2019 年度修士論文

楕円曲線上の離散対数問題

計算アルゴリズムの実装と比較

早稲田大学大学院基幹理工学研究科
数学応用数理専攻修士課程 2 年
5118A037-1
潮谷　真奈
指導教授：楫元

2020 年 2 月 7 日

1　はじめに

楕円曲線離散対数問題 (ECDLP) とは, 楕円曲線上の点 $P \in E(\mathbb{F}_q)$ と $Q \in \langle P \rangle$ が与えられたとき, 離散対数 $\log_P Q = \min\{j \in \mathbb{Z}_{\geq 0} \mid Q = jP\}$ を求める問題である. 現在, 超特異楕円曲線など一部の曲線については, 特有の構造を利用して ECDLP を解く方法が見つかっている (Weil pairing を用いた MOV 攻撃 ([1]p111) など). 一方, 一般的な楕円曲線における ECDLP は, P の位数 n に大きな素数が含まれる場合, 解読が困難な問題であるとされている ([1]p75). その困難性を利用した楕円曲線暗号は, 従来の RSA 暗号や ElGamal 暗号より安全性に優れたものとして, インターネットの暗号通信などに広く使われている ([1]p53,[2]p11).

このような ECDLP を効率的に解く方法として知られているのが, Shanks' Babystep-Giantstep Algorithm と Pollard's ρ algorithm である. これらは, どちらも計算量 $O(\sqrt{n})$ のアルゴリズムであるが, 保存すべき点を少なくできるという理由から, Pollard's ρ algorithm が優れているという見方もある ([3]p2). しかし, 具体的な楕円曲線において各アルゴリズムがどのように働くか, 例えば計算時間にどのような違いが生じるかは明らかでない. 特に, Pollard's ρ algorithm においては, 十分にランダムかつ効率的な関数 f を実現できるか否かによって, ステップ数や計算時間が大きく変動すると考えられる.

本論文では, 計算機を使用して ECDLP の作成と解読を行い, 総当たり法, Shanks' Babystep-Giantstep Algorithm, Pollard's ρ algorithm I, II のステップ数 s と計算時間 t [ms] を測定し, どれが最も効率的なアルゴリズムであるか調べた. また, n が十分大きい場合, Shanks' Babystep-Giantstep Algorithm と Pollard's ρ algorithm I, II のステップ数が理論的には $\sqrt{n}$ に比例することを踏まえて, 計算時間 t も $\sqrt{n}$ に比例する, すなわち $t/\sqrt{n}$ (本論文ではこれを計算量対時間比と呼ぶ) の値が定数になると考え, 実験によりこれを確かめた. さらにその結果から, 実際に暗号に使われているような, 位数の大きな楕円曲線における ECDLP について, 計算時間の予測を試みた. Pollard's ρ algorithm I, II における関数 f については, 直和分割の個数 m を 3〜50 の間で変化させて実験し, ステップ数や計算時間と m との関係を調べ, 「十分ランダムかつ効率的な関数 f」の実現に適した m の条件について考察した.

実験結果から, ステップ数が最少となるアルゴリズムは常に Rho-II(ただし m の値は十分大きいと仮定) であり, 計算時間が最短となるアルゴリズムは, 14 ビット以下においては BSGS, 16〜26 ビットについては Rho-II(8), 28 ビット以上については Rho-II(20) であるということがわかった. したがって, 標数が大きい場合については, ステップ数・計算時間共に Rho-II が最も効率的なアルゴリズムである, という結論が得られた. Pollard's ρ algorithm I, II における m に関しては, $m \geq 11$ であることが必要であり, 特に $m \geq 16$ であることが望ましい, という結論に至った. さらに, BSGS と Rho-II の計算量対時間比については, 定数に収束する傾向が観察できた. 特に Rho-II については, 30 ビットまで調べた時点で既に挙動が安定していた. この結果を利用し, 仮想通貨「ビットコイン」の安全性を保証するために使われている楕円曲線暗号について考察した. 現在, ビットコインでは「secp256k1」と呼ばれる楕円曲線に基づく暗号化技術が使用されている ([4]). そこで, secp256k1 における ECDLP について, 今回と同様の実験環境で Rho-II(20) を用いて計算した場合に要する時間を見積もってみたところ, 約 9.91×10^{26} 年という予測が得られた.

2　楕円曲線

はじめに, 楕円曲線に関する定義や定理を述べる.

定義 2.1 (楕円曲線)**.**
K を体, $\bar{K}$ を K の代数閉体とする.
種数 1 の非特異な代数曲線を, 楕円曲線という. 射影平面曲線としては, 以下のように書ける.

$$E : Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \ (a_1, a_2, a_3, a_4, a_6 \in \bar{K})$$

この楕円曲線は必ず, 無限遠点 $[0 : 1 : 0]$ を通る. これを O と表記する.

今後は簡単のため, $x = \frac{X}{Z}, y = \frac{Y}{Z}$ とおいてアフィン座標に変換した定義式

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \ (a_1, a_2, a_3, a_4, a_6 \in \bar{K})$$

を用いる.

また, $a_1, a_2, a_3, a_4, a_6 \in K$ のとき, E は K 上の楕円曲線であるといい, E/K と表記する.
さらに, E の K 有理点の集合を

$$E(K) = \{(x, y) \in K^2 \mid y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \{O\}$$

と表記する.

定理 2.1 (K 上同型な楕円曲線 ([5]p16))**.**
2 つの楕円曲線

$$E_1/K : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$
$$E_2/K : y^2 + \bar{a}_1xy + \bar{a}_3y = x^3 + \bar{a}_2x^2 + \bar{a}_4x + \bar{a}_6$$

が K 上同型であることは, E_1/K から E_2/K への変数変換

$$(x, y) \to (u^2x + r, u^3y + u^2sx + t) \ (u \in K^*, r, s, t \in K)$$

が存在することと同値である.

ここで, 標数 $\mathrm{char}(K) \neq 2, 3$ の場合について考える.
楕円曲線 $E/K : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ について, $(x, y) \to (x, y - \frac{a_1x+a_3}{2})$ と変換すると, E/K と同型な楕円曲線

$$E'/K : y^2 = x^3 + b_2x^2 + b_4x + b_6$$

が得られる. さらに $(x, y) \to (x - \frac{b_2}{3}, y)$ と変換すると, E'/K と同型な楕円曲線

$$E''/K : y^2 = x^3 + ax + b$$

が得られる.

定義 2.2 (Weierstrass の標準形)**.**
方程式 $y^2 = x^3 + ax + b$ を, Weierstrass の標準形という.

今後は, 簡単のため, $\mathrm{char}(K) \neq 2, 3$ の場合についてのみ考え, Weierstrass の標準形によって定義される楕円曲線 E/K を扱う.

3 楕円曲線の群構造

Bézout の定理より, 射影空間において楕円曲線と直線は (重複度を含めて) ちょうど 3 点で交わる. これを利用し, $E(K)$ に群演算を与える.

定理 3.1 (楕円曲線の加法 ([6]p187))**.**
与えられた有理点 $O' \in E(K)$ と任意の点 $P, Q \in E(K)$ について,

- $P \neq Q$ のとき, $P * Q$ は P, Q を通る直線と E とのもう 1 つの交点
- $P = Q$ のとき, $P * P$ は P における E の接線と E とのもう *1* つの交点
  (ただし P が変曲点のときは $P * P = P$)
- $P + Q = O' * (P * Q)$

と定義する.
このとき, $E(K)$ は, $+$ を二項演算として O' を単位元とする可換群とみなせる.

Proof.

可換 $P, Q \in E(K)$ について,

$$\begin{aligned} P + Q &= O' * (P * Q) \\ &= O' * (Q * P) \\ &= Q + P \end{aligned}$$

単位元 O' が単位元となることを示す. $P \in E(K)$ について,

$$\begin{aligned} O' + P = P + O' &= O' * (P * O') \\ &= P \end{aligned}$$

逆元 $O'' = O' * O'$ と定義し, $R = P * O''$ とおく.
このとき, 明らかに $O' * O'' = O', P * R = O''$ であるから,

$$\begin{aligned} R + P = P + R &= O' * (P * R) \\ &= O' * O'' \\ &= O' \end{aligned}$$

したがって, R は P の逆元である.

結合法則 [6]p188 参照.

□

以後, 無限遠点 O を単位元とみなす. これにより, $O * O = O$ となり, P の逆元 $-P$ は P と x 軸対称な点となる.

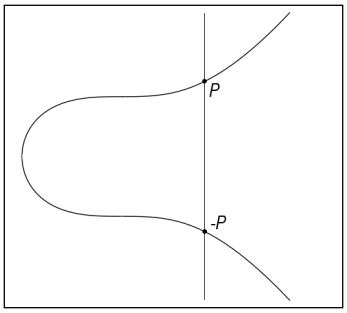

図 1　$-P$

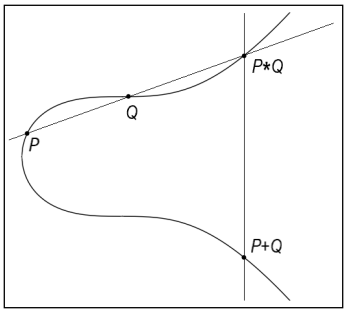

図 2　$P + Q$

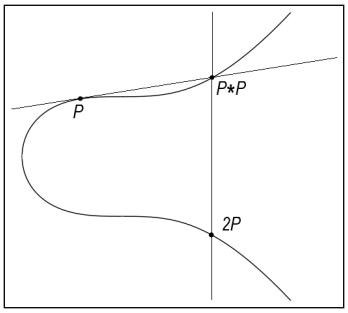

図 3　$2P$

$P, Q \in E(K) \setminus \{O\}$ について, $P + Q$ の座標を導く公式は以下の通り.

1. $P \neq \pm Q$ のとき
   $P = (x_1, y_1), Q = (x_2, y_2), P * Q = (x_3, y_3)$ とおくと,
   P と Q を結ぶ直線の方程式は

$$y = \lambda x + \nu \quad (\lambda = \frac{y_2 - y_1}{x_2 - x_1}, \nu = y_1 - \lambda x_1)$$

   これと Weierstrass の標準形を連立して解くことで,

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda x_3 + \nu$$

   $P + Q = (x_3, -y_3)$ であるから, $P + Q = (\lambda^2 - x_1 - x_2, -\lambda x_3 - \nu)$

2. $P = Q$ のとき
   $P = (x_1, y_1), P * P = (x_3, y_3)$ とおく.
   P が変曲点であれば, $2P = (x_1, -y_1)$
   P が変曲点でないとき, P における E の接線の方程式は

$$y = \lambda x + \nu \quad (\lambda = \frac{3x_1^2 + a}{2y_1}, \nu = y_1 - \lambda x_1)$$

   1. と同様にして,

$$x_3 = \lambda^2 - 2x_1$$
$$y_3 = \lambda x_3 + \nu$$

   よって, $P + Q = 2P = (\lambda^2 - 2x_1, -\lambda x_3 - \nu)$

3. $P = -Q$ のとき
   定義より, $P + Q = -Q + Q = O$

有限体上の楕円曲線の群構造に関しては, 次の定理が成り立つ.

定理 3.2 (有限体上の楕円曲線の群構造 ([1]p106))**.**
$K = \mathbb{F}_q$ (ただし q はある素数 p の冪) とすると, $E(K)$ は以下のいずれかを満たす.

1. $E(K)$ は巡回群
2. $E(K) \cong \mathbb{Z}/m_1\mathbb{Z} \times \mathbb{Z}/m_2\mathbb{Z} \quad (m_1 \mid m_2, m_1 \mid q-1)$

Proof. [1]p106 参照. □

4 楕円曲線離散対数問題 (ECDLP)

定義 4.1 (楕円曲線上の離散対数)**.**
$P \in E(\mathbb{F}_q)$ について, $\langle P \rangle \subset E(\mathbb{F}_q)$ を点 P によって生成される巡回部分群とおく. このとき, 任意の $Q \in \langle P \rangle$ に対して,

$$\log_P Q = \min\{j \in \mathbb{Z}_{\geq 0} \mid Q = jP\}$$

を, P を底とする Q の離散対数という.
また, P をベースポイントという.

定義 4.2 (楕円曲線上の離散対数問題 (ECDLP))**.**
P と Q が与えられたとき, 離散対数 $\log_P Q$ を求める問題を, 楕円曲線上の離散対数問題 (ECDLP) という.

一部の特殊な楕円曲線を除くと, P の位数に大きな素数が含まれる場合, ECDLP は解読に完全指数時間を要する困難な問題とされてる ([1]p75). これを利用した楕円曲線暗号は, RSA 暗号や ElGamal 暗号より安全性に優れ, 現在はインターネットの暗号通信プロトコルやビットコインにおけるディジタル署名などに利用されている ([2]p11,[4]).

5 ECDLP に対する攻撃アルゴリズム

ECDLP の解法の 1 つとして, 離散対数を総当たりで求める方法がある. これは, $P, 2P, 3P, \cdots$ と順に計算して $\log_P Q$ を求めるものであるが, 計算量は $O(n)$ となり, 非効率的である.
ここでは, ECDLP を解読するためのより効率的な方法について考察する.
以下, P の位数を n とおく.

5.1 Shanks' Babystep-Giantstep Algorithm

定理 5.1 ([7]p382)**.**
$N = \lceil \sqrt{n} \rceil = \min\{m \in \mathbb{Z}_{\geq 0} \mid m \geq \sqrt{n}\}$, $R = -NP$ とおく. このとき, $Q \in \langle P \rangle$ であれば, ある $0 \leq i, j < N$ が存在し, $iP = Q + jR$ が成り立つ.

Proof.

$Q = mP$ $(0 \leq m < n)$ と仮定する. $m = jN + i$ $(0 \leq i < N)$ と表すと,

$$0 \ \leq \ j = \frac{m-i}{N} \ < \frac{n-i}{\sqrt{n}} = \sqrt{n} - \frac{i}{\sqrt{n}} \ \leq \lceil \sqrt{n} \ \rceil = N$$

したがって仮定より,

$$\begin{aligned} Q &= mP \\ &= (jN + i)P \\ &= -jR + iP \\ iP &= Q + jR \ (0 \leq i, j < N) \end{aligned}$$

が成り立つ. □

この性質を利用したアルゴリズムが, Shanks' Babystep-Giantstep Algorithm ([7]p382) である.

Shanks' Babystep-Giantstep Algorithm

1. $N = \lceil \sqrt{n} \rceil = \min\{m \in \mathbb{Z}_{\geq 0} \mid m \geq \sqrt{n}\}$, $R = -NP$ を求める.
2. Babysteps : リスト $\{iP \mid 0 \leq i < N\}$ を作成する.
3. Giantsteps : $Q, Q + R, Q + 2R, \cdots$ を順に計算し,
   2. のリスト上の点と一致する $Q + jR$ $(0 \leq j < N)$ を見つける.
4. $iP = Q + jR$ であれば, $Q = (i + jN)P$ となり, $\log_P Q \equiv i + jN \pmod{n}$ が成立.

なお, $iP = Q + jR$ が見つかるまでのステップ数の期待値は, $\frac{3}{2}\sqrt{n}$ である ([8]p3).

5.2 Pollard's ρ algorithm

定義 5.1.

$S_0 \in E(\mathbb{F}_q)$ を起点とし, 点列 $S_1, S_2, \cdots$ を

$$\begin{aligned} &f : E(\mathbb{F}_q) \to E(\mathbb{F}_q), \\ &S_i = f(S_{i-1}) = \underbrace{f \circ \cdots \circ f}_{i}(S_0) \end{aligned}$$

によって定義する. ただし, f は $E(\mathbb{F}_q)$ 上の点を十分ランダムかつ効率的に定めることができる関数とする.

$E(\mathbb{F}_q)$ は有限群であるから, ある $t \in \mathbb{Z}_{\geq 0}, l \in \mathbb{N}$ が存在し, $S_t = S_{t+l}$ が成り立つ. このような t, l のうち最小のものをそれぞれ T, L と定義する.

また, $S_i = S_j$ かつ $i \neq j$ なる組 (S_i, S_j) を match という.

なお, $T + L$ の値の期待値は, $\sqrt{\frac{\pi n}{2}}$ である ([7]p383).

関数 f は, 例として以下のように定める ([9]p206).

random mapping f

1. 任意の $m \in \mathbb{N}$ を選び, $E(\mathbb{F}_q)$ を元の数がおおよそ等しい m 個の集合 $G_1, \cdots, G_m$ に直和分割する.
2. $a_1, \cdots, a_m, b_1, \cdots, b_m \in \{0, \cdots, n-1\}$ をランダムに選び,
   $M_l := a_l P + b_l Q$ $(l \in \{1, \cdots, m\})$ とおく.
3. $f : E(F_q) \to E(\mathbb{F}_q)$, $\underline{f(S) = S + M_l}$ $(S \in G_l)$ と定義する.

この性質を利用したアルゴリズムの 1 つが, Pollard's ρ algorithm I ([9]p205) である. (以下の Pollard's ρ algorithm I と後述する Pollard's ρ algorithm II は, どちらも Pollard's ρ algorithm と呼ばれるが, 本論文では区別するため I,II をつけて表記する.)

Pollard's ρ algorithm I

1. $a_0, b_0 \in \{0, \cdots, n-1\}$ をランダムに選び,
   $S_0 = a_0 P + b_0 Q$, $S_i = f(S_{i-1})$ $(i \in \mathbb{N})$ によって点列 (S_i) を定める.
2. $S_1, S_2, S_3, \cdots$ と順に求め, $S_i = S_j$ $(j \in \{0, 1, \cdots, i-1\})$ となる i を見つける.
3. $a_i P + b_i Q = a_j P + b_j Q$ であるから,
   $\gcd((b_j - b_i), n) = 1$ であれば, $\log_P Q \equiv (a_i - a_j)(b_j - b_i)^{-1} \pmod{n}$ となる.
   $\gcd((b_j - b_i), n) > 1$ であれば, 2. に戻ってやり直す.

関数 f は, 事前に M_l を求めずに

$$f : E(\mathbb{F}_q) \to E(\mathbb{F}_q),\ \underline{f(aP + bQ) = (a + a_l)P + (b + b_l)Q}\ (aP + bQ \in G_l)$$

と表現することもできる. すなわち,

$$a_i \equiv a_{i-1} + a_l,\ b_i \equiv b_{i-1} + b_l \pmod{n}\ \ (a_{i-1}P + b_{i-1}Q \in G_l)$$

によって定められる点列 $(a_i), (b_i)$ を利用して match を見つけることもできる. しかし, 計算量が増えて多大な時間がかかってしまうため, 実装の際は $a_1, \cdots, a_m, b_1, \cdots, b_m$ だけでなく $M_1, \cdots, M_m$ も先に求め, リストに格納しておく方がよい.

Shanks' Babystep-Giantstep Algorithm と Pollard's ρ algorithm I は, 一致する点を見つけるためにいくつかの点をリストに格納し続け, 新たに求めた点といちいち比較する必要がある. そこで最後に, 格納すべき点の個数をより少なく抑えたアルゴリズムについて述べる.

定理 5.2 ([7]p382)**.**
ある $T \leq i' \leq T + L - 1$ が存在し, $S_{i'} = S_{2i'}$ が成り立つ.

Proof.
定義から,

$$S_i = S_j \ \Leftrightarrow\ T \leq i \text{ かつ } i \equiv j \pmod{L}$$

すなわち,

$$S_i = S_{2i} \Leftrightarrow T \leq i \text{ かつ } i \equiv 0 \pmod{L}$$

明らかに, ある $k \in \{T, T+1, \cdots, T+(L-1)\}$ が一意的に存在し, $T \leq k$ かつ $k \equiv 0 \pmod{L}$ となるから, $k = i'$ であり, 定理が成り立つ. □

これを利用したアルゴリズムが, Pollard's ρ algorithm Ⅱ ([9]p206) である.

Pollard's ρ algorithm Ⅱ

1. $a_0, b_0 \in \{0, \cdots, n-1\}$ をランダムに選び,
   $S_0 = a_0P + b_0Q,\ S_i = f(S_{i-1})$,
   $T_0 = S_0 = a_0P + b_0Q,\ T_i = f \circ f(T_{i-1})$ $(i \in \mathbb{N})$ によって点列 $(S_i), (T_i)$ を定める.
2. $S_1, T_1, S_2, T_2, \cdots$ と順に求め, $S_i = T_i$ $(= S_{2i})$ となる i を見つける.
3. $a_iP + b_iQ = a_{2i}P + b_{2i}Q$ であるから,
   $\gcd((b_{2i} - b_i), n) = 1$ であれば, $\log_P Q \equiv (a_i - a_{2i})(b_{2i} - b_i)^{-1} \pmod{n}$ となる.
   $\gcd((b_{2i} - b_i), n) > 1$ であれば, 2. に戻ってやり直す.

S_0 や関数 f の定め方には様々な方法があるが, 今回は上記のように定義する.

Shanks' Babystep-Giantstep Algorithm と Pollard's ρ algorithm Ⅰ,Ⅱ を比較すると, 計算量は全て $O(\sqrt{n})$ であるが, 保存すべき点が少ないという点で Pollard's ρ algorithm Ⅱ の方が優れた方法であるとされている ([3]p2). しかし, 任意の ECDLP についてこれらのアルゴリズムが具体的にどのように働くかは, 明らかでない. そこで本論文では, 計算機を使用して実際に ECDLP の作成・解読を行い, 総当たり法と Shanks' Babystep-Giantstep Algorithm, Pollard's ρ algorithm Ⅰ,Ⅱ について, ステップ数 s と計算時間 t [ms] を計測した.

また, Shanks' Babystep-Giantstep Algorithm と Pollard's ρ algorithm Ⅰ,Ⅱ が計算量 $O(\sqrt{n})$ のアルゴリズムであることを考慮すると, n が十分大きい場合, ステップ数は $\sqrt{n}$ に比例した値になると考えられる. それに伴い, 計算時間も $\sqrt{n}$ に比例する, すなわち $t/\sqrt{n}$ の値が定数に収束するのではないかと予想できる. これが正しければ, 現実的には実験が不可能なほど位数が大きい楕円曲線における ECDLP についても, 解読に必要な計算時間を見積もることが可能となる. そこで, 各アルゴリズムにおける $t/\sqrt{n}$ の値も求め, 観察した. なお, 本論文では $t/\sqrt{n}$ を「計算量対時間比」と呼ぶ.

5.3 備考

アルゴリズム名の表記 : 簡単のため, 各アルゴリズムを以下のように省略して表記する.

- 総当たり法 → 総当たり
- Shanks' Babystep-Giantstep Algorithm → BSGS
- Pollard's ρ algorithm Ⅰ $(m = 3)$ → Rho-Ⅰ(3)
- Pollard's ρ algorithm Ⅱ $(m = 3)$ → Rho-Ⅱ(3)

6　実験方法

計算機を用いて, ECDLP の作成と解読を行い, 解読に要したステップ数と計算時間を測定した. なお, 本章に掲載したソースコードは, 全て今回の実験のために自作したものである.

6.1　計算環境

実験に用いた計算環境は, 以下の通りである.

- OS:Windows 10 Pro
- プロセッサ:Intel(R)Core(TM)i7-8650U CPU @1.90GHz 2.11GHz
- 実装 RAM:16.0GB
- システムの種類:64 ビットオペレーティングシステム, x64 ベースプロセッサ
- 数式処理プログラム:Sagemath 8.8

6.2　ECDLP の作成

以下に従って楕円曲線を選び, $\log_P Q$ を求める ECDLP を作成した.

1. 楕円曲線の定義式 :
   - $y^2 = x^3 + 7$
   - $y^2 = x^3 + 2$
   - $y^2 = x^3 - 3x + C_N$
     ($C_N := 189582862855666080004086685444939264155046809686793210757872346725 64$)
2. 係数体は, 以下の条件を満たす $\mathbb{F}_p$ とした.
   - p のビット数は $10, 12, 14, \cdots, 30$
   - 曲線が non-singular
   - 群 $E(\mathbb{F}_p)$ が素数位数の巡回群
3. 点 P, Q をランダムに選ぶ. これを任意の回数行うことで, ECDLP を複数作成した.

なお, 楕円曲線の定義式の選択は, SafeCurves([10]) を参考にした.

6.3　実験 1

総当たり, BSGS, Rho-I, Rho-Ⅱ で ECDLP の解読を行い, それぞれのステップ数を求めた. ただし, 1 つの $E/\mathbb{F}_p$ について ECDLP を 1000 問作成して解読し, そのステップ数の平均を, 各アルゴリズムが要したステップ数 s として定めた. Rho-I, Rho-Ⅱ における直和分割の個数は $m = 3, 4, \cdots, 50$ とした.

使用したソースコードは以下の通り. (例 : 定義式 $y^2 = x^3 + 7$, 標数 10 ビット)

ソースコード 1　ステップ数

```
import random
a=0
b=0
```

```
c=7
bit=10
repeat=10
end=0
while end==0:
    p=random.sample(list(primes(2^(bit-1),2^bit)),1)[0]
    d=(-4*a^3*c+a^2*b^2+18*a*b*c-4*b^3-27*c^2)%p
    if d!=0:
       K=GF(p)
       E=EllipticCurve(K,[0,a%p,0,b%p,c%p])
       o=E.order()
       if is_prime(o):
          print 標数"p=",p
          print "y^2 = x^3+",a%p,"x^2+",b%p,"x+",c%p
          print "order=",o
          end=1
list_Squ=[]
for j in range(0,(p+1)/2):
    list_Squ.append(j^2%p)
i=0
while (i^3+a*i^2+b*i+c)%p not in list_Squ:
      i+=1
j=list_Squ.index((i^3+a*i^2+b*i+c)%p)
Point=E([i,j])
step_so=0
step_BG=0
step_R1=[0,0,0]
step_R2=[0,0,0]
for i in range(3,21):
    step_R1.append(0)
    step_R2.append(0)
for rep in range(repeat):
    P=randint(1,o-1)*Point
    Q=randint(1,o-1)*Point
    def baby_giant(x,y):
        N=ceil(sqrt(o))
        R=-(N*x)
        baby=0*x
        baby_list=[baby]
        for i in range(N-1):
            baby+=x
            baby_list.append(baby)
        giant=y
        j=0
        while giant not in baby_list:
              giant+=R
              j+=1
        i=baby_list.index(giant)
        return ((i+j*N)%o,N+j+1)
    (answer,st)=baby_giant(P,Q)
    step_so+=answer
    step_BG+=st
    for set in range(3,21):
        def rho1(x,y):
            end=0
            while end==0:
                  a=randint(0,o-1)
```

```
                    b=randint(0,o-1)
                    list_S=[a*x+b*y]
                    list_Sab=[(a,b)]
                    list_m=[]
                    list_mab=[]
                    for i in range(set):
                        a=randint(0,o-1)
                        b=randint(0,o-1)
                        list_m.append(a*x+b*y)
                        list_mab.append((a,b))
                    i=0
                    end_2=0
                    while end_2==0:
                          S=list_S[i]
                          S_xmod=int(S[0])%set
                          new_S=S+list_m[S_xmod]
                          new_Sa=(list_Sab[i][0]+list_mab[S_xmod][0])%o
                          new_Sb=(list_Sab[i][1]+list_mab[S_xmod][1])%o
                          if new_S in list_S:
                             end_2=1
                             Sa=list_Sab[list_S.index(new_S)][0]
                             Sb=list_Sab[list_S.index(new_S)][1]
                             if gcd(Sb-new_Sb,o)==1:
                                end=1
                                return ((new_Sa-Sa)*inverse_mod(Sb-new_Sb,o)%o,len(list_S)
+1)
                          else:
                             list_S.append(new_S)
                             list_Sab.append((new_Sa,new_Sb))
                             i+=1
        st=rho1(P,Q)[1]
        step_R1[set]+=st
    for set in range(3,21):
        def rho2(x,y):
            end=0
            while end==0:
                  list_m=[]
                  list_mab=[]
                  for i in range(set):
                      a=randint(0,o-1)
                      b=randint(0,o-1)
                      list_m.append(a*x+b*y)
                      list_mab.append((a,b))
                  a=randint(0,o-1)
                  b=randint(0,o-1)
                  S=a*x+b*y
                  Sab=(a,b)
                  T=S
                  Tab=Sab
                  i=1
                  end_2=0
                  while end_2==0:
                        i+=1
                        S_xmod=int(S[0])%set
                        new_S=S+list_m[S_xmod]
                        new_Sa=(Sab[0]+list_mab[S_xmod][0])%o
                        new_Sb=(Sab[1]+list_mab[S_xmod][1])%o
```

```
                        T__xmod=int(T[0])%set
                        T_=T+list_m[T__xmod]
                        T_a=(Tab[0]+list_mab[T__xmod][0])%o
                        T_b=(Tab[1]+list_mab[T__xmod][1])%o
                        T_xmod=int(T_[0])%set
                        new_T=T_+list_m[T_xmod]
                        new_Ta=(T_a+list_mab[T_xmod][0])%o
                        new_Tb=(T_b+list_mab[T_xmod][1])%o
                        if new_S==new_T:
                            end_2=1
                            if gcd(new_Sb-new_Tb,o)==1:
                                end=1
                                return ((new_Ta-new_Sa)*inverse_mod(new_Sb-new_Tb,o)%o,i)
                        else:
                            S=new_S
                            Sab=(new_Sa,new_Sb)
                            T=new_T
                            Tab=(new_Ta,new_Tb)
        st=rho2(P,Q)[1]
        step_R2[set]+=st
step_BG=n(step_BG/repeat)
for i in range(3,21):
    step_R1[i]=n(step_R1[i]/repeat)
    step_R2[i]=n(step_R2[i]/repeat)
step_so=n(step_so/repeat)
print 期待値"BG",n(3*sqrt(o)/2)
print 期待値"Rho1",n(sqrt(pi*o/2)+1)
print 総当たり"",step_so
print "BG",step_BG
print "Rho1",step_R1
print "Rho2",step_R2
```

6.4 実験 2

総当たり, BSGS, Rho-I, Rho-Ⅱ で ECDLP の解読を行い, それぞれの計算時間を求めた. ただし, 1 つの $E/\mathbb{F}_p$ について ECDLP を 30 問解読して計算時間の平均を求め, それを各アルゴリズムが要した計算時間 t [ms] として定めた. 使用する楕円曲線の定義式は $y^2 = x^3 + 7$ とし, Rho-I, Rho-Ⅱ における直和分割の個数は $m = 3, 8, 20$ とした. (m の値は, [11]p918,[12]p66,[13]p1644 を参考に選択した.)

さらに, BSGS, Rho-I, Rho-Ⅱ については計算量対時間比 $t/\sqrt{n}$ を求め, 比較した.

使用したソースコードは以下の通り.

(例 : 定義式 $y^2 = x^3 + 7$, 標数 10 ビット)

ソースコード 2　ECDLP の作成

```
import random
a=0
b=7
bit=10
times=10
end=0
while end==0:
    p=random.sample(list(primes(2^(bit-1),2^bit)),1)[0]
```

```
    d=(-4*a^3-27*b^2)%p
    if d!=0:
       K=GF(p)
       E=EllipticCurve(K,[a%p,b%p])
       o=E.order()
       if is_prime(o):
          print 標数"p=",p
          print "y^2 = x^3+",a%p,"x+",b%p
          print "order=",o
          end=1
list_Squ=[]
for j in range(0,(p+1)/2):
    list_Squ.append(j^2%p)
i=0
while (i^3+a*i+b)%p not in list_Squ:
      i+=1
j=list_Squ.index((i^3+a*i+b)%p)
Point=E([i,j])
for l in range(10):
    P=randint(1,o-1)*Point
    Q=randint(1,o-1)*Point
    print "P=",P
    print "Q=",Q
```

(例：定義式 $y^2 = x^3 + 7$, 標数 547, $P = (315, 316)$, $Q = (15, 537)$)

ソースコード 3　定義

```
p=547
P_x=315
P_y=316
Q_x=15
Q_y=537
a=0
b=7
K=GF(p)
E=EllipticCurve(K,[a%p,b%p])
o=E.order()
P=E([P_x,P_y])
Q=E([Q_x,Q_y])
```

ソースコード 4　総当たり

```
def soatari(x,y):
    A=x
    i=1
    while A!=y:
          A=A+x
          i+=1
    return i
%time soatari(P,Q)
```

ソースコード 5　BSGS

```
def baby_giant(x,y):
    N=ceil(sqrt(o))
```

```
    R=-(N*x)
    baby=0*x
    baby_list=[baby]
    for i in range(N-1):
        baby+=x
        baby_list.append(baby)
    giant=y
    j=0
    while giant not in baby_list:
          giant+=R
          j+=1
    i=baby_list.index(giant)
    return (i+j*N)%o
%time  baby_giant(P,Q)
```

ソースコード 6　Rho-I(3)

```
import random
def rho(x,y):
    set=3
    end=0
    while end==0:
          a=randint(0,o-1)
          b=randint(0,o-1)
          list_S=[a*x+b*y]
          list_Sab=[(a,b)]
          list_m=[]
          list_mab=[]
          for i in range(set):
              a=randint(0,o-1)
              b=randint(0,o-1)
              list_m.append(a*x+b*y)
              list_mab.append((a,b))
          i=0
          end_2=0
          while end_2==0:
                S=list_S[i]
                S_xmod=int(S[0])%set
                new_S=S+list_m[S_xmod]
                new_Sa=(list_Sab[i][0]+list_mab[S_xmod][0])%o
                new_Sb=(list_Sab[i][1]+list_mab[S_xmod][1])%o
                if new_S in list_S:
                   end_2=1
                   Sa=list_Sab[list_S.index(new_S)][0]
                   Sb=list_Sab[list_S.index(new_S)][1]
                   if gcd(Sb-new_Sb,o)==1:
                      end=1
                      return (new_Sa-Sa)*inverse_mod(Sb-new_Sb,o)%o
                else:
                   list_S.append(new_S)
                   list_Sab.append((new_Sa,new_Sb))
                   i+=1
%time rho(P,Q)
```

ソースコード 7　Rho-II(3)

```
import random
```

```
def rho(x,y):
    set=3
    end=0
    while end==0:
          list_m=[]
          list_mab=[]
          for i in range(set):
              a=randint(0,o-1)
              b=randint(0,o-1)
              list_m.append(a*x+b*y)
              list_mab.append((a,b))
          a=randint(0,o-1)
          b=randint(0,o-1)
          S=a*x+b*y
          Sab=(a,b)
          T=S
          Tab=Sab
          end_2=0
          while end_2==0:
                S_xmod=int(S[0])%set
                new_S=S+list_m[S_xmod]
                new_Sa=(Sab[0]+list_mab[S_xmod][0])%o
                new_Sb=(Sab[1]+list_mab[S_xmod][1])%o
                T__xmod=int(T[0])%set
                T_=T+list_m[T__xmod]
                T_a=(Tab[0]+list_mab[T__xmod][0])%o
                T_b=(Tab[1]+list_mab[T__xmod][1])%o
                T_xmod=int(T_[0])%set
                new_T=T_+list_m[T_xmod]
                new_Ta=(T_a+list_mab[T_xmod][0])%o
                new_Tb=(T_b+list_mab[T_xmod][1])%o
                if new_S==new_T:
                   end_2=1
                   if gcd(new_Sb-new_Tb,o)==1:
                      end=1
                      return (new_Ta-new_Sa)*inverse_mod(new_Sb-new_Tb,o)%o
                else:
                   S=new_S
                   Sab=(new_Sa,new_Sb)
                   T=new_T
                   Tab=(new_Ta,new_Tb)
%time rho(P,Q)
```

7　実験結果

7.1　実験 1

実験結果は, 表 1〜9 の通りである. 表 2,3,5,6,8,9 については, 同じ楕円曲線における Rho-I(3)〜(50) と Rho-Ⅱ(3)〜(50) それぞれについて, ステップ数が最も多い 5 つを橙色, 最も少ない 5 つを黄色で着色した. 表 1,4,7 については, ステップ数の実測値と期待値の差が最も大きい 5 つを橙色, 最も小さい 5 つを水色にした.

まず, 各アルゴリズムのステップ数を比較する. 今回の実験から, ステップ数が最も少ないアルゴリズムは Rho-Ⅱ であることがわかった. ただし, m の値は十分大きくとる必要がある (表 2,3,5,6,8,9). 総当たりについては, 全ての曲線においてステップ数が著しく多いことが観察できた.

次に, Rho-I,Ⅱ における直和分割の個数 m とステップ数の関係を見てみると, Rho-I,Ⅱ 共に, m が小さいほどステップ数が多くなる傾向がみられた (表 2,3,5,6,8,9). 特に Rho-I については, 表 1 と表 2, 表 4 と表 5, 表 7 と表 8 を比較してわかる通り, 「ステップ数の実測値と期待値の差が最も大きい 5 つ」と「ステップ数が多い 5 つ」が一致する結果となった. また, 「ステップ数の実測値と期待値の差が最も小さい 5 つ」と「ステップ数が少ない 5 つ」は, 完全に重なったわけではないが, 分布の様子が似ていた. 3 つの曲線全てにおいて, 期待値から外れた 5 つは $m \leq 10$, 期待値に近い 5 つは $m \geq 16$ の範囲に分布していた (表 1,4,7).

表 1　総当たり,BSGS,Rho-I のステップ数 (期待値との比較)　$[y^2 = x^3 + 7]$

標数	p	547	3511	11839	47017	194119	881539	2744713
	bit	10	12	14	16	18	20	22
n		547	3433	12049	47353	194917	880981	2746417
sの期待値	BSGS	35.082	87.888	164.652	326.411	662.241	1407.909	2485.848
	Rho-I	30.313	74.434	138.574	273.730	554.331	1177.368	2078.032
s	総当たり	273.488	1686.259	6287.919	23409.155	95410.809	430588.574	1334742.700
	BSGS	35.920	88.071	167.671	325.885	658.364	1398.057	2463.541
	Rho-I(3)	47.045	123.323	240.938	503.749	1096.188	2400.395	4537.627
	Rho-I(4)	38.838	96.634	183.229	365.701	759.749	1580.652	2857.279
	Rho-I(5)	36.404	88.595	164.888	328.418	654.931	1398.207	2491.537
	Rho-I(6)	34.472	85.673	160.360	305.563	631.797	1331.963	2374.995
	Rho-I(7)	32.912	83.971	148.607	296.930	616.520	1289.028	2357.589
	Rho-I(8)	32.865	81.428	149.799	300.260	587.900	1284.035	2252.272
	Rho-I(9)	33.356	82.486	148.688	292.405	577.755	1251.140	2258.565
	Rho-I(10)	32.664	80.762	151.439	289.784	609.567	1249.031	2209.169
	Rho-I(11)	32.434	79.250	148.489	291.293	597.774	1241.638	2222.169
	Rho-I(12)	32.322	78.051	146.949	281.465	599.103	1235.939	2104.026
	Rho-I(13)	32.042	79.215	143.234	283.498	569.152	1210.706	2106.247
	Rho-I(14)	32.282	77.214	146.387	284.327	581.773	1211.617	2160.343
	Rho-I(15)	31.774	77.314	143.443	286.829	572.197	1235.438	2122.745
	Rho-I(16)	31.792	78.600	141.103	285.901	574.317	1193.242	2188.094
	Rho-I(17)	31.468	75.397	145.101	284.538	572.965	1211.300	2096.734
	Rho-I(18)	31.002	76.061	144.361	282.251	587.282	1245.227	2158.488
	Rho-I(19)	30.657	77.178	145.758	280.654	597.035	1211.766	2187.496
	Rho-I(20)	31.376	75.331	144.431	287.889	577.523	1205.988	2143.799
	Rho-I(21)	32.011	79.287	142.979	283.639	568.427	1201.866	2149.878
	Rho-I(22)	31.248	74.361	140.473	278.349	568.434	1229.791	2168.768
	Rho-I(23)	30.891	78.010	142.156	280.452	568.827	1226.291	2180.467
	Rho-I(24)	30.786	77.754	139.872	280.485	577.171	1176.654	2175.949
	Rho-I(25)	30.683	76.372	143.372	276.264	570.766	1202.800	2088.978
	Rho-I(26)	30.431	76.264	143.216	273.736	564.792	1168.765	2069.492
	Rho-I(27)	31.565	76.225	143.182	281.464	581.500	1206.772	2086.471
	Rho-I(28)	30.156	77.263	140.269	279.540	577.939	1193.366	2043.145
	Rho-I(29)	30.485	77.928	142.928	271.781	558.278	1187.583	2118.069
	Rho-I(30)	31.073	75.277	145.646	281.322	572.544	1218.928	2147.381
	Rho-I(31)	31.048	75.179	142.552	284.724	565.167	1207.605	2100.650
	Rho-I(32)	30.312	75.214	138.733	279.176	561.306	1234.199	2172.899
	Rho-I(33)	30.898	74.129	137.201	278.384	564.324	1183.022	2116.767
	Rho-I(34)	29.972	74.660	144.447	281.426	560.600	1189.205	2127.349
	Rho-I(35)	30.841	77.577	139.864	278.723	567.305	1187.285	2069.279
	Rho-I(36)	31.427	76.723	138.705	278.689	573.307	1196.072	2123.811
	Rho-I(37)	29.755	74.887	140.457	289.426	567.559	1190.085	2146.196
	Rho-I(38)	30.141	76.378	143.159	273.748	559.848	1203.061	2112.161
	Rho-I(39)	29.810	75.921	140.332	281.122	574.754	1167.207	2124.322
	Rho-I(40)	30.451	74.646	142.400	274.213	572.518	1186.992	2123.668
	Rho-I(41)	29.880	74.606	136.437	274.015	550.474	1165.311	2187.536
	Rho-I(42)	30.644	74.087	140.591	276.476	559.104	1184.640	2043.446
	Rho-I(43)	31.077	76.055	137.873	282.601	575.343	1183.772	2150.929
	Rho-I(44)	30.062	78.395	140.050	273.786	561.108	1205.835	2035.671
	Rho-I(45)	30.383	75.281	138.912	282.872	563.097	1180.467	2126.154
	Rho-I(46)	30.869	73.136	141.358	279.089	579.024	1178.257	2150.056
	Rho-I(47)	30.402	75.086	139.484	278.555	565.882	1213.603	2106.513
	Rho-I(48)	30.340	75.554	139.266	285.036	570.035	1209.485	2082.557
	Rho-I(49)	30.033	74.703	136.796	286.856	557.713	1185.253	2112.565
	Rho-I(50)	30.431	76.996	140.614	277.811	571.844	1196.336	2077.680

表 2　総当たり,BSGS,Rho-I のステップ数 (最多・最少)　$[y^2 = x^3 + 7]$

標数	p	547	3511	11839	47017	194119	881539	2744713
	bit	10	12	14	16	18	20	22
n		547	3433	12049	47353	194917	880981	2746417
sの期待値	BSGS	35.082	87.888	164.652	326.411	662.241	1407.909	2485.848
	Rho-I	30.313	74.434	138.574	273.730	554.331	1177.368	2078.032
s	総当たり	273.488	1686.259	6287.919	23409.155	95410.809	430588.574	1334742.700
	BSGS	35.920	88.071	167.671	325.885	658.364	1398.057	2463.541
	Rho-I(3)	47.045	123.323	240.938	503.749	1096.188	2400.395	4537.627
	Rho-I(4)	38.838	96.634	183.229	365.701	759.749	1580.652	2857.279
	Rho-I(5)	36.404	88.595	164.888	328.418	654.931	1398.207	2491.537
	Rho-I(6)	34.472	85.673	160.360	305.563	631.797	1331.963	2374.995
	Rho-I(7)	32.912	83.971	148.607	296.930	616.520	1289.028	2357.589
	Rho-I(8)	32.865	81.428	149.799	300.260	587.900	1284.035	2252.272
	Rho-I(9)	33.356	82.486	148.688	292.405	577.755	1251.140	2258.565
	Rho-I(10)	32.664	80.762	151.439	289.784	609.567	1249.031	2209.169
	Rho-I(11)	32.434	79.250	148.489	291.293	597.774	1241.638	2222.169
	Rho-I(12)	32.322	78.051	146.949	281.465	599.103	1235.939	2104.026
	Rho-I(13)	32.042	79.215	143.234	283.498	569.152	1210.706	2106.247
	Rho-I(14)	32.282	77.214	146.387	284.327	581.773	1211.617	2160.343
	Rho-I(15)	31.774	77.314	143.443	286.829	572.197	1235.438	2122.745
	Rho-I(16)	31.792	78.600	141.103	285.901	574.317	1193.242	2188.094
	Rho-I(17)	31.468	75.397	145.101	284.538	572.965	1211.300	2096.734
	Rho-I(18)	31.002	76.061	144.361	282.251	587.282	1245.227	2158.488
	Rho-I(19)	30.657	77.178	145.758	280.654	597.035	1211.766	2187.496
	Rho-I(20)	31.376	75.331	144.431	287.889	577.523	1205.988	2143.799
	Rho-I(21)	32.011	79.287	142.979	283.639	568.427	1201.866	2149.878
	Rho-I(22)	31.248	74.361	140.473	278.349	568.434	1229.791	2168.768
	Rho-I(23)	30.891	78.010	142.156	280.452	568.827	1226.291	2180.467
	Rho-I(24)	30.786	77.754	139.872	280.485	577.171	1176.654	2175.949
	Rho-I(25)	30.683	76.372	143.372	276.264	570.766	1202.800	2088.978
	Rho-I(26)	30.431	76.264	143.216	273.736	564.792	1168.765	2069.492
	Rho-I(27)	31.565	76.225	143.182	281.464	581.500	1206.772	2086.471
	Rho-I(28)	30.156	77.263	140.269	279.540	577.939	1193.366	2043.145
	Rho-I(29)	30.485	77.928	142.928	271.781	558.278	1187.583	2118.069
	Rho-I(30)	31.073	75.277	145.646	281.322	572.544	1218.928	2147.381
	Rho-I(31)	31.048	75.179	142.552	284.724	565.167	1207.605	2100.650
	Rho-I(32)	30.312	75.214	138.733	279.176	561.306	1234.199	2172.899
	Rho-I(33)	30.898	74.129	137.201	278.384	564.324	1183.022	2116.767
	Rho-I(34)	29.972	74.660	144.447	281.426	560.600	1189.205	2127.349
	Rho-I(35)	30.841	77.577	139.864	278.723	567.305	1187.285	2069.279
	Rho-I(36)	31.427	76.723	138.705	278.689	573.307	1196.072	2123.811
	Rho-I(37)	29.755	74.887	140.457	289.426	567.559	1190.085	2146.196
	Rho-I(38)	30.141	76.378	143.159	273.748	559.848	1203.061	2112.161
	Rho-I(39)	29.810	75.921	140.332	281.122	574.754	1167.207	2124.322
	Rho-I(40)	30.451	74.646	142.400	274.213	572.518	1186.992	2123.668
	Rho-I(41)	29.880	74.606	136.437	274.015	550.474	1165.311	2187.536
	Rho-I(42)	30.644	74.087	140.591	276.476	559.104	1184.640	2043.446
	Rho-I(43)	31.077	76.055	137.873	282.601	575.343	1183.772	2150.929
	Rho-I(44)	30.062	78.395	140.050	273.786	561.108	1205.835	2035.671
	Rho-I(45)	30.383	75.281	138.912	282.872	563.097	1180.467	2126.154
	Rho-I(46)	30.869	73.136	141.358	279.089	579.024	1178.257	2150.056
	Rho-I(47)	30.402	75.086	139.484	278.555	565.882	1213.603	2106.513
	Rho-I(48)	30.340	75.554	139.266	285.036	570.035	1209.485	2082.557
	Rho-I(49)	30.033	74.703	136.796	286.856	557.713	1185.253	2112.565
	Rho-I(50)	30.431	76.996	140.614	277.811	571.844	1196.336	2077.680

表 3　Rho-II のステップ数（最多・最少）　$[y^2 = x^3 + 7]$

標数	p	547	3511	11839	47017	194119	881539	2744713
	bit	10	12	14	16	18	20	22
n		547	3433	12049	47353	194917	880981	2746417
s	Rho-II(3)	38.543	101.720	202.827	419.245	862.866	1944.628	3633.208
	Rho-II(4)	31.422	80.885	152.934	288.814	619.227	1314.354	2284.849
	Rho-II(5)	28.768	72.280	137.001	273.093	552.934	1135.385	2063.606
	Rho-II(6)	28.013	71.076	131.661	259.110	510.049	1100.332	1943.237
	Rho-II(7)	27.279	67.025	126.213	245.395	507.166	1079.850	1915.664
	Rho-II(8)	26.671	67.607	123.147	242.095	493.634	1052.115	1874.270
	Rho-II(9)	25.964	65.435	126.227	238.289	489.704	1040.131	1786.262
	Rho-II(10)	26.066	64.680	119.062	242.265	498.202	1057.490	1805.020
	Rho-II(11)	25.765	62.422	119.075	239.343	476.075	1014.097	1797.229
	Rho-II(12)	25.532	63.179	121.866	233.224	479.494	1006.540	1813.831
	Rho-II(13)	25.709	62.848	117.872	233.890	488.115	1020.623	1711.149
	Rho-II(14)	25.514	62.227	115.259	228.938	476.430	1026.264	1763.085
	Rho-II(15)	25.474	63.484	118.402	228.952	481.980	1015.250	1819.964
	Rho-II(16)	25.628	63.541	118.199	234.543	474.490	997.690	1764.495
	Rho-II(17)	25.309	61.686	120.251	227.845	462.379	997.693	1740.113
	Rho-II(18)	25.335	64.250	118.473	232.706	459.200	1002.993	1793.827
	Rho-II(19)	24.579	60.789	118.357	229.095	469.669	959.915	1703.603
	Rho-II(20)	25.786	64.471	118.096	236.160	478.878	1034.599	1798.025
	Rho-II(21)	24.802	61.971	114.844	234.588	455.216	1007.128	1734.122
	Rho-II(22)	24.885	63.447	118.482	229.381	466.510	1005.399	1726.162
	Rho-II(23)	25.272	60.793	116.699	233.195	462.452	981.816	1761.026
	Rho-II(24)	25.013	64.166	116.607	235.029	468.256	1004.890	1771.809
	Rho-II(25)	24.995	62.431	114.597	228.457	475.993	972.390	1798.504
	Rho-II(26)	25.504	61.814	116.521	231.583	479.444	992.457	1758.936
	Rho-II(27)	24.647	62.212	116.753	223.745	460.284	986.541	1734.951
	Rho-II(28)	24.506	64.987	116.941	228.250	465.614	962.495	1761.806
	Rho-II(29)	24.841	61.276	116.340	225.956	472.212	988.065	1680.871
	Rho-II(30)	25.387	63.551	114.519	243.796	461.715	1002.624	1736.379
	Rho-II(31)	25.254	62.397	116.708	228.276	476.216	962.339	1700.236
	Rho-II(32)	24.373	62.009	117.924	225.221	477.589	984.062	1736.772
	Rho-II(33)	25.964	63.975	115.775	230.646	463.536	942.920	1774.100
	Rho-II(34)	25.348	62.630	115.100	231.178	464.809	991.535	1723.691
	Rho-II(35)	25.245	62.277	116.555	228.275	471.504	979.302	1707.627
	Rho-II(36)	25.256	63.058	115.008	230.363	458.904	970.873	1754.686
	Rho-II(37)	25.415	59.878	116.319	217.852	479.563	962.338	1681.031
	Rho-II(38)	25.410	62.462	115.035	230.946	448.164	1000.384	1766.243
	Rho-II(39)	24.544	60.714	112.589	234.781	458.761	981.923	1749.607
	Rho-II(40)	24.579	61.597	114.323	227.888	453.608	1001.853	1700.406
	Rho-II(41)	24.953	60.256	114.417	229.711	466.313	918.297	1736.455
	Rho-II(42)	24.964	61.229	115.951	233.232	457.266	976.425	1714.383
	Rho-II(43)	25.642	60.152	115.016	223.957	460.035	964.876	1733.855
	Rho-II(44)	25.360	63.836	116.808	225.090	458.364	973.530	1776.841
	Rho-II(45)	25.333	62.003	115.704	230.258	455.657	1004.581	1676.743
	Rho-II(46)	24.549	62.229	118.890	227.277	467.338	982.590	1699.967
	Rho-II(47)	24.744	61.891	117.084	226.497	470.954	994.089	1700.001
	Rho-II(48)	25.471	62.427	115.801	228.336	455.998	941.596	1702.607
	Rho-II(49)	24.338	60.621	113.799	225.601	456.488	975.594	1705.841
	Rho-II(50)	24.811	59.573	116.956	232.146	464.016	976.814	1732.133

表 4　総当たり,BSGS,Rho-I のステップ数 (期待値との比較)　$[y^2 = x^3 + 2]$

標数	p	661	3931	16069	62131	132637	934069	2501677
	bit	10	12	14	16	18	20	22
n		613	4021	15823	62533	133183	932257	2499403
sの期待値	BSGS	37.138	95.117	188.684	375.099	547.414	1448.302	2371.425
	Rho-I	32.031	80.474	158.654	314.411	458.388	1211.118	1982.427
s	総当たり	301.289	2010.163	7896.408	31107.488	66359.736	471097.504	1243815.967
	BSGS	37.578	95.902	189.170	375.422	547.289	1454.189	2368.234
	Rho-I(3)	48.052	141.061	283.131	606.427	884.645	2501.374	4240.117
	Rho-I(4)	41.480	106.641	211.938	424.652	632.103	1619.058	2680.050
	Rho-I(5)	37.413	96.451	196.361	370.985	543.481	1485.139	2361.360
	Rho-I(6)	37.313	89.254	181.150	352.958	511.993	1374.828	2210.926
	Rho-I(7)	36.437	90.221	179.571	347.486	527.191	1335.808	2158.274
	Rho-I(8)	35.250	89.796	169.693	353.039	497.927	1309.520	2131.339
	Rho-I(9)	34.823	88.359	168.976	341.810	496.750	1275.869	2192.258
	Rho-I(10)	34.190	85.988	168.650	335.768	480.327	1284.850	2080.843
	Rho-I(11)	34.456	86.618	166.784	325.905	484.271	1278.225	2132.727
	Rho-I(12)	33.573	86.423	167.647	328.188	477.318	1284.883	2053.946
	Rho-I(13)	33.296	85.686	166.956	328.158	485.364	1239.724	2047.378
	Rho-I(14)	33.102	82.501	166.705	321.849	474.967	1255.747	2107.695
	Rho-I(15)	34.514	82.904	161.265	318.730	478.049	1259.397	2055.111
	Rho-I(16)	33.073	82.013	165.021	326.973	458.838	1279.829	2025.273
	Rho-I(17)	33.599	84.456	162.513	327.355	479.772	1227.393	1980.370
	Rho-I(18)	32.869	84.728	167.147	329.098	479.046	1263.161	2033.716
	Rho-I(19)	33.539	85.652	160.880	325.043	460.765	1252.536	2006.770
	Rho-I(20)	33.209	83.487	169.137	321.933	484.640	1265.744	2139.078
	Rho-I(21)	32.408	81.824	162.343	321.042	476.252	1252.164	2021.251
	Rho-I(22)	31.663	83.747	166.038	328.195	472.886	1213.426	2016.742
	Rho-I(23)	32.318	83.184	159.568	310.828	466.600	1257.052	1988.407
	Rho-I(24)	33.343	81.927	162.480	320.524	483.846	1217.181	2076.085
	Rho-I(25)	32.903	83.576	159.235	325.985	473.366	1216.875	2076.316
	Rho-I(26)	33.174	78.837	158.227	313.800	464.759	1244.725	2036.244
	Rho-I(27)	32.658	83.231	163.825	324.069	466.050	1248.187	2006.501
	Rho-I(28)	33.226	82.046	160.369	316.951	468.873	1208.232	2019.680
	Rho-I(29)	32.567	82.555	162.493	317.982	467.102	1215.229	1985.000
	Rho-I(30)	31.701	82.668	159.160	321.325	451.844	1244.192	1979.836
	Rho-I(31)	32.263	81.910	159.745	326.118	481.464	1244.794	2040.698
	Rho-I(32)	33.263	81.602	168.411	324.665	462.606	1192.840	2007.089
	Rho-I(33)	32.524	83.875	164.413	324.683	463.637	1238.155	2050.714
	Rho-I(34)	33.084	83.498	160.767	315.134	459.293	1263.233	2009.464
	Rho-I(35)	32.421	80.584	164.217	315.736	456.006	1236.962	2083.962
	Rho-I(36)	32.424	79.501	160.356	319.479	467.043	1223.243	1983.553
	Rho-I(37)	31.883	82.843	157.847	318.370	479.525	1234.292	1998.143
	Rho-I(38)	31.667	80.242	156.957	321.498	470.343	1233.417	2053.398
	Rho-I(39)	32.638	82.258	158.487	317.837	462.043	1251.486	1974.916
	Rho-I(40)	31.113	82.078	160.419	307.836	457.545	1251.963	1959.870
	Rho-I(41)	32.006	82.551	164.931	315.243	467.739	1240.864	1973.108
	Rho-I(42)	32.161	83.338	160.196	320.870	481.871	1216.319	1956.317
	Rho-I(43)	32.140	81.165	162.414	328.663	458.601	1249.448	2031.993
	Rho-I(44)	32.535	79.409	160.837	317.918	472.409	1230.326	1971.764
	Rho-I(45)	32.142	79.507	160.530	318.082	465.409	1195.501	2029.610
	Rho-I(46)	32.465	79.325	159.352	309.415	467.445	1205.336	2038.584
	Rho-I(47)	32.799	81.546	159.939	327.588	465.578	1233.313	2013.180
	Rho-I(48)	32.481	82.746	161.293	313.805	468.558	1196.246	2000.910
	Rho-I(49)	32.192	81.751	163.782	315.197	448.072	1242.371	2039.131
	Rho-I(50)	33.189	82.508	161.155	321.697	469.816	1220.089	2006.702

表 5　総当たり,BSGS,Rho-I のステップ数 (最多・最少)　$[y^2 = x^3 + 2]$

標数	p	661	3931	16069	62131	132637	934069	2501677
	bit	10	12	14	16	18	20	22
n		613	4021	15823	62533	133183	932257	2499403
sの期待値	BSGS	37.138	95.117	188.684	375.099	547.414	1448.302	2371.425
	Rho-I	32.031	80.474	158.654	314.411	458.388	1211.118	1982.427
s	総当たり	301.289	2010.163	7896.408	31107.488	66359.736	471097.504	1243815.967
	BSGS	37.578	95.902	189.170	375.422	547.289	1454.189	2368.234
	Rho-I(3)	48.052	141.061	283.131	606.427	884.645	2501.374	4240.117
	Rho-I(4)	41.480	106.641	211.938	424.652	632.103	1619.058	2680.050
	Rho-I(5)	37.413	96.451	196.361	370.985	543.481	1485.139	2361.360
	Rho-I(6)	37.313	89.254	181.150	352.958	511.993	1374.828	2210.926
	Rho-I(7)	36.437	90.221	179.571	347.486	527.191	1335.808	2158.274
	Rho-I(8)	35.250	89.796	169.693	353.039	497.927	1309.520	2131.339
	Rho-I(9)	34.823	88.359	168.976	341.810	496.750	1275.869	2192.258
	Rho-I(10)	34.190	85.988	168.650	335.768	480.327	1284.850	2080.843
	Rho-I(11)	34.456	86.618	166.784	325.905	484.271	1278.225	2132.727
	Rho-I(12)	33.573	86.423	167.647	328.188	477.318	1284.883	2053.946
	Rho-I(13)	33.296	85.686	166.956	328.158	485.364	1239.724	2047.378
	Rho-I(14)	33.102	82.501	166.705	321.849	474.967	1255.747	2107.695
	Rho-I(15)	34.514	82.904	161.265	318.730	478.049	1259.397	2055.111
	Rho-I(16)	33.073	82.013	165.021	326.973	458.838	1279.829	2025.273
	Rho-I(17)	33.599	84.456	162.513	327.355	479.772	1227.393	1980.370
	Rho-I(18)	32.869	84.728	167.147	329.098	479.046	1263.161	2033.716
	Rho-I(19)	33.539	85.652	160.880	325.043	460.765	1252.536	2006.770
	Rho-I(20)	33.209	83.487	169.137	321.933	484.640	1265.744	2139.078
	Rho-I(21)	32.408	81.824	162.343	321.042	476.252	1252.164	2021.251
	Rho-I(22)	31.663	83.747	166.038	328.195	472.886	1213.426	2016.742
	Rho-I(23)	32.318	83.184	159.568	310.828	466.600	1257.052	1988.407
	Rho-I(24)	33.343	81.927	162.480	320.524	483.846	1217.181	2076.085
	Rho-I(25)	32.903	83.576	159.235	325.985	473.366	1216.875	2076.316
	Rho-I(26)	33.174	78.837	158.227	313.800	464.759	1244.725	2036.244
	Rho-I(27)	32.658	83.231	163.825	324.069	466.050	1248.187	2006.501
	Rho-I(28)	33.226	82.046	160.369	316.951	468.873	1208.232	2019.680
	Rho-I(29)	32.567	82.555	162.493	317.982	467.102	1215.229	1985.000
	Rho-I(30)	31.701	82.668	159.160	321.325	451.844	1244.192	1979.836
	Rho-I(31)	32.263	81.910	159.745	326.118	481.464	1244.794	2040.698
	Rho-I(32)	33.263	81.602	168.411	324.665	462.606	1192.840	2007.089
	Rho-I(33)	32.524	83.875	164.413	324.683	463.637	1238.155	2050.714
	Rho-I(34)	33.084	83.498	160.767	315.134	459.293	1263.233	2009.464
	Rho-I(35)	32.421	80.584	164.217	315.736	456.006	1236.962	2083.962
	Rho-I(36)	32.424	79.501	160.356	319.479	467.043	1223.243	1983.553
	Rho-I(37)	31.883	82.843	157.847	318.370	479.525	1234.292	1998.143
	Rho-I(38)	31.667	80.242	156.957	321.498	470.343	1233.417	2053.398
	Rho-I(39)	32.638	82.258	158.487	317.837	462.043	1251.486	1974.916
	Rho-I(40)	31.113	82.078	160.419	307.836	457.545	1251.963	1959.870
	Rho-I(41)	32.006	82.551	164.931	315.243	467.739	1240.864	1973.108
	Rho-I(42)	32.161	83.338	160.196	320.870	481.871	1216.319	1956.317
	Rho-I(43)	32.140	81.165	162.414	328.663	458.601	1249.448	2031.993
	Rho-I(44)	32.535	79.409	160.837	317.918	472.409	1230.326	1971.764
	Rho-I(45)	32.142	79.507	160.530	318.082	465.409	1195.501	2029.610
	Rho-I(46)	32.465	79.325	159.352	309.415	467.445	1205.336	2038.584
	Rho-I(47)	32.799	81.546	159.939	327.588	465.578	1233.313	2013.180
	Rho-I(48)	32.481	82.746	161.293	313.805	468.558	1196.246	2000.910
	Rho-I(49)	32.192	81.751	163.782	315.197	448.072	1242.371	2039.131
	Rho-I(50)	33.189	82.508	161.155	321.697	469.816	1220.089	2006.702

表 6　Rho-II のステップ数（最多・最少）　$[y^2 = x^3 + 2]$

標数	p	661	3931	16069	62131	132637	934069	2501677
	bit	10	12	14	16	18	20	22
n		613	4021	15823	62533	133183	932257	2499403
s	Rho-II(3)	39.682	108.503	231.142	499.903	692.864	1996.110	3374.741
	Rho-II(4)	32.900	86.439	177.411	349.847	501.287	1334.987	2163.175
	Rho-II(5)	31.629	77.230	155.018	307.629	443.495	1186.168	1930.552
	Rho-II(6)	29.737	76.331	146.515	294.178	429.938	1107.233	1877.171
	Rho-II(7)	28.374	74.016	142.335	284.273	417.285	1068.667	1795.779
	Rho-II(8)	27.671	69.176	143.611	280.057	419.880	1115.076	1846.655
	Rho-II(9)	27.991	72.299	140.366	274.343	399.690	1066.194	1774.357
	Rho-II(10)	27.349	71.449	136.212	286.902	407.456	1068.711	1770.761
	Rho-II(11)	28.156	69.552	138.410	266.403	392.351	1047.667	1768.180
	Rho-II(12)	26.726	69.904	137.226	275.568	396.678	1039.137	1722.827
	Rho-II(13)	26.565	70.029	136.259	263.961	397.044	1068.595	1662.940
	Rho-II(14)	27.690	68.122	139.960	265.493	391.534	1062.856	1713.621
	Rho-II(15)	27.825	68.784	131.281	266.489	383.398	1054.280	1667.911
	Rho-II(16)	28.052	70.247	140.079	265.465	389.674	1048.084	1707.259
	Rho-II(17)	26.068	68.326	137.891	259.549	397.124	1019.093	1659.176
	Rho-II(18)	27.266	70.093	132.175	270.303	386.124	1038.791	1700.609
	Rho-II(19)	27.037	67.458	131.584	271.681	381.895	1029.479	1671.724
	Rho-II(20)	26.482	68.491	130.910	262.401	379.476	1039.301	1682.293
	Rho-II(21)	26.622	67.184	136.737	270.818	388.802	974.096	1685.694
	Rho-II(22)	27.042	67.213	136.993	269.782	375.831	1002.477	1699.368
	Rho-II(23)	26.912	67.647	135.653	261.303	375.369	1030.497	1691.385
	Rho-II(24)	26.570	70.516	135.004	267.144	378.641	1020.832	1571.155
	Rho-II(25)	26.674	68.822	132.831	264.107	387.332	1015.736	1692.225
	Rho-II(26)	26.573	67.956	134.667	273.180	384.233	1027.738	1638.198
	Rho-II(27)	26.025	65.826	132.057	261.313	379.130	1035.924	1657.268
	Rho-II(28)	26.727	66.400	129.909	262.739	382.464	997.474	1709.575
	Rho-II(29)	27.632	66.676	131.152	258.014	369.228	998.146	1662.189
	Rho-II(30)	26.571	66.381	133.822	257.985	378.241	1013.922	1666.535
	Rho-II(31)	26.927	66.347	134.330	262.308	383.116	1020.906	1696.268
	Rho-II(32)	26.523	65.916	134.109	260.564	371.035	1024.658	1649.046
	Rho-II(33)	26.290	67.233	129.026	256.447	377.665	996.925	1657.870
	Rho-II(34)	27.565	68.205	131.171	261.916	379.932	1023.667	1614.698
	Rho-II(35)	26.747	67.696	132.230	256.769	385.858	1034.531	1655.937
	Rho-II(36)	27.143	68.684	132.408	258.060	388.971	1012.020	1670.724
	Rho-II(37)	25.997	67.225	135.427	264.491	388.991	1043.675	1642.424
	Rho-II(38)	26.091	66.865	133.802	265.183	381.433	1021.637	1575.163
	Rho-II(39)	26.182	67.224	133.224	263.377	388.477	990.291	1644.159
	Rho-II(40)	25.079	67.366	128.669	264.061	384.477	1017.707	1657.332
	Rho-II(41)	26.017	65.707	130.497	265.516	379.709	1021.047	1656.764
	Rho-II(42)	26.650	67.884	131.391	263.471	375.637	1021.435	1696.716
	Rho-II(43)	25.925	67.893	128.447	267.457	381.372	1011.294	1695.794
	Rho-II(44)	26.473	67.312	126.785	260.676	381.167	1033.313	1656.980
	Rho-II(45)	25.862	66.622	129.057	265.786	384.887	1029.069	1607.659
	Rho-II(46)	27.251	65.651	131.164	262.935	374.812	1022.499	1644.962
	Rho-II(47)	26.519	67.374	129.218	265.532	373.934	982.858	1650.350
	Rho-II(48)	26.954	69.160	135.111	259.419	379.937	980.228	1641.721
	Rho-II(49)	26.200	65.299	131.476	265.961	378.527	998.873	1663.350
	Rho-II(50)	26.670	65.500	132.266	275.456	385.811	1008.146	1672.006

表 7　総当たり,BSGS,Rho-I のステップ数 (期待値との比較)　$[y^2 = x^3 - 3x + C_N]$

標数	p	859	4093	15401	56591	194017	723451	2212699
	bit	10	12	14	16	18	20	22
n		827	4211	15569	56611	193573	723739	2210389
sの期待値	BSGS	43.136	97.338	187.164	356.896	659.954	1276.093	2230.107
	Rho-I	37.042	82.330	157.383	299.202	552.420	1067.230	1864.349
s	総当たり	416.099	2132.160	7594.123	28198.783	97095.602	360417.516	1137680.133
	BSGS	43.874	98.322	186.267	356.976	661.156	1275.015	2252.576
	Rho-I(3)	57.831	140.878	283.921	562.401	1076.484	2201.571	3881.094
	Rho-I(4)	47.343	110.232	212.774	403.528	753.864	1418.594	2561.008
	Rho-I(5)	44.418	95.917	189.819	359.714	678.387	1294.181	2250.581
	Rho-I(6)	42.826	96.400	182.947	336.236	630.447	1212.737	2113.112
	Rho-I(7)	42.296	90.604	176.536	337.880	603.101	1180.527	2035.927
	Rho-I(8)	40.704	87.884	171.035	328.680	610.237	1145.154	2006.762
	Rho-I(9)	39.846	86.095	166.814	314.197	596.415	1153.446	1989.345
	Rho-I(10)	40.101	90.720	171.051	319.531	582.179	1131.807	1924.617
	Rho-I(11)	39.284	89.112	167.057	315.557	576.469	1118.124	1948.676
	Rho-I(12)	38.469	89.013	167.434	310.102	575.423	1131.793	1951.605
	Rho-I(13)	39.434	85.199	165.442	318.127	583.321	1097.831	1952.788
	Rho-I(14)	38.446	86.939	161.609	311.944	587.462	1097.183	1944.179
	Rho-I(15)	38.985	86.973	164.584	315.310	562.415	1115.844	1921.763
	Rho-I(16)	38.604	84.562	167.156	315.135	567.123	1133.376	1946.637
	Rho-I(17)	37.374	84.135	163.485	306.347	569.874	1107.670	1907.545
	Rho-I(18)	37.970	84.892	161.732	313.033	579.215	1085.617	1922.380
	Rho-I(19)	39.154	85.563	161.367	305.880	578.943	1097.267	1890.089
	Rho-I(20)	38.073	83.899	161.671	308.031	551.191	1108.320	1900.527
	Rho-I(21)	39.001	81.625	161.420	309.410	564.426	1095.544	1934.275
	Rho-I(22)	38.461	84.655	162.525	315.344	570.316	1108.021	1899.180
	Rho-I(23)	38.347	82.511	159.923	301.356	572.412	1096.065	1849.801
	Rho-I(24)	37.383	84.716	165.870	296.796	575.867	1079.406	1924.254
	Rho-I(25)	38.570	83.617	161.099	295.948	569.253	1095.130	1953.781
	Rho-I(26)	37.624	85.863	158.968	309.138	559.011	1094.566	1897.056
	Rho-I(27)	37.949	84.333	159.174	311.999	571.240	1094.602	1901.492
	Rho-I(28)	39.020	83.700	158.624	310.791	576.862	1074.389	1838.504
	Rho-I(29)	37.621	84.856	159.251	303.660	565.784	1083.000	1871.699
	Rho-I(30)	37.358	85.202	161.197	307.923	548.271	1062.063	1943.369
	Rho-I(31)	37.376	86.196	161.419	308.640	559.280	1087.953	1878.459
	Rho-I(32)	37.383	84.311	159.230	308.378	574.413	1086.245	1954.420
	Rho-I(33)	37.627	84.297	166.667	304.393	569.254	1064.087	1957.448
	Rho-I(34)	37.835	84.153	159.403	308.990	565.156	1094.335	1911.870
	Rho-I(35)	38.361	83.476	159.962	309.061	572.327	1114.729	1879.435
	Rho-I(36)	36.343	83.489	163.178	302.275	562.308	1070.897	1897.394
	Rho-I(37)	36.453	83.172	160.744	298.371	549.360	1096.159	1864.398
	Rho-I(38)	37.565	87.015	163.845	305.724	553.712	1094.036	1946.227
	Rho-I(39)	37.000	83.877	162.797	299.216	548.834	1062.189	1910.549
	Rho-I(40)	38.388	84.534	159.702	306.044	569.588	1084.456	1848.535
	Rho-I(41)	36.559	83.867	157.208	297.951	562.631	1105.594	1895.049
	Rho-I(42)	37.853	85.499	155.639	301.242	557.197	1093.126	1837.711
	Rho-I(43)	37.403	83.278	160.130	303.690	550.932	1055.673	1917.294
	Rho-I(44)	37.463	83.045	166.369	299.137	565.684	1075.763	1855.560
	Rho-I(45)	36.985	82.779	161.757	303.475	558.950	1067.510	1894.819
	Rho-I(46)	36.491	82.607	162.214	300.621	559.745	1102.775	1910.333
	Rho-I(47)	37.667	85.105	159.548	303.041	573.661	1079.805	1884.889
	Rho-I(48)	37.440	85.276	156.620	300.418	569.362	1085.539	1883.373
	Rho-I(49)	37.346	83.160	158.798	296.310	547.276	1114.174	1914.066
	Rho-I(50)	36.487	82.548	159.601	303.490	557.634	1060.803	1851.916

表 8　総当たり,BSGS,Rho-I のステップ数 (最多・最少)　$[y^2 = x^3 - 3x + C_N]$

標数	p	859	4093	15401	56591	194017	723451	2212699
	bit	10	12	14	16	18	20	22
n		827	4211	15569	56611	193573	723739	2210389
sの期待値	BSGS	43.136	97.338	187.164	356.896	659.954	1276.093	2230.107
	Rho-I	37.042	82.330	157.383	299.202	552.420	1067.230	1864.349
s	総当たり	416.099	2132.160	7594.123	28198.783	97095.602	360417.516	1137680.133
	BSGS	43.874	98.322	186.267	356.976	661.156	1275.015	2252.576
	Rho-I(3)	57.831	140.878	283.921	562.401	1076.484	2201.571	3881.094
	Rho-I(4)	47.343	110.232	212.774	403.528	753.864	1418.594	2561.008
	Rho-I(5)	44.418	95.917	189.819	359.714	678.387	1294.181	2250.581
	Rho-I(6)	42.826	96.400	182.947	336.236	630.447	1212.737	2113.112
	Rho-I(7)	42.296	90.604	176.536	337.880	603.101	1180.527	2035.927
	Rho-I(8)	40.704	87.884	171.035	328.680	610.237	1145.154	2006.762
	Rho-I(9)	39.846	86.095	166.814	314.197	596.415	1153.446	1989.345
	Rho-I(10)	40.101	90.720	171.051	319.531	582.179	1131.807	1924.617
	Rho-I(11)	39.284	89.112	167.057	315.557	576.469	1118.124	1948.676
	Rho-I(12)	38.469	89.013	167.434	310.102	575.423	1131.793	1951.605
	Rho-I(13)	39.434	85.199	165.442	318.127	583.321	1097.831	1952.788
	Rho-I(14)	38.446	86.939	161.609	311.944	587.462	1097.183	1944.179
	Rho-I(15)	38.985	86.973	164.584	315.310	562.415	1115.844	1921.763
	Rho-I(16)	38.604	84.562	167.156	315.135	567.123	1133.376	1946.637
	Rho-I(17)	37.374	84.135	163.485	306.347	569.874	1107.670	1907.545
	Rho-I(18)	37.970	84.892	161.732	313.033	579.215	1085.617	1922.380
	Rho-I(19)	39.154	85.563	161.367	305.880	578.943	1097.267	1890.089
	Rho-I(20)	38.073	83.899	161.671	308.031	551.191	1108.320	1900.527
	Rho-I(21)	39.001	81.625	161.420	309.410	564.426	1095.544	1934.275
	Rho-I(22)	38.461	84.655	162.525	315.344	570.316	1108.021	1899.180
	Rho-I(23)	38.347	82.511	159.923	301.356	572.412	1096.065	1849.801
	Rho-I(24)	37.383	84.716	165.870	296.796	575.867	1079.406	1924.254
	Rho-I(25)	38.570	83.617	161.099	295.948	569.253	1095.130	1953.781
	Rho-I(26)	37.624	85.863	158.968	309.138	559.011	1094.566	1897.056
	Rho-I(27)	37.949	84.333	159.174	311.999	571.240	1094.602	1901.492
	Rho-I(28)	39.020	83.700	158.624	310.791	576.862	1074.389	1838.504
	Rho-I(29)	37.621	84.856	159.251	303.660	565.784	1083.000	1871.699
	Rho-I(30)	37.358	85.202	161.197	307.923	548.271	1062.063	1943.369
	Rho-I(31)	37.376	86.196	161.419	308.640	559.280	1087.953	1878.459
	Rho-I(32)	37.383	84.311	159.230	308.378	574.413	1086.245	1954.420
	Rho-I(33)	37.627	84.297	166.667	304.393	569.254	1064.087	1957.448
	Rho-I(34)	37.835	84.153	159.403	308.990	565.156	1094.335	1911.870
	Rho-I(35)	38.361	83.476	159.962	309.061	572.327	1114.729	1879.435
	Rho-I(36)	36.343	83.489	163.178	302.275	562.308	1070.897	1897.394
	Rho-I(37)	36.453	83.172	160.744	298.371	549.360	1096.159	1864.398
	Rho-I(38)	37.565	87.015	163.845	305.724	553.712	1094.036	1946.227
	Rho-I(39)	37.000	83.877	162.797	299.216	548.834	1062.189	1910.549
	Rho-I(40)	38.388	84.534	159.702	306.044	569.588	1084.456	1848.535
	Rho-I(41)	36.559	83.867	157.208	297.951	562.631	1105.594	1895.049
	Rho-I(42)	37.853	85.499	155.639	301.242	557.197	1093.126	1837.711
	Rho-I(43)	37.403	83.278	160.130	303.690	550.932	1055.673	1917.294
	Rho-I(44)	37.463	83.045	166.369	299.137	565.684	1075.763	1855.560
	Rho-I(45)	36.985	82.779	161.757	303.475	558.950	1067.510	1894.819
	Rho-I(46)	36.491	82.607	162.214	300.621	559.745	1102.775	1910.333
	Rho-I(47)	37.667	85.105	159.548	303.041	573.661	1079.805	1884.889
	Rho-I(48)	37.440	85.276	156.620	300.418	569.362	1085.539	1883.373
	Rho-I(49)	37.346	83.160	158.798	296.310	547.276	1114.174	1914.066
	Rho-I(50)	36.487	82.548	159.601	303.490	557.634	1060.803	1851.916

表 9　Rho-II のステップ数 (最多・最少)　$[y^2 = x^3 - 3x + C_N]$

標数	p	859	4093	15401	56591	194017	723451	2212699
	bit	10	12	14	16	18	20	22
n		827	4211	15569	56611	193573	723739	2210389
s	Rho-II(3)	47.536	112.365	228.371	462.809	850.949	1735.268	3127.699
	Rho-II(4)	38.005	87.542	171.127	326.787	605.390	1148.853	2029.922
	Rho-II(5)	37.284	80.381	155.227	286.242	521.671	1079.452	1788.572
	Rho-II(6)	35.225	76.837	150.324	281.101	500.870	1001.910	1698.750
	Rho-II(7)	33.157	74.399	141.072	267.863	513.508	974.055	1692.770
	Rho-II(8)	34.079	71.654	140.957	271.338	504.726	941.973	1668.933
	Rho-II(9)	32.765	73.073	136.043	269.774	488.221	931.903	1625.371
	Rho-II(10)	31.945	70.754	137.701	265.471	460.933	931.914	1620.692
	Rho-II(11)	32.519	71.034	135.891	255.254	482.214	936.891	1601.429
	Rho-II(12)	31.252	72.110	135.440	254.854	494.763	931.358	1539.023
	Rho-II(13)	31.829	69.696	135.770	256.624	466.288	938.882	1601.815
	Rho-II(14)	30.495	70.655	132.978	258.113	479.387	910.898	1605.434
	Rho-II(15)	31.501	68.822	136.515	259.286	479.883	895.163	1601.823
	Rho-II(16)	32.046	68.196	132.707	253.984	481.403	889.618	1611.786
	Rho-II(17)	31.013	69.018	135.837	256.376	470.024	900.947	1553.915
	Rho-II(18)	30.882	70.769	133.967	259.030	478.591	912.401	1613.151
	Rho-II(19)	31.270	68.725	137.565	246.645	467.972	909.886	1563.925
	Rho-II(20)	30.816	70.877	138.583	257.066	485.869	928.053	1558.403
	Rho-II(21)	29.991	69.361	135.250	255.126	462.070	908.462	1597.619
	Rho-II(22)	31.665	68.585	136.056	258.017	460.810	914.930	1561.940
	Rho-II(23)	31.504	69.077	135.603	257.620	463.491	875.556	1595.068
	Rho-II(24)	30.972	69.220	131.474	250.957	460.036	928.199	1548.712
	Rho-II(25)	30.293	70.572	132.042	247.871	459.045	940.297	1552.457
	Rho-II(26)	31.116	70.498	131.078	250.654	469.998	900.428	1624.381
	Rho-II(27)	31.157	70.591	130.517	256.153	469.374	905.789	1564.624
	Rho-II(28)	31.299	69.685	131.235	250.569	458.874	910.451	1552.052
	Rho-II(29)	31.152	65.279	131.457	257.471	466.517	874.446	1563.666
	Rho-II(30)	31.462	66.993	130.073	248.238	456.056	888.494	1531.153
	Rho-II(31)	30.604	69.306	130.083	249.324	456.744	921.168	1537.191
	Rho-II(32)	29.966	69.566	134.713	246.427	471.221	885.212	1589.255
	Rho-II(33)	30.977	67.368	129.814	252.542	466.338	898.431	1519.044
	Rho-II(34)	31.403	69.306	129.667	255.811	459.793	878.860	1559.187
	Rho-II(35)	30.306	67.685	135.698	247.294	459.470	898.441	1545.427
	Rho-II(36)	30.497	67.028	127.414	251.147	456.266	897.886	1525.237
	Rho-II(37)	30.709	67.771	133.380	250.827	462.544	896.497	1535.057
	Rho-II(38)	31.139	70.075	132.451	248.700	466.083	885.108	1550.112
	Rho-II(39)	31.309	69.541	128.827	243.032	467.953	901.432	1567.521
	Rho-II(40)	30.116	68.424	134.378	257.020	455.951	888.737	1562.777
	Rho-II(41)	29.941	66.800	135.335	250.170	453.047	895.949	1509.551
	Rho-II(42)	31.450	68.624	132.650	245.823	463.587	851.698	1566.502
	Rho-II(43)	31.400	68.212	130.791	248.732	469.217	892.388	1588.341
	Rho-II(44)	32.027	66.876	134.660	249.077	455.938	878.971	1532.720
	Rho-II(45)	31.267	68.000	131.755	246.544	457.973	868.106	1577.727
	Rho-II(46)	30.393	69.807	131.401	250.593	461.427	892.055	1560.950
	Rho-II(47)	30.493	70.709	128.359	246.175	459.624	877.163	1596.402
	Rho-II(48)	30.641	65.813	130.266	243.199	471.416	903.101	1524.050
	Rho-II(49)	30.167	68.771	126.642	250.982	445.497	891.007	1533.201
	Rho-II(50)	30.807	66.867	129.411	249.779	459.047	882.738	1497.848

7.2　実験 2

実験結果は, 表 10,11, 図 4〜8 の通りである. 表の斜線部に関しては, 実験に長い時間を要するため, 計測を省略した. 図 4〜6 と図 7,8 は, どちらも縦軸に計算量対時間比 $t/\sqrt{n}$ をとっているが, 前者の横軸は標数, 後者の横軸は標数のビット数としている. 各アルゴリズムの計算量対時間比と標数との関係を観察するには図 4〜6, 異なる m における計算量対時間比を比較したり増減を見たりするには図 7,8 を参照するとよい.

今回の実験から, 計算時間が最短のアルゴリズムは, 14 ビット以下においては BSGS, 16〜26 ビットについては Rho-Ⅱ(8), 28 ビット以上については Rho-Ⅱ(20) となることがわかった. 総当たりの計算時間は, 10 ビットの場合のみ Rho-I(20) と Rho-Ⅱ(8), (20) より短くなったが, 12 ビット以上においては他のどのアルゴリズムよりも長くなった (表 10).

ここで, Rho における直和分割の個数 m に注目する. Rho-I については, 14 ビット以下の場合は $m = 20$ の計算時間が最も長かったが, 16 ビット以上では $m = 3$ が最長となった. Rho-Ⅱ についても, 16 ビット・18 ビットを境目として同様の逆転現象がみられた (表 10, 図 7,8).

最後に, 各アルゴリズムにおける計算量対時間比の値を見てみる. BSGS と Rho-I については, 標数の増加

につれて始めは減少し, その後増加した (表 11, 図 4,5,7). 一方, Rho-Ⅱ については, 増減を繰り返しながらも全体としては減少する傾向があった (表 11, 図 6,8). さらに, BSGS と Rho-Ⅱ においては, 標数を大きくしていくと計算量対時間比が収束する傾向がみられた (図 4,6).

表 10　計算時間　$[y^2 = x^3 + 7]$

標数	p	547	3511	11839	47017	194119	881539	2744713	15801199	64971103	258652951	544707991
	bit	10	12	14	16	18	20	22	24	26	28	30
n		547	3433	12049	47353	194917	880981	2746417	15798577	64976101	258684961	544722709
t [ms]	総当たり	9.27	56.2	196	780	2380	11500	32300				
	BSGS	5.80	8.25	15.3	37.4	78.4	296	754	4810	19000	83100	130000
	Rho-I(3)	6.16	15.1	40.8	136	371	1720	6540	46900	158000		
	Rho-I(8)	8.76	14.0	23.7	52.2	205	407	2530	11400	37100		
	Rho-I(20)	17.9	24.6	42.2	69.2	160	552	1430	9770	38400		
	Rho-II(3)	7.88	15.2	25.1	43.4	76.3	178	286	714	1330	3270	4410
	Rho-II(8)	9.54	16.7	23.0	35.0	63.1	119	184	380	708	1660	2600
	Rho-II(20)	18.0	27.1	35.7	52.6	73.0	126	215	430	825	1620	2140

表 11　計算量対時間比　$[y^2 = x^3 + 7]$

標数	p	547	3511	11839	47017	194119	881539	2744713	15801199	64971103	258652951	544707991
	bit	10	12	14	16	18	20	22	24	26	28	30
n		547	3433	12049	47353	194917	880981	2746417	15798577	64976101	258684961	544722709
t/sqrt(n) [ms]	BSGS	0.248	0.141	0.139	0.172	0.178	0.315	0.455	1.21	2.36	5.17	5.57
	Rho-I(3)	0.263	0.257	0.372	0.623	0.840	1.83	3.94	11.8	19.7		
	Rho-I(8)	0.375	0.239	0.216	0.240	0.465	0.434	1.53	2.86	4.60		
	Rho-I(20)	0.767	0.420	0.385	0.318	0.362	0.588	0.867	2.46	4.77		
	Rho-II(3)	0.337	0.259	0.229	0.199	0.173	0.189	0.173	0.180	0.165	0.203	0.189
	Rho-II(8)	0.408	0.285	0.210	0.161	0.143	0.127	0.111	0.0956	0.0878	0.103	0.111
	Rho-II(20)	0.768	0.462	0.325	0.242	0.165	0.134	0.130	0.108	0.102	0.100	0.0918

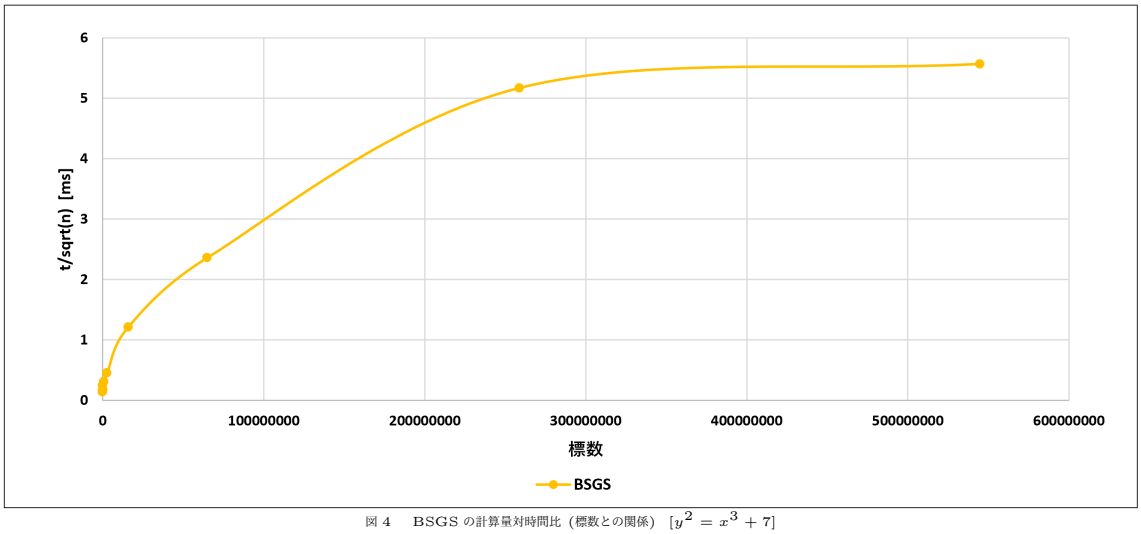

図 4　BSGS の計算量対時間比 (標数との関係)　$[y^2 = x^3 + 7]$

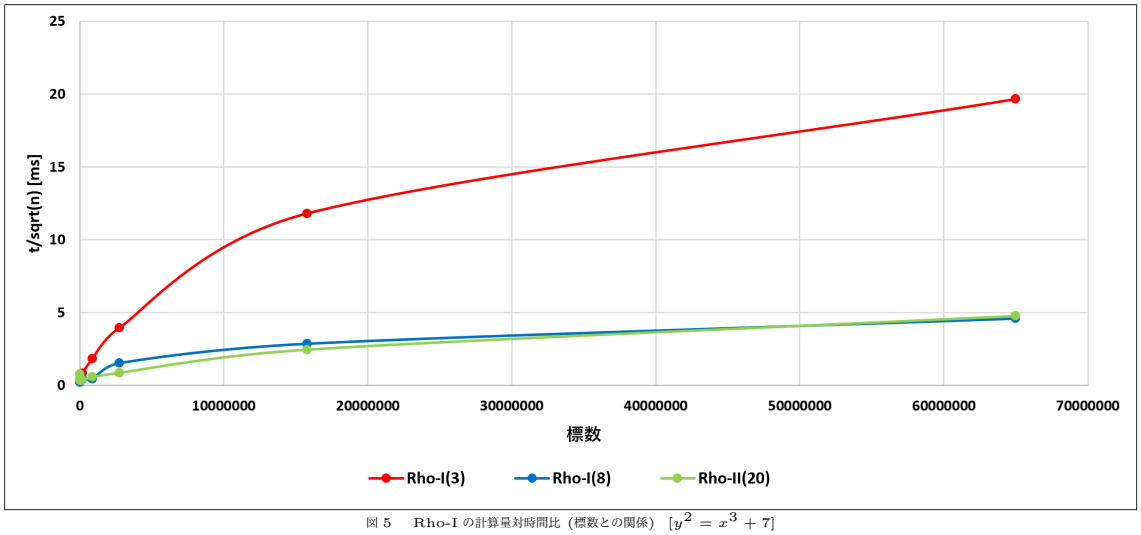

図 5　Rho-I の計算量対時間比（標数との関係）　$[y^2 = x^3 + 7]$

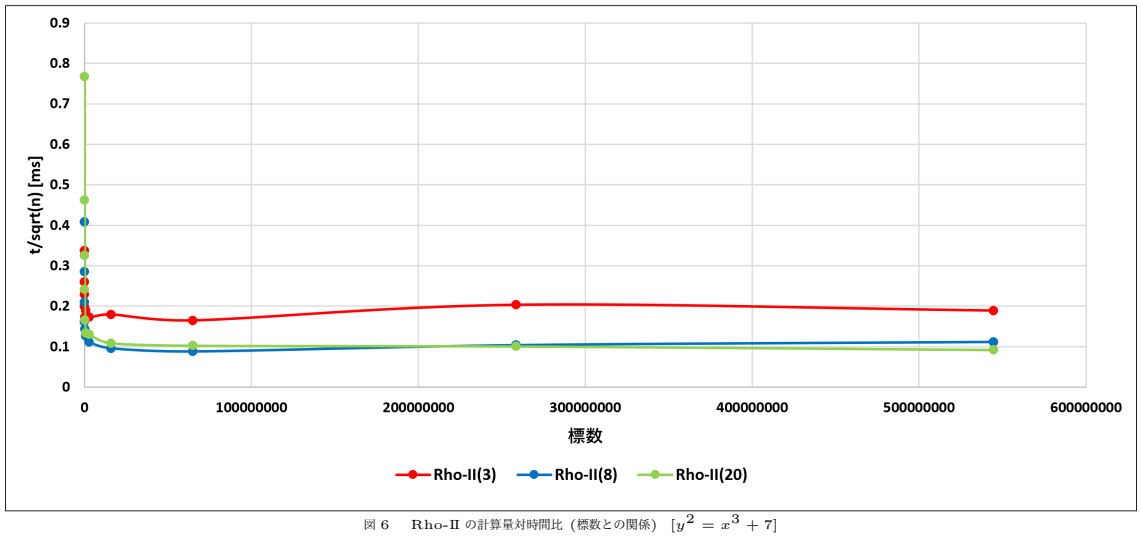

図 6　Rho-II の計算量対時間比（標数との関係）　$[y^2 = x^3 + 7]$

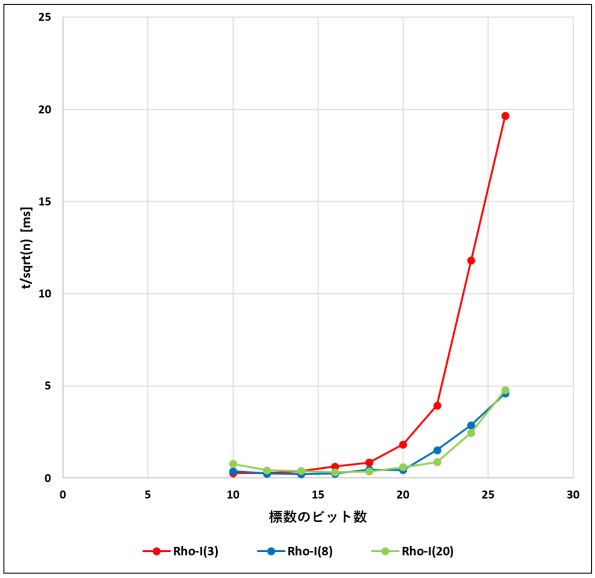

図 7　Rho-I の計算量対時間比（m による比較）$[y^2 = x^3 + 7]$

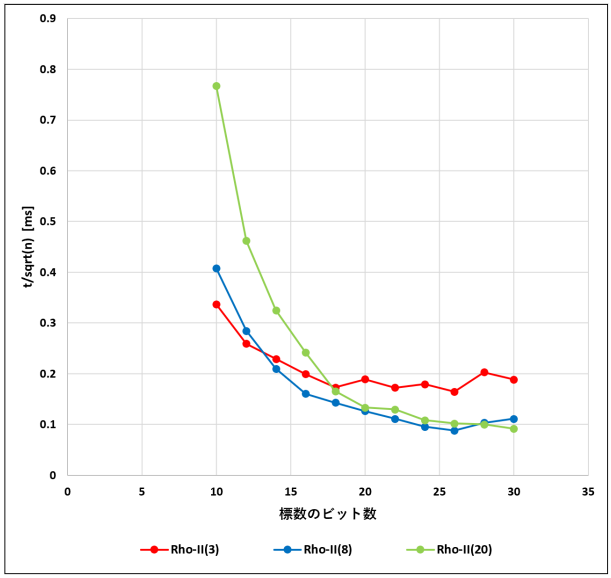

図 8　Rho-II の計算量対時間比（m による比較）$[y^2 = x^3 + 7]$

8　結論と考察

今回の実験で, 標数が 12 ビット以上の場合, BSGS と Rho-I,II が総当たりに比べて非常に効率的であることが観察できた. また, ステップ数が最少となるアルゴリズムは常に Rho-II(ただし m の値は十分大きいと仮定) だったのに対し, 計算時間が最短となるアルゴリズムは, 14 ビット以下においては BSGS, 16〜26 ビットについては Rho-II(8), 28 ビット以上については Rho-II(20) であるということがわかった. 結果として, 標数が大きい場合については, ステップ数・計算時間共に Rho-II が最も効率的なアルゴリズムであるという結論が得られた.

ここで, Rho-I,II の m の値について, ステップ数における優位性と計算時間における優位性に違いが生じた理由を考える. 実験の結果, ステップ数は常に $m = 20$ が最少であった一方, 標数が小さい場合の計算時間は $m = 20$ が最長となった. これは, 関数 f の定義にかかる時間などによる影響が, 標数が小さい場合において顕著に現れたためだと思われる. 直和分割の個数が多いほど, f の定義のために多くのランダムな点 $(M_1, M_2, \cdots, M_m)$ を求める必要があり, 時間がかかったと考えられる.

さらに, Rho-I における直和分割の個数 m とステップ数との関係に注目する. 今回の実験では, 「ステップ数の実測値と期待値の差が最も小さい 5 つ」が $m \geq 16$ の範囲に分布し, 「ステップ数の実測値と期待値の差が最も大きい 5 つ」が $m \leq 10$ の範囲に集中した. それを踏まえ, 「十分にランダムで効率的な関数」の実現には, $m \geq 11$ であることが必要であり, 特に $m \geq 16$ が望ましい, という結論に至った.

最後に, 今回の実験結果をもとに, 実際に情報通信などで使われている楕円曲線暗号について考察した. 例として, 仮想通貨「ビットコイン」の安全性を保証するために使われている楕円曲線暗号を挙げる. 現在, ビットコインでは「secp256k1」(定義式：$y^2 = x^3 + 7$, 標数 256 ビット) と呼ばれる楕円曲線に基づく暗号化技術が使用されている ([4],[10]). 今回の実験では, BSGS と Rho-II について, 標数を大きくしていくと計算量対時間比が定数に近づくという傾向がみられ, 特に Rho-II については, 30 ビットまでの段階で既に挙動が安定していた. そこで, 今回と同様の実験環境で, Rho-II(20) を用いて secp256k1 における ECDLP を解読することを考

えた. 実験結果から, 計算量対時間比は約 0.0918 になると予想できる (表 11, 図 6). secp256k1 の位数は

$$115792089237316195423570985008687907852837564279074904382605163141518161494337$$

であるから, $t/\sqrt{n} = 0.0918$ と仮定して計算した場合, 解読に約 9.91×10^{26} 年かかるという予測が得られた. 宇宙の誕生が約 1.38×10^{10} 年前である ([14]) と言われていることを考えると, これがいかに長い年月であるかわかるだろう. 実用的な楕円曲線暗号についてより正確な情報を得るためには, さらに多くの試行と標数の拡大を行い, 標数が十分大きい場合における計算量対時間比の極限値を得ることが必要である.

参考文献

[1] 辻井重男, 笠原正雄編著: 暗号理論と楕円曲線. 森北出版, 2008.

[2] 清藤武暢: 次世代公開鍵暗号「楕円曲線暗号」とその適切な活用に向けて. 第 14 回情報セキュリティ・シンポジウム,2012.

[3] EdLyn Teske: *Speeding Up Pollard's Rho Method for Computing Discrete Logarithms.* Algorithmic number theory, 1998.

[4] *Bitcoin* 日本語情報サイト. https://jpbitcoin.com/

[5] Afred J. Menezes and Neal Koblitz: *Elliptic Curve Public Key Cryptosystems.* Springer Science+Business Media, 1993.

[6] 川又雄二郎: 射影空間の幾何学. 朝倉書店, 2001.

[7] J. H. Silverman: *The Arithmetic of Elliptic Curves.* 2nd Edition, GTM106, Springer, 2016.

[8] Steven D. Galbraith, Ping Wang and Fangguo Zhang: *Computing Elliptic Curve Discrete Logarithms with Improved Baby-step Giant-step Algorithm.* Adv. Math. Commun. 11(2017), no. 3.

[9] 宮地充子: 代数学から学ぶ暗号理論. 日本評論社, 2012.

[10] *SafeCurves:choosing safe curves for elliptic-curve cryptography.* http://safecurves.cr.yp.to/

[11] J. M. Pollard: *Monte Carlo Methods for Index Computation (mod p).* Mathematics of Computation, volume 32, no.143, 1978, 918–924.

[12] J.Sattler and C. P. Schnorr: *Generating Random Walks in Groups.* Ann. Univ. Sci. Budapest. Sect. Comput. 6, 1985.

[13] EdLyn Teske: *A Space Efficient Algorithm for Group Structure Computation.* Mathematics of Computation, volume 67, no.224, 1998, 1637–1663.

[14] European Space Agency: *Cosmic Detectives.* 2013. https://www.esa.int/Science_Exploration/Space_Science/Cosmic_detectives

[15] J. H. シルバーマン, J. テイト著: 楕円曲線論入門. 足立恒雄, 木田雅成, 小松啓一, 田谷久雄 訳, 丸善出版, 2001.

[16] *sonickun.log.* http://sonickun.hatenablog.com/

[17] 小暮昭仁: 有限体上での楕円曲線の有理点群位数計算. 早稲田大学大学院修士論文, 2019.