Memoirs of the Faculty of Science
Kochi University (Information Science)
Vol. 28 (2007), No. 1

楕円曲線の位数計算アルゴリズムの研究

森陽介、塩田研一
高知大学大学院理学研究科数理情報科学専攻

概 要

現在、公開鍵暗号で最も利用されているのは素因数分解の困難性を利用した RSA 暗号であるが、コンピュータの性能向上により従来の鍵サイズでは必ずしも安全ではなくなりつつある。それに伴い、注目されているのが楕円曲線を用いた暗号理論である。有限体上の楕円曲線は無数に存在するが、任意の楕円曲線が必ずしも暗号化に適しているとは限らない。Pohlig-Hellman 法による攻撃を回避するためには、曲線の位数を調べる必要がある。本研究では楕円曲線の位数計算について研究および実験を行った。

1 はじめに

有限体上の楕円曲線の位数を計算する方法に Schoof 法、および Schoof-Elkies-Atkin 法がある。

位数を求める際に、楕円曲線の Frobenius 写像のトレース t を計算しなければならない。Schoof 法はこれを l 一等分多項式と呼ばれる、約 $\frac{l^2}{2}$ 次の多項式を用いて計算するのに対し、Schoof-Elkies-Atkin 法はモジュラー多項式を用いることにより約 l 次の多項式処理で済ませるものである。

Schoof-Elkies-Atkin 法では小さな素数 l は Elkies 素数と Atkin 素数に分類される。このうち Atkin 素数たちは $t \mod l$ の候補値を複数生み出すため、Atkin 素数が多い程 t の候補値も指数関数的に増大してしまう。そこで、（1）Atkin 素数の情報を統合する Baby Step-Giant Step アルゴリズムを実行する際に、候補値が均等に分布するよう素数を組み分けをする。（2）候補値を多く持つ Atkin 素数を破棄して代わりに Elkies 素数を新たに採用する、などの改良を加えた。

2 Schoof 法

有限体 F_q 上の楕円曲線

$$E : y^2 = x^3 + ax + b, \ \ (\ a, b \in F_q\ )$$

における有理点の個数を $N = \sharp E(F_q)$ と表す。E 上の Frobenius 写像のトレースを t と表すと $N = q+1-t$ が成り立ち、Hasse の定理より $|t| \leq 2\sqrt{q}$ となることが知られている。

Schoof 法は位数 N を多項式時間で計算するものである。アイデアとして Frobenius 写像を利用して、十分たくさんの素数 l に対して、$N \mod l$ の値を特定し、中国剰余アルゴリズムを用いて、N の真の値を求めるものである。

Frobenius 写像 $\psi : (x, y) \to (x^q, y^q)$ については

$$\psi^2 - t\psi + q = 0$$

が成り立つ。

次に $E[l]$ を E の l 分点全体とし、$\psi^2 - t\psi + q = 0$ を $E[l]$ 上で考えると、

$$(\psi^2 - t\psi + q)P = 0 \quad \forall P \in E[l]$$

である。この P について

$$(\psi^2 - t'\psi + q)P = 0$$

となるような t' が見つけられたとすれば、

$$t \equiv t' \mod l$$

である。$l_1 \times l_2 \times l_3 \times \cdots \times l_{max} > 4\sqrt{q}$ を満たすような十分たくさんの素数 l_i について $t \equiv t_i \mod l_i$ を満たす t_i を計算し、中国剰余定理を用いると解 t は $\mod l_1 \times l_2 \times l_3 \times \cdots \times l_{max}$ で確定する。

Hasse **の定理**

$$|t| \leq 2\sqrt{q}$$

より、これで t の値が確定し、N が求められる。

3 Schoof-Elkies-Atkin 法

3.1 Schoof-Elkies-Atkin 法

Schoof-Elkies-Atkin 法は、 Schoof 法の計算上の実効性を改良するために開発された。Schoof 法では、計算時に次数の非常に高い l-等分多項式を使う必要があった。これに対し、 Schoof-Elkies-Atkin 法はより小さい次数の多項式で処理するものである。その主要部分は l を法としてとった Frobenius 写像の特性方程式

$$W_l(\psi) = \psi^2 - t_l\psi + q_l = 0$$

の根が素数 l を法として有限体 F_l に属するかどうか、すなわち判別式 $\Delta_t = t^2 - 4q$ が l を法として平方数であるかどうかに依存している。

平方数である場合は、『Elkies 素数』といわれ、そうでないときは『Atkin 素数』といわれる。この『Elkies 素数』、『Atkin 素数』を使い、位数を求めるのに必要なトレース t を求める。

3.2 Elkies 素数

Elkies 素数の場合、 Frobenius 写像の固有値を求めることから l を法としたときのトレースの値を求めることができる。

l は奇素数で、体の標数 p と l は異なるものとする。l が Elkies 素数であるとき判別式 Δ_t は有限体 F_l で平方数であり、 ψ の特性多項式 W_l は 2 つの根 λ と μ を有限体 F_l に持つ。それらは l を法とした Frobenius 写像の固有値である。

Frobenius の特性方程式は

$$\psi^2 - t\psi + q = (\psi - \lambda)(\psi - \mu)$$

である。これより、

$$(x^q, y^q) = [\lambda](x, y) \ \cdots \ (*)$$

の関係がある。

この関係を使い、$1 \leq \lambda \leq l-1$ の間でループを回して固有値 λ を見つけ出す。実際には式 $(*)$ の x 座標は正負に関わらず同じなので $1 \leq \lambda \leq \frac{l-1}{2}$ でループを回して符号の正負については y 座標について計算させて決定する。固有値 λ が見つかったら、

$$\mu \equiv \frac{q}{\lambda} \mod l,$$
$$t \equiv \lambda + \mu \mod l$$

とおき、トレースを求める。

基本的には Schoof のアルゴリズムと同様の手順をとるが、相違点として以下の 2 つが挙げられる。

- 等分多項式 $f_l(x)$ の代わりにモジュラー多項式を使って求めた多項式 $F_l(x)$ を使う。
- Elkies 素数 l に対して Frobenius 写像の固有空間がねじれ群の中にあることを保証する。

これらによって、計算時の次数の軽減と、(x^{q^2}, y^{q^2}) の計算が不要になった。

3.3 Atkin 素数

$\Delta_t = t^2 - 4q$ が素数 l を法として平方数でないとき、 l は Atkin 素数と呼ばれる。

Atkin 素数では、 Elkies 素数の場合とは異なり、Atkin 素数 l を法としたときのトレースがとり得る値（候補値）の部分集合を求める。

Elkies 素数の場合と同様に、多項式 $\psi^2 - t\psi + q \pmod l$ を分解する。素数 l を法としては Δ_t が平方数でないため、 Atkin 素数においてこの多項式は F_{l^2} で分解し、2 つの根 $\lambda, \mu \in F_{l^2} - F_l$ を持つ。 $\gamma_r = \frac{\lambda}{\mu}$ がとり得る値は限られ、

$$t \equiv \lambda + \mu \pmod l \ , \ q \equiv \lambda\mu \pmod l$$

からトレース $t_l \equiv t \pmod l$ に対する可能な値の集合が得られる。

4 Schoof-Elkies-Atkin 法

ここでは，有限体を素体 F_p に限定する。

アルゴリズム (Schoof-Elkies-Atkin)
入力：有限体 F_p 上の楕円曲線 E
出力： $E(F_p)$ の位数

1. 素数 l の上限設定。
2. $M \leftarrow 1, l \leftarrow 2, A \leftarrow \{\}, E \leftarrow \{\}$,
3. While $M < 4\sqrt{p}$ do:

4. Atkin 素数か Elkies 素数かの決定。
5. If l が Elkies 素数
6. then Elkies 素数の場合の計算。
7. If l が Atkin 素数
8. then Atkin 素数の場合の計算。
9. $M \leftarrow M \times l$,
10. $l \leftarrow next\ prime(l)$,
11. トレースの決定ステップ。
12. Return $p+1-t$.

5 素数 l の上限設定

Schoof のときと同様に

$$\prod_{\substack{l\ :\ prime \\ 2 \le l \le l_{max}}} > 4\sqrt{p}$$

となるような素数 l_{max} を求める。

6 Elkies 素数か Atkin 素数かの決定

アルゴリズム (Elkies **素数か** Atkin **素数かの決定**)
入力：体の標数 p , 素数 l , モジュラー多項式 Φ_l , j -不変量 j
出力： 素数 l が、Elkies 素数か Atkin 素数かの決定

1. $l=2$ でのトレースの決定。
2. $l \geq 3$ について、$GCD(\ x^p - x, \Phi_l(x,j)\ )$ を計算。
3. Step 2 で求めた方程式の次数より、Elkies 素数か Atkin 素数かの決定。

6.1 Step 1 について

$l=2$ のときに限り、$y=0$, すなわち、

$$x^3 + ax + b \ = \ 0$$

が F_p で解を持つならば、

$$t \equiv 0 \mod 2 ,$$

解を持たなければ、

$$t \equiv 1 \mod 2$$

となる。

6.2 Step 2 について

$\Phi_l(x,j)$ はモジュラー多項式、$j = 1728\frac{4a^3}{4a^3+27b^2}$ を表している。

6.3 Step 3 について

Step 2 で求めた方程式の次数が 1,2, $l+1$ ならば Elkies 素数、 0 ならば Atkin 素数である。

7 Elkies 素数の場合の計算

アルゴリズム (Elkies **素数**)
入力： $a,b \in$ 有限体 F_p, $F_l(x) \in F_p[x]$
出力：固有値 c (mod l)

1. For $n=-1$ to $\frac{l+3}{2}$ do:
2. $\bar{f}_n$ から $\psi_n(x,y) \mod F_l(x)$ を求める。
3. $l_x(x) \leftarrow x^p \mod F_l(x)$,
4. $l_y(x,y) \leftarrow y(x^3+ax+b)^{\frac{p-1}{2}} \mod F_l(x)$,
5. For $n=-1$ to $\frac{l-1}{2}$ do:
6. $r_x(x,y) \leftarrow x(x,y) + {\psi_{\alpha}}^2(x,y) - \psi_{\alpha-1}(x,y) \times \psi_{\alpha+1}(x,y)$,
7. If $l_x(x) \times {\psi_{\alpha}}^2(x,y) \equiv rx(x,y) \mod F_l(x)$
8. then $r_y(x,y) \leftarrow \psi_{\alpha+2}(x,y) \times \psi^2_{\alpha-1}(x,y) + \psi_{\alpha-2}(x,y) \times \psi^2_{\alpha+1}(x,y)$,

9. If $4y \times l_y(x,y) \times \psi_{\mathfrak{a}}^3(x,y) \equiv r_y(x,y) \mod F_l(x)$
10. then Return $c \equiv \mathfrak{a} + p \times \mathfrak{a}^{-1} \mod l$.
11. If $4y \times l_y(x,y) \times \psi_{\mathfrak{a}}^3(x,y)$
$\equiv -r_y(x,y) \mod F_l(x)$
12. then Return $c \equiv -\mathfrak{a} - p \times \mathfrak{a}^{-1} \mod l$.

l 次等分多項式をそのまま使うのではなく、l 次等分多項式の因子である次数 $\frac{l-1}{2}$ の多項式 $F_l(x)$ を法として等分多項式の漸化式を計算する。

8 Atkin 素数の場合の計算

アルゴリズム (Atkin **素数**)
入力：有限体 F_p 上の楕円曲線 E と素数 l
出力：(T,l) T はトレース t (mod l) のとり得る値の集合

1. $T \leftarrow \{\}$,
2. r の決定。
3. $F_{l^2} = F_l[\sqrt{d}]$ の生成元 g を決定。
4. $\Gamma \leftarrow \{g^{\frac{i(l^2-1)}{r}} : GCD(i,r) = 1\}$,
5. For each $\gamma_r \in \Gamma$ do:
6. $\gamma_r = g_1 + g_2\sqrt{d}$ と表す。
7. $z \leftarrow \frac{p(g_1+1)}{2}$ (mod l) ,
8. If z は l を法として平方数 then do:
9. $x \leftarrow \sqrt{z}$ (mod l) ,
10. $T \leftarrow T \cup \{2x, -2x\}$,
11. Return (T,l) .

8.1 Step 2 について

実際には、

$$x^{p^r} - x \equiv 0 \text{ (mod } \Phi_l(x,j) \text{)}$$

となるような r のことである。ここで、元 γ_r は位数が F_{l^2} においてちょうど r の元である。

8.2 Step 3 について

平方非剰余 $d \in$ 有限体 F_l 、生成元 $g = g_a + g_b\sqrt{d}$ とおく。

$l^2 - 1$ を素因数分解する。

$$l^2 - 1 = {q_1}^{e_1} \times {q_2}^{e_2} \times \cdots \times {q_i}^{e_i} .$$

$1 \le g_a, g_b < l$ の中から、

$$g^{\frac{l^2-1}{q_1}} \neq 1, g^{\frac{l^2-1}{q_2}} \neq 1, \cdots, g^{\frac{l^2-1}{q_i}} \neq 1$$

となる g_a, g_b を見つけて、生成元 g を決定する。(l が小さいので単純検索でもよい。)

8.3 Step 4 について

r と素な $i \in 1, \cdots, r-1$ について

$$g^{\frac{i(l^2-1)}{r}} = (g_a + g_b\sqrt{d})^{\frac{i(l^2-1)}{r}}$$

を計算して、集合 Γ に入れる。

8.4 Step 5 から 10 について

$\lambda = x_1 + x_2\sqrt{d}$, $\gamma_r = g_1 + g_2\sqrt{d}$ と表す。x_1, x_2 の値は未知だが、g_1, g_2 のとり得る値の候補は集合 Γ に格納済みである。

μ は λ の共役なので、$\mu = x_1 - x_2\sqrt{d}$ となる。よって、

$$p \equiv \lambda\mu \equiv x_1^2 - dx_2^2 \text{ (mod } l \text{) } \cdots (1)$$

また、

$$g_1 + g_2\sqrt{d} = \gamma_r = \frac{\lambda}{\mu} = \frac{\lambda^2}{\lambda\mu}$$

$$= \frac{1}{p}(x_1^2 + dx_2^2 + 2x_1x_2\sqrt{d})$$

より、

$$pg_1 \equiv x_1^2 + dx_2^2 \text{ (mod } l \text{) } \cdots (2)$$

$$pg_2 \equiv 2x_1x_2 \text{ (mod } l \text{)}$$

である。(1)、(2) より $x_1^2 = \frac{p(g_1+1)}{2}$ となる。

ここからトレースの候補値は $t \equiv \pm 2x_1$ (mod l) として得られる。

9 トレースの決定

アルゴリズム（ **トレースの決定** ）
入力：有限体 F_p 上の楕円曲線 E、Elkies 素数・Atkin 素数で計算された結果
出力：有限体 F_p 上の楕円曲線 E のトレース t (mod l)

1. Elkies 素数から得られた情報は中国剰余定理を使って、1つ (t_3, m_3) に統合する。
   $t \equiv t_3 \mod m_3$, （$m_3 = Elkies$ 素数の全ての積）.
2. Atkin 素数は2つの集合に分け、それぞれのトレースの候補を中国剰余定理を使って (r_1, r_2) にまとめる。（後述）
3. 楕円曲線上の任意の点 P を取る。
4. $[p+1-t_3]P - [r_1 m_2 m_3]P$ を計算して、r_1 の値とともに表に格納する。
5. r_2 のとりうる値に対して、$[r_2 m_1 m_3]P$ を計算して Step 4 で求めた表と照合する。
6. 一致する値があれば、r_2 の値が決定する。
7. $t = t_3 + m_3(m_1 r_2 + m_2 r_1)$ を計算してトレースを決定する。

9.1 Step 2 について

Atkin 素数を2つの部分集合 S_1,S_2 に分け、それぞれすべての候補値に中国剰余定理を使って候補値を絞り込み、

- S_1 に関する候補を
$$t \equiv t_1 \ (\mod m_1),$$
- S_2 に関する候補を
$$t \equiv t_2 \ (\mod m_2)$$

とする。ここで、m_1, m_2, m_3 は互いに素である。
次に，すべての t_1, t_2 に対して、

$$r_1 \equiv \frac{(t_1 - t_3)}{m_2 m_3} \mod m_1,$$

$$r_2 \equiv \frac{(t_2 - t_3)}{m_1 m_3} \mod m_2$$

を計算して、さらに絞り込む。ここで、

- 2つの集合への分割方法の選択、
- 候補値を多く持ち過ぎる Atkin 素数の削除

が、大きく計算量に影響するので工夫を要する。

10 実験

開発言語：Mathematica
実験環境：CPU　2 GHz PowerPC G5
　　メモリ　1 GB DDR SDRAM
　　OS　Mac OS X

10.1 Schoof 法と Schoof-Elkies-Atkin 法による位数計算時間の比較

素数 p の値をかえて $E : y^2 = x^3 + ax + b \quad a, b \in F_p$ の有理点の個数を計算。

10.2 Atkin 素数の分割方法別の比較

Schoof 法と Schoof-Elkies-Atkin 法の中の Baby Step-Giant Step において、Atkin 素数を2つに分けなければならない。
そこで、Atkin 素数の分割法別に比較し、位数計算にどのくらいの影響があるか実験した。

- 単純に半分に分割。
- 一方が少なく分割。
- 各 Atkin 素数が持つ候補値の個数が同数になるように分割。

10.3 Baby Step-Giant Step 改良 (1)・(2)

Baby Step-Giant Step の改良により、位数計算全体の計算時間の短縮を試みた。改良前の Baby Step-Giant Step 法との比較も行った。

10.4 単純 Schoof-Elkies-Atkin 法と Atkin 素数の削除を用いた改良 Schoof-Elkies-Atkin 法との比較

Atkin 素数の中で候補値が多いものは、検索数を飛躍的に増大させてしまうので削除する。その代わり新しい素数をリストに加える。

単純 Schoof-Elkies-Atkin 法との比較を行った。

11 実行結果

11.1 Schoof 法と Schoof-Elkies-Atkin 法による位数計算時間の比較

楕円曲線 $E : y^2 = x^3 + ax + b \quad a, b \in F_p$
SEA 法 : Schoof-Elkies-Atkin 法

標数 p の bit 数	SEA 法での計算時間 (秒)	Schoof 法での計算時間 (秒)
10	8.4	1.6
20	2.4	12.4
30	6.6	384.3
40	8.3	92,959
50	16.1	-
80	127.6	-
100	1,084	-
120	6,728	-
160	160,272	-
180	177,446	-

標数 p の bit 数を 10bit からはじめ徐々に桁数を上げていった。

30bit を過ぎると Schoof 法は飛躍的に計算時間が延びた。これより先の計測は時間がかかりすぎてしまうため、計測を省略した。

一方、SEA 法は順調に計算を続け、高速に位数を計算していることがわかる。さすがに 120bit を超えると計算時間はかかるが、楕円曲線暗号が安全であると言われる鍵長 170bit であり、それを超える 180bit の位数計算に成功した。

11.2 Atkin 素数の分割方法別の比較

11.2.1 Atkin 素数の分割方法

BS-GS 法 : Baby Step-Giant Step 法

(a) 単純に半分に分割。

(b) 一方が少なく分割。

(c) 各 Atkin 素数が持つ候補値の個数が同数になるように分割。

11.2.2 標数 p=120892581961462917470 6189 (80 bit)

位数計算時に使われる全素数
= { 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41 }

Atkin 素数の各分割法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
(a)	6.5	103.8
(b)	43.6	135.7
(c)	5.8	104.5

Atkin 素数 = { 7, 13, 19, 23, 31, 37 }
各 Atkin 素数の候補値の個数
= { 4, 6, 8, 4, 2, 18 }

11.2.3 標数 p=123456789012345678901 2353 (80 bit)

位数計算時に使われる全素数
= { 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41 }

Atkin 素数の各分割法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
(a)	22.5	218.7
(b)	98.4	283.5
(c)	7.0	220.6

Atkin 素数 = { 5, 11, 19, 29, 37, 41 }
各 Atkin 素数の候補値の個数
= { 2, 4, 4, 4, 18, 12 }

Atkin 素数の候補値の数がほぼ同数で分布しているため (a),(c) による違いはほとんど見られないが、(b) の Baby Step-Giant Step にかかった時間・全体の計算時間ともに増加が見られた通り、分割法によって計算時間は影響を受けることがわかる。

11.2.4 標数 p=126765060022822940149 6703205653 (100 bit)

位数計算時に使われる全素数
= { 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43 }

Atkin 素数の各分割法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
(a)	783.2	1,084.8
(b)	-	-
(c)	135.0	432.9

Atkin 素数 = { 3, 13, 17, 19, 23, 29, 31, 37, 41 }
各 Atkin 素数の候補値の個数
= { 2, 6, 6, 4, 4, 8, 16, 18, 6 }

11.2.5 標数 p=126765060024057508039 7937773317 (100 bit)

位数計算時に使われる全素数
= { 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43 }

Atkin 素数の各分割法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
(a)	1,430	1,938
(b)	-	-
(c)	65.1	550.2

Atkin 素数 = { 3, 5, 7, 11, 17, 29, 31, 37, 41 }
各 Atkin 素数の候補値の個数
= { 1, 2, 4, 1, 6, 8, 16, 18, 12 }

11.2.6 標数 p=10^{40} + 3 (120 bit)

位数計算時に使われる全素数
= { 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59 }

Atkin 素数の各分割法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
(a)	4,391	6,728
(b)	-	-
(c)	187	1,868

Atkin 素数 = { 3, 5, 11, 17, 19, 23, 29, 47, 53 }
各 Atkin 素数の候補値の個数
= { 1, 1, 4, 6, 8, 8, 8, 16, 18 }

(a),(c) で計算時間が大きく異なった。(b) については他の実験結果より計算時間が飛躍的に増加することが予想されたため、実験を試みるも計算を途中で終了した。

11.3 Baby Step-Giant Step 改良 (1)

Baby Step では楕円曲線 E 上の任意の点 P を元に $a_1P, a_2P, \cdots, a_tP$ の形のリストを作成する。$a_1P, a_2P, \cdots, a_tP$ をひとつひとつ反復 2 倍法で計算するとその度に $P, 2P, 4P, \cdots$ を計算してしまう。そこで $a_1, a_2, a_3, \cdots, a_t$ を $a_1 < a_2 < \cdots < a_t$ とソートしておき、

$$a_2P = (a_2 - a_1)P + a_1P,$$

$$a_3P = (a_3 - a_2)P + a_2P,$$

$$\cdots$$

$$a_tP = (a_t - a_{t-1})P + a_{t-1}P.$$

と計算する。Giant Step も同様に計算する。

11.3.1 標数 p=126765060022822940149 6703205653 (100 bit)

位数計算法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
単純 BS-GS 法	135	432
改良 BS-GS 法	84	383

11.3.2 標数 p=10^{40} + 3 (120 bit)

位数計算法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
単純 BS-GS 法	216	1711
改良 BS-GS 法	105	1428

11.4 Baby Step-Giant Step 改良 (2)

改良 (1) と同様に、$a_1, a_2, a_3, \cdots, a_t$ を $a_1 < a_2 < \cdots < a_t$ とソートしておく。今回は、$a_1P, a_2P, \cdots, a_tP$ を以下のように計算する。

アルゴリズム (Baby Step)
入力：Baby Step の元となる値 $a_1, a_2, a_3, \cdots, a_t$
出力：集合 $\{a_1P, a_2P, \cdots, a_tP\}$

1. $a_1, a_2, a_3, \cdots, a_t$ を $a_1 < a_2 < \cdots < a_t$ とソートする。
2. $R = P, SET = \{\}, Q_1 = O, Q_2 = O, \cdots, Q_t = O$,
3. While $a_t > 0$ do:
4. For $n = 1$ to $n = t$ do:
5. If a_n が奇数 do:
6. $Q_n = Q_n + R$,
7. $a_n = a_n/2$,
8. $R = R + R$,
9. $SET \leftarrow SET \cup \{R\}$,
10. Return $\{Q_1, Q_2, \cdots, Q_t\}$.

Giant Step では、Baby Step のステップ 9 で作られた集合 SET を使う。集合 SET には、

$$\{P, 2P, 4P, 8P, 16P, 32P, \cdots\}$$

が格納されているので、Giant Step のリスト $b_1P, b_2P, \cdots, b_tP$ を反復 2 倍法で計算する際にその計算が省略できる。ただし、集合 SET を格納するだけのメモリが必要となる。

11.4.1 標数 p=12676506002405750803 97937773317 (100 bit)

位数計算法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
単純 BS-GS 法	65	550
改良 BS-GS 法	30	517

11.4.2 標数 p=10^{40}+1234567953 (120 bit)

位数計算法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
単純 BS-GS 法	92	2650
改良 BS-GS 法	40	2506

これら 2 つの改良により Baby Step-Giant Step 法の計算は元々の計算法よりも約半分の時間で計算が可能である。

11.5 単純 Schoof-Elkies-Atkin 法と Atkin 素数の削除を用いた改良 Schoof-Elkies-Atkin 法との比較

11.5.1 標数 p=12676506002282294014 96703217737 (100 bit)

Atkin 素数の各分割法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
(a)	8.0	267.5
(b)	15.4	263.5
(c)	1.5	262.0

位数計算時に使われる全素数
= { 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43 }
Atkin 素数 = { 3, 7, 11, 31, 37 }

標数 $p = 100$bit の位数計算時間は、通常 1000 秒を超える。しかし、上記の計算では、どの分割法を使っても通常の $\frac{1}{5}$ 程度の計算時間しかかかってない。これは、位数計算時に使われる全素数に対して Atkin 素

数の割合が他のものと比べ少ないからであると考えられる。

そこで、Atkin 素数を厳選し，候補値の総数を減らすことがどの程度位数計算時間に影響するか実験を行った。まず、Schoof-Elkies-Atkin 法に改良を加える。

- Atkin 素数の中で候補値の個数が 10 個を超える Atkin 素数は削除する。
- 削除した分は、初期設定した素数の上限を超えて新しい素数をリストに加える。

これら 2 つの改良を行った Schoof-Elkies-Atkin 法を改良 Schoof-Elkies-Atkin 法とする。以下は、単純 Schoof-Elkies-Atkin 法との比較を行った結果である。

楕円曲線　$E : y^2 = x^3 + ax + b \quad a, b \in F_p$
SEA 法 : Schoof-Elkies-Atkin 法
BS-GS 法 : Baby Step-Giant Step 法

11.5.2 標数 p=120892581961462917470 6189 (80 bit)

位数計算法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
単純 SEA 法	6.5	103.8
改良 SEA 法	0.4	244.1

初期設定された全素数
= { 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41 }
追加された素数 = { 43, 47 }

11.5.3 標数 p=126765060022822940149 6703205653 (100 bit)

位数計算法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
単純 SEA 法	783.2	1,084
改良 SEA 法	11.1	905.4

初期設定された全素数
= { 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43 }
追加された素数 = { 47, 53, 59 }

11.5.4 標数 p=10^{40} + 3(120 bit)

位数計算法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
単純 SEA 法	4,391	6,728
改良 SEA 法	61.3	27,122

初期設定された全素数
={ 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59 }
追加された素数={ 61, 67, 71, 73 }

11.5.5 標数 p=146150163733090291820 368483271628301965593254298 3 (160 bit)

位数計算法	BS-GS 法にかかった時間 (秒)	全体の計算時間 (秒)
単純 SEA 法	120,204	160,272
改良 SEA 法	0.2	220,101

初期設定された全素数
={ 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71 }
追加された素数={ 73, 79, 83, 89, 97, 101, 103, 107 }

6.3.2 において BS-GS 法にかかった時間・全体の計算時間ともに大幅に縮小した。しかし、その他を見てみると BS-GS 法にかかった時間は限りなく 0 秒に近いにも関わらず、全体の計算時間は縮小するどころか増大してしまっている。

全体の計算時間が伸びたものの共通点として、初期設定された素数に比べ追加された素数が多いこと挙げられる。追加された素数が多いということはその分、高次の多項式の計算が増える。その結果、全体の計算時間の増大に繋がったと考えられる。

12　結論

楕円曲線によって位数は変化し、楕円曲線のパラメータでは位数を推測することは出来ない。

位数計算アルゴリズムは代表的なものとして４つある。その内、Weil 予想を用いる方法や虚数乗法を用いる方法は高速に計算可能であるが、適用できる曲線が限定され、その限定が暗号解読の手がかりになりかねない。一番暗号への応用性・自由度が高い位数計算アルゴリズムは、Schoof 法や Schoof-Elkies-Atkin 法である。しかし、Schoof 法では実行時間が $O(\log^8 q)$ 程度と実用的ではなかった。

本研究で実装した Schoof-Elkies-Atkin 法は楕円曲線暗号が安全であると言われる鍵長 170bit を超える楕円曲線の位数を実用的な時間で計算することが出来た。

また、Baby Step-Giant Step 法において Atkin 素数をバランスよく２つのグループに分割することで計算時間は大きく節約できた。さらに Atkin 素数を厳選し、候補値の集合のサイズ全体に上限を与えることも一部では効果的であった。

しかし、Atkin 素数を削減することには課題が残った。削除した分だけ元々設定されていた素数の上限を超える素数を扱わなくてはならなくなるため、通常よりも高次の多項式を扱わなくてはならなくなってしまう点である。

Atkin 素数の削除による高次の多項式がもたらす計算時間の増加を加味しつつ、Atkin 素数の削除の基準をどこに設定するのかを今後の研究課題としたい。

参　考　文　献

1. イアン・F・ブラケ、ガディエル・セロッシ、ナイジェル・P・スマート　著　鈴木治郎　訳：楕円曲線暗号（ピアソン・エデュケーション）
2. Andreas Enge 著：Elliptic Curves and Their Applications to Cryptography, an Introduction（Kluwer Academic Publishers）
3. Neal Koblitz 著：A Course in Number Theory and Cryptography（Springer-Verlag）
4. 富士通研究所, ”数学データ モジュラー多項式 (Modular Polynomial)
”http://jp.fujitsu.com/group/labs/techinfo/freeware/modularpoly/ ”