Algorithm 2 の Step 8 の計算で $\#\mathcal{J}(\mathbb{F}_p)/4$ が求まり、Algorithm 2 の出力も $\#\mathcal{J}(\mathbb{F}_p)/4$ であった。そして、Algorithm 1 の Step 3 で $\#\mathcal{J}(\mathbb{F}_p)$ が求まった。

位数計算に成功した曲線 C_1, C_2 に対する Algorithm 2 の実行時間の平均と主要ステップの実行時間の平均を表 4 の “B-easy” に示す。

Algorithm 1 の Step 3 の計算時間は計算に成功した 2 本の曲線の平均が 17 秒であった。

計算に失敗した 3198 本はすべて Algorithm 2 の 1 回目の Step 5 で計算が中断された。計算に失敗した曲線に対する Algorithm 2 の実行時間の平均と主要ステップの実行時間の平均を表 4 の “B-hard” に示す。

表 4: 26bit の B に対する Algorithm 2 の実行時間（秒）

Step	4	5	7	9-15	合計
B-easy	1353	1524	137	0	3014
B-hard	1332	2441	–	–	3773

計算に失敗した曲線に対する Algorithm 2 の Step 5 の実行時間が計算に成功した場合と比較して大きいが、Step 5 に利用している baby-step giant-step アルゴリズムは計算範囲を越えた位数に対して最も時間がかかることによる。位数計算に成功する場合にも Step 5 に最悪ではこの程度の時間が必要となると考えられる。

6.2 28bit の B を用いた実験

28bit の $B = 201326591$ を選択しランダムな 700 本の種数 2 の超楕円曲線 C に対して Algorithm 2 を適用した。その結果、1 本の曲線の位数計算に成功した。$B = 201326591$ に対して $u = \log_{2^{160}} B \approx 5.800$ より、式 (12) を用いて評価すると $\sigma(u) \approx 1.956\text{e-}3$ であり、位数計算に成功する曲線が 500 本に 1 本程度存在すると見積もられる。本実験の結果はこの見積りに矛盾しない。

位数計算に成功した曲線は

$$
\begin{aligned}
C_3 : Y^2 &= X^5 + 552762277014110066567019X^3 \\
&\quad +173431987584674038834235X^2 \\
&\quad +740422370910667228591845X \\
&\quad +438535256292463903908449
\end{aligned}
$$

である。C_3 の Jacobian の位数は

$$\begin{aligned}\#\mathcal{J}(\mathbb{F}_p) &= 6299865836701364581413939014020575\\ &\quad 86100719385872\\ &= 2^4 \cdot 61^2 \cdot 1306051 \cdot 3058229 \cdot 93116659\\ &\quad \cdot 146370013 \cdot 194375719136689\end{aligned}$$

である。$\#\mathcal{J}(\mathbb{F}_p)$ は 159bit である。また、C_3 の 2 次ツイストの Jacobian の位数は

$$\begin{aligned}\#\mathcal{J}_t(\mathbb{F}_p) &= 6299865836719257804022099983800611\\ &\quad 39717556378672\\ &= 2^4 \cdot 97 \cdot 44959 \cdot 354314243\\ &\quad \cdot 25482048040193200577075035167103\end{aligned}$$

で与えられる。$\#\mathcal{J}_t(\mathbb{F}_p)$ の最大素因数は 104bit であり $\mathcal{J}_t$ は暗号利用には適さない。

C_3 に対する計算においても、Algorithm 2 の Step 1-8 は 1 回だけ実行された。Algorithm 2 の Step 8 の計算で $\#\mathcal{J}(\mathbb{F}_p)/8$ が求まり、Algorithm 2 の終了時点で $\#\mathcal{J}(\mathbb{F}_p)/4$ が求まった。そして、Algorithm 1 の Step 3 で $\#\mathcal{J}(\mathbb{F}_p)$ が求まった。

C_3 に対する Algorithm 2 の実行時間と主要ステップの実行時間を表 4 の “B-easy” に示す。

Algorithm 1 の Step 3 の計算時間は 0.3 秒であった。

26bit の B を利用したときと同様に、計算に失敗した曲線はすべて Algorithm 2 の 1 回目の Step 5 で計算が中断された。計算に失敗した曲線に対する Algorithm 2 の実行時間の平均と主要ステップの実行時間の平均を表 4 の “B-hard” に示す。

表 5: 28bit の B に対する Algorithm 2 の実行時間（秒）

Step	4	5	7	9-15	合計
B-easy	5678	5439	4577	0	15695
B-hard	5483	10149	–	–	15632

26bit の B と比較すると実行時間が 4 倍程度長くなっているが、B が約 4 倍大きいので妥当な結果であると考えられる。

表 4,5 から得られる曲線 1 本あたりの計算時間と位数計算に成功するまでに必要な試行回数を乗ずることで、26bit の B では位数計算に成功するまでに約 1890 時間、28bit の B では約 2170 時間を必要とし、B を 26bit に設定した方が約 15%効率がよいことが分かる。したがって、最適な B の選択に関する議論の妥当性が裏付けられたが、一方で最適値に近い B を利用する限りは計算量が B の選択に敏感ではないので、本論文で示したパラメータ設定手順によって広い範囲の位数に対して効率的な位数計算が可能であると考えられる。

謝辞　本研究は JSPS 科研費 25400055 の助成を受けたものです。

参考文献

[ADH94] 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*, Algorithmic Number Theory - ANTS-I, LNCS 877, Springer, 1994, pp. 28–40.

[AH96] L. M. Adleman and M. D. Huang, *Counting rational points on curves and Abelian varieties over finite fields*, ANTS-II, LNCS 1122, Springer, 1996, pp. 1–16.

[BP96] E. Bach and R. Peralta, *Asymptotic semismoothness probabilities*, Math. Comp. **65** (1996), no. 216, 1701–1715.

[GH00] P. Gaudry and R. Harley, *Counting points on hyperelliptic curves over finite fields*, ANTS-IV, LNCS 1838, Springer, 2000, pp. 313–332.

[GS04] P. Gaudry and É. Schost, *Construction of secure random curves of genus 2 over prime fields*, EUROCRYPT 2004, LNCS 3027, Springer, 2004, pp. 239–256.

[GS12] ———, *Genus 2 point counting over prime fields*, J. Symbolic Comput. **47** (2012), 368–400.

[Kam91] W. Kampkötter, *Explizite Gleichungen für Jacobische Varietäten hyperelliptischer Kurven*, Ph.D. thesis, GH Essen, 1991.

[Ked01] K. S. Kedlaya, *Counting points on hyperelliptic curves using Monsky-Washinitzer cohomology*, J. Ramanujan Math. Soc. **16** (2001), no. 4, 323–338.

[LL06] R. Lercier and D. Lubicz, *A quasi quadratic time algorithm for hyperelliptic curve point counting*, The Ramanujan J. **12** (2006), no. 3, 399–423.

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

[MCT02] 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, LNCS 2369, Springer, 2002, pp. 461–474.

[Mes02] J.-F. Mestre, *Algorithme pour compter des points de courbes en petite caractéristique et petit genre*, `http://webusers.imj-prg.fr/~jean-francois.mestre/rennescrypto.ps`, 2002.

[Pil90] J. Pila, *Frobenius maps of Abelian varieties and finding roots of unity in finite fields*, Math. Comp. **55** (1990), 745–763.

[Sag] *Sage - open-source mathematical software system*, `http://www.sagemath.org/`.

[Sch85] R. Schoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Math. Comp. **44** (1985), 483–494.

[Sch95] ———, *Counting points on elliptic curves over finite fields*, J. Théorie des Nombers de Bordeaux **7** (1995), 219–254.

[Sha71] D. Shanks, *Class number, a theory of factorization, and genera*, Proc. of Symp. Math. Soc. 20, 1971, pp. 415–440.

[Sut07a] A. V. Sutherland, *Gallery of Jacobians*, `http://www-math.mit.edu/~drew/ZetaFunctions.html`, 2007.

[Sut07b] ———, *Order computations in generic groups*, Ph.D. thesis, Massachusetts Institute of Technology, 2007.

[Sut09] ———, *A generic approach to searching for Jacobians*, Math. Comp. **78** (2009), 485–507.

[Ver03] F. Vercauteren, *Computing zeta functions of curves over finite fields*, Ph.D. thesis, Katholieke Universiteit Leuven, 2003.