数理解析研究所講究録
1038巻 1998年 230-243

Atkin, Elkiesらによる Schoof のアルゴリズム改良の実装について

小暮 淳 [1)]　　伊豆 哲也 [2)]　　横山 和弘 [3)]

Abstract. 楕円暗号において, 演算を行う楕円曲線の選択は安全性を大きく左右する. 任意に選んだ楕円曲線の群位数を求めることができる点において, Schoof のアルゴリズムは優れているが, 計算時間がかかるという欠点があった. 我々は, Atkin, Elkies らによる Schoof アルゴリズムの改良を実装することにより, 実用的な時間内で楕円曲線の群位数を求めることを可能とし, 安全な楕円暗号用パラメタを生成するプログラムを開発した.

1. はじめに

1.1. 楕円暗号におけるパラメタの選択

楕円曲線暗号において,どういった楕円曲線を使用するかというパラメタの選択は,その安全性を大きく左右する.安全性の大きな要因となるのは,有限体上定義された楕円曲線の有理点群位数であり,位数が既存の攻撃法に適さないようなパラメタを選択しなければならない.このようなパラメタを構成する方法は, いくつか知られている. ランダムな曲線の群の位数を計算する Schoof アルゴリズム ([18]), 与えられた位数に対して, それを群の位数として持つパラメタを求める方法 (CM 法 [2],[3], [14] など), Weil 予想を利用する方法 ([12] 等参照) などが代表的である.

1.2. 既存の攻撃法

有限体上定義された楕円曲線での離散対数問題に対しては,準指数時間アルゴリズムは知られておらず,最良のアルゴリズムは square root method であり, その計算量は群の位数を n とすると,$O(\sqrt{n})$ となる.しかしながら,ある種の性質を満たす楕円曲線に対しては,その離散対数問題に有効な多項式時間アルゴリズムが知られている.主なものを挙げると,

(1)Pohlig-Hellman アルゴリズム ([12] 等参照)

位数が小さな素因子の積に分解できるとき, 中国剰余定理により, 各因子での離散対数問題に帰着できる.

[1)] 富士通 (株), Jun Kogure, kogure@rp.open.cs.fujitsu.co.jp
[2)] (株) 富士通研究所, Tetsuya Izu, izu@flab.fujitsu.co.jp
[3)] (株) 富士通研究所, Kazuhiro Yokoyama, yokoyama@para.flab.fujitsu.co.jp

(2)Menezes-Okamoto-Vanstone 帰着 ([13])

supersingular な楕円曲線上の離散対数問題は,元の係数体の 6 次以下の拡大体における離散対数問題に帰着できる.

(3)Smart-Satoh-Araki([22],[17])

anomalous な楕円曲線上の離散対数問題は,等しい位数を持つ有限体の離散対数問題に帰着することができる.

こういった性質を持つ楕円曲線は,楕円暗号には適さないため,使用を避けるべきである.

1.3. パラメタ選択方式の比較

前節からわかるように,何らかの特別な性質を持つ楕円曲線は,その性質を利用した攻撃を受けやすい.楕円暗号に適した楕円曲線を選ぶ際には,特別な性質を持たないよう,ランダムに選ぶことが重要である. Schoof の方法は,曲線をランダムに選択することができる点,及び,暗号強度上良い性質を持つ位数となる曲線の存在が保証されている点 (H.W.Lenstra Jr.[10] による) において優れているが, Schoof アルゴリズムに時間がかかりすぎるという欠点があった.

一方 CM 法の場合,計算時間は少なくてすむが,判別式 D に対する Hilbert Polynomial を計算する際, D の値が小さくなくてはならないという条件がつくために,求められるパラメタに偏りが出てくると考えられる点,また,そのようなパラメタをもつ曲線が,どの程度の割合で存在するかという保証が知られていない点において,楕円暗号で使用するには問題があると考えられる.

1.4. Schoof アルゴリズムの改良

Schoof オリジナル方法の計算量は,係数体を標数 p の素体とした場合, $O((log^8(p))$ であった.この算法において, dominant step となる部分は, $x^{p^2} \bmod f_\ell(x)$ の計算であった. ℓ 分点を決定する $f_\ell(x)$ の次数が $(\ell^2-1)/2$ であるため, ℓ が増大すると,その 4 乗オーダで計算量が増大してしまう.

Atkin, Elkies らは, modular polynomial を用い, ℓ 分点の 1 次元部分空間を定める方程式を explicit に求めることにより,計算量を $O(\log^6(p))$ に改良した (SEA 法 [17]). また, Couveignes, Morain らは, isogeny cycles を用いることにより, ℓ^k 分点を利用することにより, Schoof アルゴリズムにおいて, $t \bmod \ell$ の代わりに, $t \bmod \ell^k$ を求める方法を考案した [4].

我々は, 160 bit 素数 p を位数とする有限体上の楕円曲線の有理点計算を目標とし,汎用数式処理システム Risa/Asir ([16]) 上で実装し,実験を行った.プログラムを Asir 言語 (インタプリタ) で書いた. 数式処理言語で書く利点は, (i) 因数分解, GCD 等の数式処理の用意する操作を使うことで実装が容易になる, (ii) 細かな変更に対応でき,算法の正当性や動作確認が容易である, の 2 点である. Risa は数式処理計算用の c-library として使えることより, すべてを c 言語等の処理系で行うことにより, より一層の高速化が可能になると考える.

現時点では, 標数160bit素数の素体上定義された楕円曲線の有理点群位数を, 最良の場合10分以内で求めることに成功し, 安全な楕円暗号用パラメタを生成するプログラムを開発した. 以下, 2, 3節でその概要及び詳細を述べ, 4節で実装結果について述べる.

2. SEA(Schoof-Elkies-Atkin) 法

以下, p を大きな奇素数とし, E を $y^2 = x^3 + Ax + B$, $A, B \in GF(p)$ で与えられる楕円曲線とする. SEA は Schoof [18] の提案した多項式時間算法に Elkies, Atkin の改良を加えたもので, Morain 等 [6], [11] が SEA 法と呼ぶのに習う.

2.1. SEA 法の概略

まず、Schoof 法の概略から説明する. ([12] 等を参照.) 以下では $\mathbf{Z}, \mathbf{Q}, \mathbf{C}$ で有理整数環, 有理数体, 複素数体を表す.

Schoof 法の基礎: Schoof の算法は ℓ-分点, すなわち ℓ 倍すると無限遠点 O になる点の性質を使っている. E の ℓ 分点全体 $E[\ell]$ は加法群として $\mathbf{Z}/\ell\mathbf{Z} \oplus \mathbf{Z}/\ell\mathbf{Z}$ に同型で, その上に Frobenius 写像 $\phi : (x, y) \to (x^p, y^p)$ が線形写像として作用する. つまり, $GF(\ell)$ 上の2次元線形空間上の線形写像となり, その固有多項式を

$$\phi^2 - t\phi + p = 0 \tag{1}$$

とすれば, $\#E = p + 1 - t$ である. (Tate module $T_\ell(E)$ 上の自己準同型写像として満たす式でもある.) つまり, $P \in E[\ell]$ を取り,

$$\phi^2(P) + pP = t_\ell \phi(P) \tag{2}$$

なる t_ℓ を見つければ, $t \equiv t_\ell \pmod{\ell}$ となる. ここで t_ℓ は $0, 1, 2, ...$ と順に探すことになる。($P \in E[\ell]$ より, $\ell P = O$ であり, t_ℓ は mod ℓ で決まる.) t については, Hasse の定理により

$$-2\sqrt{p} \le t \le 2\sqrt{p} \tag{3}$$

となる. 従って, ℓ をいくつか取り, その積が $4\sqrt{p}$ を越えるようにすれば, 中国剰余定理により t が計算できる. 素数分布定理の系として, 必要な素数の最大を L とすれば $L = O(\log(p))$ であることが示される.

無限遠点以外の ℓ 分点の異る x 成分全体を根とする多項式を ℓ 分多項式と呼び f_ℓ で表す. この時, 方程式 (2) は $GF(p)[x, y]/(y^2 - x^3 - Ax - B, f_\ell(x))$ 上で計算される. f_ℓ の次数は $(\ell^2 - 1)/2$ であり, この次数の大きさが効率に問題を引き起こす: 各 ℓ に対し $\ell = O(\log(p))$ であり, $\deg(f_\ell) = O(\log^2(p))$ である. 各 t_ℓ の計算では、$x^{p^2} \pmod{f_\ell}$, $y^{p^2} \pmod{f_\ell, y^2 - x^3 - Ax - B}$) の計算が dominant であり, 通常の乗算法を用いた場合 $O(\log^7(p))$ binary steps を必要とする。よって最終的な計算量は $O(\log^8(p))$ となる.

Elkies [7] のアイデアは, その計算を (可能な場合に) f_ℓ の因子でその次数が $(\ell - 1)/2$ となる g_ℓ を利用するもので, t_ℓ の計算は $O(\log^5(p))$ binary steps で済む. このような素数は 1/2 の確率で分布しており, Elkies の改良は 確率的算法ではあるが, その期待計算量が $O(\log^6(p))$ を実現した. Elkies のアイデアの実現法として, [7],[15], [5], [4] 等がある.

Atkin [1] は $t \bmod \ell$ の値の可能性を絞ることで効率化を図った. Atkin の方法は Elkies の方法を補間することができ, 両者の改良を入れたものが SEA である.

SEA の基本スキーム ([11],[15] を参照.) 以下では E は non super-singular とする.
(I) $t \bmod \ell$ 計算: 素数 ℓ を 2 から順に取り以下の操作を行う. 素数の積が $4\sqrt{p}$ を越えた時点で (I) を終了する.
(1) modular polynomial $\Phi_\ell(X, J)$ を計算する. ($GF(p)$ 上の 2 変数多項式である.)
(2) $\bar{\Phi}(X) = \Phi_\ell(X, j(E)) \bmod p$ が modulo p で根を持つかどうかを調べる.

(2-E) 根を持つ場合. (ℓ を Elkies 素数と呼ぶ.)
Elkies のアイデアを使い, $f_\ell(X)$ の因子 g_ℓ を計算する.
次に, 以下を満たす値 k を探す.

$$(X^p, Y^p) = k(X, Y)$$

これより $t \equiv k + p/k \pmod{\ell}$ が分かる.

(2-A) 根を持たない場合. (ℓ を Atkin 素数と呼ぶ.)
$\bar{\Phi}(X) \bmod p$ の因数分解の型より t の可能な値を決める. それらの集合を $\mathcal{T}_\ell$ とする.

以下 (I) で得られた Elkies 素数全体を $\mathcal{E}$, Atkin 素数全体を $\mathcal{A}$ とする.
(II) t 計算: 位数の候補 $p + 1 - T$, ここで

$$T \bmod \ell = t_\ell \text{ for } \ell \in \mathcal{E}, \quad T \bmod \ell \in \mathcal{T}_\ell \text{ for } \ell \in \mathcal{A},$$

の中から正しい位数を選択する.

SEA の実装については Atkin 素数の取扱が重要である. なぜならば, $\mathcal{T}_\ell$ の個数が ℓ とあまりかわらなければ, 候補の個数が膨大になり, 効率的にはなりえない. そこで, Atkin 素数の場合, $\mathcal{T}_\ell$ が多いものは選択しないという戦略が入る. Elkies 素数には, 次に述べる isogeny cycle を行うかどうかの戦略が入る. 戦略に関する詳細は次の節で議論する.

isogeny cycle の利用: Morain 等 [15] [6] の研究により, ℓ が Elkies 素数の場合に, $t \bmod \ell$ から $t \bmod \ell^2$, $t \bmod \ell^3$, が効率良く計算できる. すなわち、ℓ^k 分多項式 f_{ℓ^k} の因子 g_{ℓ^k} が isogeny cycle を利用して計算できる. この時、$\deg(g_{\ell^k}) = \ell^{k-1}(\ell - 1)/2$ であり, $\deg(f_{\ell^k}) = (\ell^{2k} - 1)/2$ の約平方根となる. (実際には, より次数の小さい因子を使うこともできる.)

isogeny cycle を利用して $t_{\ell^k} = t \bmod \ell^k$ を求める場合, SEA の基本スキーム の (I) 中の素数の積をつくる部分で ℓ を ℓ^k に置き換え, (II) の中の位数の候補 $p + 1 - T$ は $T \equiv t_{\ell^k} \pmod{\ell^k}$ を満たすようにとることになる.

2.2. Atkin-Elkies の改良

以下 E は non super-singular とし, modular polynomail Φ_ℓ を使った方法について説明する. Φ_ℓ をその他の「同等」な多項式に置き換える方法 [7], [15] もあるが, 今回の実装では扱わなかった. 実装の中の Elkies 素数の場合の g_ℓ の構成については [19] にしたがった. 以下は, いくつかの論文より抽出したものをまとめ直したものである. 基礎的な数学知識は [21] を参照されたい.

2.2.1. 数学的背景

$\mathbf{C}$ 上の楕円曲線 E_0 で E の持ち上げ $E_0 : y^2 = x^3 + A_0x + B_0$ を考える。(A_0, B_0 を有理整数とし、$A_0 \equiv A \pmod p$, $B_0 \equiv B \pmod p$ となるものを考えてもよい.) この時, 格子 $L = \mathbf{Z}\omega_1 \oplus \mathbf{Z}\omega_2$、$Im(\omega_2/\omega_1) > 0$ が存在し, Weierstrass の $\mathcal{P}$ 関数より, 同型

$$\mathbf{C}/L \ni z \to (\mathcal{P}(z), d\mathcal{P}/dz(z)) \in E_0(\mathbf{C})$$

が与えられる. 以下 $\tau = \omega_2/\omega_1$ とおく. ℓ-分多項式については、$\mathbf{C}/L$ での ℓ 等分点より,

$$f_\ell^{E_0}(x) = \prod_{1 \le r \le (\ell-1)/2, 0 \le s < \ell} (x - \mathcal{P}((r + s\tau)/\ell))$$

となる。$f_\ell^{E_0}$ は $\mathbf{Q}(A_0, B_0)$ 上の多項式となる. $E_0[\ell] \subset E_0(K)$ なる代数拡大体 を K, 整数環を O_K とすると, p の上にある O_K の素イデアル $\mathcal{M}$ が存在して, $E[\ell] \subset E(O_K/\mathcal{M})$ とできる. (剰余類環 $O_K/\mathcal{M}$ を $GF(p)$ の拡大体と見る.) この時, O_k から $O_K/\mathcal{M}$ への射影 $proj_K$ により, 各 $P \in E_0[\ell]$ に対して $proj_K(P) \in E[\ell]$ となる対応ができる. $proj_K(f_\ell^{E_0}(x))$ が f_ℓ となる.

modular polynomial: Φ_ℓ が modular polynomial of oder ℓ であるとは, $\Phi_\ell(x, y) \in \mathbf{Z}[x, y]$ であり

$$\Phi_\ell(x, j(\tau)) = (x - j(\ell\tau)) \prod_{i=0}^{\ell-1} (x - j((\tau + i)/\ell))$$

となるものをいう. $\Phi_\ell(x, j(\tau))$ についても modular polynomial と言う. 上記の $\Phi_\ell(x, j(\tau))$ の $GF(p)$ での像を E の modular polynomial と言い, 同じ記号 Φ_ℓ で表す.

$\mathbf{E}[\ell]$ とその部分群および isogeny (同種写像): $E[\ell]$ には $\ell + 1$ 個の位数 ℓ の部分群が存在する. それを $C_1, \ldots, C_{\ell+1}$ と書いておく. $E_0[\ell]$ についても部分群 $C_{0,1}, \ldots, C_{0,\ell+1}$ が存在し, 射影 $proj_K$ により, $proj_K(C_{0,i}) = C_i$ となる対応ができる. この時次が成り立つ. (証明は [19] を参照.)

補題 1 各 C_i に対して, 楕円曲線 E_i と *isogeny* $\psi_i : E \to E_i$ で $Kernel(\psi_i) = C_i$ になるものが存在する. $\Phi_\ell(x, j(E))$ の根は E_i の *j-invariant* $j(E_i)$ である. 特に $j(E_i)$ が $GF(p)$ の元の場合には, E_i は $GF(p)$ 上の曲線として実現できる.

一般的な記法として, 上記の E_i を E/C_i と表す. 各 ψ_i を ℓ-isogeny と呼ぶ.

補題 2 $\Phi_\ell(x, j(E))$ が $GF(p)$ 上で既約因子 $f_1, \ldots, f_s$ を持った時、$\deg(f_1) + \cdots + \deg(f_s)$ を *degree partiton* と呼ぶ. *degree partition* について次のいずれかが成り立つ.
(i) $1 + r$. ϕ は $E[\ell]$ 上で重根となる固有値を持つが非対角. 判別式 $t^2 - 4p$ は $\bmod \ell$ で平方数.
(ii) $1 + 1 + r + \cdots + r$. ϕ は $E[\ell]$ 上で固有値を $GF(\ell)$ に持つ. $t^2 - 4p$ は $\bmod \ell$ で平方数.
(iii) $r + \cdots + r$, $r > 1$. ϕ は $E[\ell]$ 上で固有値として互いに共役なものを $GF(\ell^2)$ に持つ. $t^2 - 4p$ は $\bmod \ell$ で非平方数.
いずれの場合も $r = |\phi|$ (これは $PGL(\ell, 2)$ の中での位数) であり, $t^2 \equiv (\zeta + \zeta^{-1})^2 p \pmod \ell$ となる. ここで ζ は $\bar{G}F(\ell)$ ($GF(\ell)$ の代数閉体) での 1 の r 乗根.

(i)(ii) の場合に ℓ を Elkies 素数と言い, (iii) の場合に Atkin 素数と言う. Elkies/Atkin 素数は t^2-4p が平方数かどうかできまるため, 平均として 1/2 の割合が期待される.

Elkies/Atkin 素数の判定および r の計算は, $\Phi_\ell(x,j(E))$ の Distinct Degree Decomposition (DDD) により計算できる. すなわち, $1 \le k \le \ell+1$ に対する

$$\gcd(x^{p^k}-x, \Phi_\ell(x,j(E)))$$

の次数により簡単にできる。この計算は、$O(\ell^2 \log^3(p))$ 程度でできる. さらに, その根の計算は 2 次多項式の因子分解になるが probabilistic method または、mod p での 2 乗根計算法を使えば $O(\log^5(p))$ 以下で計算できる. (最新の算法については [20] を参照.)

2.2.2. Atkin 素数

ℓ が Atkin 素数の場合, $\mathcal{T}_\ell$ の個数は以下になる. ([11] 参照.) したがって, この値を見て SEA において ℓ を使うかどうかが判定される. $\mathcal{T}_\ell$ の具体的な計算法については実装の部分で説明する.

補題 3 $\#\mathcal{T}_\ell = \varphi(r)$, ここで $\varphi(r)$ は *Euler* 関数.

2.2.3. Elkies 素数

ℓ が Elkies 素数の場合, すなわち, $\Phi_\ell(x,j(E))$ が $GF(p)$ で根を持つ場合を考える. $E[\ell]$ には, $GF(\ell)$ 上の ϕ の固有部分空間 C が存在し, $j(E/C)$ は $\Phi_\ell(x,j(E))$ の根である. この時, $P \in C$ に対して,

$$\phi(P) = sP$$

なる固有値 s が存在する. (s として, $-(p-1)/2 \le s \le (p-1)/2$ にとる.) この s より $t \equiv s+p/s \pmod{\ell}$ となる. C の元の異る x 座標全体の集合を $\mathcal{C}$ とすると, $\#\mathcal{C} = (\ell-1)/2$ であり, C が ϕ 不変であることより,

$$g_\ell(x) = \prod_{\alpha \in \mathcal{C}} (x-\alpha)$$

は $GF(p)$ 上の多項式で f_ℓ の因子となる. E_0/C_0 を対応する E_0 の isogeny とすれば, C_0 の元より同様にして得られる $g_\ell^{E_0}$ の射影 $proj_K$ での像が g_ℓ である. ℓ-isogeny $\psi : E \to E/C$ を見ると

$$\psi : E \ni (x,y) \to (k_1(x)/g_\ell^2(x), k_2(x,y)/g_\ell^3(x)) \in E/C$$

となる. ($\deg(k_1) = \ell$ であることに注意する.) 以下 [19] による g_ℓ の構成についての概要を述べる.

g_ℓ の計算法: 次の 4 つの部分からなる.

(E-1) $\Phi_\ell(x,j(E))$ の $GF(p)$ 上の根 $\hat{j}$ と, それに対応する楕円曲線 E/C の Wierestrass の標準形を決める: $E/C : y^2 = x^3 + \hat{A}x + \hat{B}$. ここで, $\hat{j} = j(E/C)$.

(E-2) $A, B, \hat{A}, \hat{B}$ より $g_\ell(x)$ の $(\ell-3)/2$ 次の係数 $a_{(\ell-3)/2}$ を計算.

(E-3) $A, B, \hat{A}, \hat{B}$ より $E, E/C$ に対応する Weierstrass の $\mathcal{P}$ 関数の Laurent 級数の係数 $c_k, \hat{c}_k$ を求める。

$$\mathcal{P}(z) = 1/z^2 + \sum_{i=1} c_k z^{2k}, \quad \hat{\mathcal{P}}(z) = 1/z^2 + \sum_{i=1} \hat{c}_k z^{2k}.$$

(E-4) $a_{(\ell-3)/2}, c_1, \ldots, c_{(\ell-1)/2}, \hat{c}_1, \ldots, \hat{c}_{(\ell-1)/2}$ より g_ℓ の他の係数を計算する。

上記の計算は, 基本的には $E_0, E_0/C_0$ に関する計算に他ならない. この計算を有限体 $GF(p)$ 上の計算と見なすことで $E, E/C$ に関する計算となる. ($E_0, E_0/C_0$ での計算に現れる全ての数を含む代数的拡大体 K とその整数環 O_K およびその素イデアル $\mathcal{M}$ で $O_K/\mathcal{M} = GF(p)$ となるものが存在し, すべての計算を mod $(\mathcal{M})$ で見たものが $E, E/C$ に関する計算となる.) 各ステップの計算は「解析的手法」による. つまり 関数 $j(q), E_4(q), E_6(q)$ の関係式の組合せから導出される.

(E-1) (E-2) の計算法の概略： E_0/C_0 に対応する格子を $\hat{L}$ とおく. 格子 L の基底を適当に変更することで,

$$L = \mathbf{Z}\omega_1 \oplus \mathbf{Z}\omega_2, \ \hat{L} = \mathbf{Z}\omega_1 \oplus \mathbf{Z}\ell\omega_2$$

とできる. ここで $\tau = \omega_2/\omega_1$, $\mathcal{I}m(\tau) > 0$ である. ℓ-isogeny $E \to E/C$ ($E_0 \to E_0/C_0$) は $\mathbf{C}/L \to \mathbf{C}/\hat{L}$ の準同型として $z \to \ell z$ に対応する. これより, $j(E_0) = j(\tau)$, $j(E_0/C_0) = j(\ell\tau)$ であり, $j(E) = j(E_0) \bmod \mathcal{M}$, $j(E/C) = j(E_0/C_0) \bmod \mathcal{M}$ である. z の代わりに $q = exp(2\pi iz)$ を考えると, $exp(2\pi i\ell z) = q^\ell$ になることに注意する. Weierstrass の標準形の一般論を使って

$$E_0 : y^2 = x^3 - E_4(q)/48 + E_6(q)/864, \ E_0/C_0 : y^2 = x^3 - E_4(q^\ell)/48 + E_6(q^\ell)/864$$

を得る. $A \equiv -E_4(q)/48 \pmod{\mathcal{M}}$, $B \equiv E_6(q)/864 \pmod{\mathcal{M}}$, $\hat{A} \equiv -E_4(q^\ell)/48 \pmod{\mathcal{M}}$, $\hat{B} \equiv E_6(q^\ell)/864 \pmod{\mathcal{M}}$ となる. Jacobi の関係式より, $j(q), \hat{j}(q) = j(q^\ell), E_4(q), E_6(q)$ の値 $(\bmod \mathcal{M})$ が分かり, Φ_ℓ の偏微分を利用して (ここでは略す), $E_4(q^\ell), E_6(q^\ell) \bmod \mathcal{M}$ が計算できる. 結果として E/C が計算できる.

注意: この方法では, $\hat{j}$ が $\Phi_\ell(x, j(E))$ の重根の場合には計算ができない. この場合, $End(E_0)$ の判別式 Δ に対し $|\Delta| \leq 4\ell^2$ となるので, 計算量が $O(\log^4(p))$ の Cornacchia の効率的方法が適用できる. ([19] を参照.) また, このことは E は CM 法で構成できるものでもあることを意味する. 今回の実装では, この場合を検出し除外している.

C の元の x 座標に関しては、次の重要な関係式が得られる。

$$p_1 = \sum_{(x,y)\in C\setminus\mathcal{O}} x = \frac{\ell(E_2(q) - \ell E_2(q^\ell))}{12} \tag{4}$$

式 (4) を $A, B, j, \hat{j}$ で表すことにより $a_{(\ell-3)/2}$ が得られる. 同じ x 座標を持つ点が 2 つあることより, $-p_1/2$ が求める係数 $a_{(\ell-3)/2}$ である.

(E-3)(E-4) の計算法の概略： E_0 の Weierestrass の $\mathcal{P}$ 関数 $\mathcal{P}(z)$ の係数は以下で求まる. $c_1 = -\frac{A}{5}, c_2 = -\frac{B}{7}$, 以下帰納的に

$$c_k = \frac{3}{(k-2)(2k+3)} \sum_{j=1}^{k-2} c_j c_{k-1-j}$$

次に g_p の残りの係数を決めるのであるが, その前に E/C として別の標準形のものに換える: $E/C : y^2 = x^3 + \hat{A}\ell^4 + \hat{B}\ell^6$. これは $\mathbf{C}/(\mathbf{Z}\omega_1 + \mathbf{Z}\omega_2) \to \mathbf{C}/(\mathbf{Z}\frac{\omega_1}{\ell} + \mathbf{Z}\omega_2)$ に対応するものである. $c_k \in \mathbf{Q}(\omega_1, \omega_2)$ である. 次が成り立つ.

$$z^{\ell-1} g_\ell(\mathcal{P}(z)) = exp(-\frac{1}{2}p_1 z^2 - \sum_{k=1}^{\infty} \frac{\hat{c}_k - \ell c_k}{(2k+1)(2k+2)} z^{2k+2}) \tag{5}$$

$K = \mathbf{Q}(\omega_1, \omega_2)$ の整数環を O_K とすれば, その極大イデアル $\mathcal{M}$ で $O_K/\mathcal{M} = GF(p)$ なるものが存在し, 式 (5) を mod $\mathcal{M}$ で見ることにより, $GF(p)$ 上の関係式が得られる. 具体的には g_ℓ の残りの係数は式 (5) の両辺の z の巾の係数比較で得られる. 例えば、z^2 の項の比較より $a_{(p-3)/2} = -p_1/2$ を, z^4 の項の比較より $a_{(p-5)/2} = \frac{1}{8}p_1^2 - \frac{\hat{c}_1 - \ell c_1}{12} - \frac{\ell-1}{2}c_1$ を得る.

2.2.4. isogeny cycle の利用:

isogeny cycle を利用した方法は計算量的にオーダーが良くなるわけではない. この手法は多分に効率的な実装を目指したものである. また, 用意した modular polynomial が不足した場合には, この手法により計算が続けられるという利点もある. その概略を説明する.([6], [5] を参照.) ℓ を Elkies 素数とし、C を $E[\ell]$ の ϕ の固有部分空間, $\psi : E \to E/C$ を ℓ-isogeny とする. 以下今までの記法をそのまま使う. まず, 次が成り立つことを注意する.

補題 4 ℓ は E/C の *Elkies* 素数である. *Tate module* $T_\ell(E)$ の部分群 C^* で ϕ 不変なものが存在し, 各正整数 k に対して, $E[\ell^k] \cap C^*$ は指数 $\ell + 1$ の部分群となる.

E/C の ℓ-isogeny $\hat{\psi} : E/C \to E'$ を考える. $\mathbf{C}$ 上の楕円曲線への持ち上げを $E_0, E_0/C_0, E'_0$ とする. ℓ-isogeny E' の中で $E_0 \cong \mathbf{C}/L$, $E_0/C_0 = \mathbf{C}/\hat{L}$, $E' = \mathbf{C}/L'$ となる同型に対し

$$L = \mathbf{Z}\omega_1 \oplus \mathbf{Z}\omega_2, \quad \hat{L} = \mathbf{Z}\omega_1 \oplus \mathbf{Z}\ell\omega_2, \quad L' = \mathbf{Z}\omega_1 \oplus \mathbf{Z}\ell^2\omega_2$$

となるようなものが存在し, $\psi, \hat{\psi}$ に対応する準同型写像が ℓ 倍写像になる.

よって isogeny の積 $\psi\hat{\psi} : E \to E'$ は ℓ^2 倍写像 $\mathbf{C}/L \to \mathbf{C}/L'$ に対応し, kernel は $E[\ell^2]$ の位数 ℓ^2 の部分群で $\{(a\omega_1)/\ell^2 \mid a = 0, 1, \ldots, \ell^2 - 1\}$ に対応するものとなる.

$$\psi : E \ni (x, y) \to (k_1(x)/g_\ell^2(x), k_2(x, y)/g_\ell^3(x)) \in E/C$$
$$\hat{\psi} : E/C \ni (x, y) \to (\hat{k}_1(x)/\hat{g}_\ell^2(x), \hat{k}_2(x, y)/\hat{g}_\ell^3(x)) \in E'$$

とすれば, 合成 $\hat{g}_\ell(k_1)$ は次数 $\ell(\ell-1)/2$ の多項式で, これらは、$Kernel(\psi\hat{\psi}) \setminus Kernel(\psi)$ の元の異なる x 座標を根とする. これを g_{ℓ^2} と記す.

E′ の選択: $j(E')$ は $\Phi_\ell(x, j(E/C))$ の根である. $\Phi_\ell(j(E), j(E/C)) = 0$ であるので, かならず $GF(p)$ 上の根のひとつは $j(E)$ である. $\Phi_\ell(x, j(E/C))$ が $GF(p)$ 上で根を 1 つしかない場合には, E' の選択は唯一である. $GF(p)$ 上の根が 2 つある場合には, $j(E)$ でないものが求める $j(E')$ となる. $j(E)$ を選んだ場合には, $L' = \mathbf{Z}\ell\omega_1 \oplus \mathbf{Z}\ell\omega_2$ となり, Kernel は $E[\ell]$ になる. この場合 $E' \cong E$.

g_{ℓ^2} の計算: 次のステップよりなる. ただし E/C は計算済とする. つまり副産物として, E, E/C に対応する Weierstrass の $\mathcal{P}$ 関数 $\mathcal{P}(z)$, $\hat{\mathcal{P}}(z)$ が計算されているものとする.

(I-1) E/C に関する ℓ-isogeny より, $\hat{g}_\ell$ を計算する.
(I-2) ℓ-isogeny $\psi : E \to E/C$ の x-成分 $\frac{k_1(x)}{g_\ell(x)^2}$ の分子 $k_1(x)$ を以下の関係式の z 展開の各係数比較により計算する.

$$\hat{\mathcal{P}}(z) = \frac{k_1(\mathcal{P}(z))}{g_\ell(\mathcal{P}(z))^2}$$

(I-3) 関数合成 $\hat{g}_\ell(k_1(x))$ を計算する. これが g_{ℓ^2} となる.
以上の議論を繰り返すことにより, g_{ℓ^k} を計算することができる.

3. SEA+isogeny cycle の実装

今回は, 160 bit 素数 p を位数とする有限体上の楕円曲線の有理点計算を目標とし, 汎用数式処理システム Risa/Asir ([16]) 上で実装し, 実験を行った. プログラムを Asir 言語 (インタプリタ) で書いた. 数式処理言語で書く利点は, (i) 因数分解, GCD 等の数式処理の用意する操作を使うことで実装が容易になる, (ii) 細かな変更に対応でき, 算法の正当性や動作確認が容易である, の 2 点である. Risa は数式処理計算用の c-library として使えることより, すべてを c 言語等の処理系で行うことにより, より一層の高速化が可能になると考える.

SEA+isogeny cycle, 以下では SEA+i と略す, の実装では大きな戦略として, (1) Atkin 素数の選択, (2) Elkies 素数の場合の isogeny cycle の計算をするかどうか, の 2 点が重要である. この 2 点はそれらの選択により, 如何に計算効率が変化するかを見極める必要がある. そのためにも個々の操作の効率化を行い, 最適化することが望ましい. 本節では, まず個々の操作の効率化にあたっていくつかの例を示し, 次に選択における戦略についての概略を示す.

今回の実装にあたり modular polynomial Φ_ℓ を別途計算し, $\ell = 97$ まで用意した. (伊豆 [8] を参照.) これは, modular polynomial の計算自体が研究の対象でもあることにもよる. modular polynomial 以外の「同等」な多項式についても実装実験を行う予定である.

3.1. 各部分における効率化

ここでは, 各部分の効率化について, いくつかの例をあげて示す.

3.1.1. 整数乗算, 多項式乗算

SEA+i は $GF(p)$ 上の多項式の操作の塊である. その基礎となる整数演算や多項式演算が通常の 2 乗オーダー算法では効率的とはいえない. ([11] を参照.) Risa/Asir の提供する乗算は Karatsuba 法なので, 今回の実装ではその機能をそのまま使用した. 計算量の観点から言えば, 通常算法が $O(n^2)$ であるのに対して Karatsuba は $O(n^{1.6})$ となる. したがって, 全体は $O(n^{5.2})$ 算法になると期待される. 結果として計画通りの効率的な計算が達成できたと言える. 一層の効率化にあたり FFT を使用することを考えている.

3.1.2. dominant steps in (I)

SEA+i の (I) 部分において, 計算量としても, 実際の計算においても dominant な部分は, 以下の箇所である. (通常の整数乗算, 多項式乗算を用いた場合に $O(\log^5(p))$ binary steps)
(a) $\Phi_\ell(x, j(E))$ の DDD 計算.

(b) 固有値 $\phi(P) = sP$ の計算.

(a),(b) を更に検討すると, (a)(b) 内で次の 2 つの操作が問題となっている.

(c) ある多項式 $h(x)$, ここで $\deg(h) = O(\log(p))$, に関する $x^{p^k} \bmod h(x)$.

(d) mP の計算, ここで $1 \leq m \leq |s|$.

(c) は (a),(b) 共に現れ, (d) においては, 等分多項式を最悪 $(\ell^k - 1)/2$ まで計算しなければならいことによる. ($k \geq 2$ は isogeny cycle の場合) 計算効率のために, f_m の計算は $\bmod g_{\ell^k}(x)$ で行う. (これをしないと $O(\log^7(p))$ かかる.)

(c) に関しては, multiplication table を構成することで, 計算効率を上げることができる. ([20]) しかし, 劇的に効果を上げるには FFT を導入する必要がある. (d) についても, 同様である.

3.1.3. (II) の効率化

(II) の方法についてはいくつか実現法が考えられるが, 今回の実装では以下の方法を実験した. サンプルとして $E(GF(p))$ の点 P を取り, それをふるいとして位数の候補 N に対して NP が無限遠点 $\mathcal{O}$ になるかを調べる方法である. 効率化のポイントは楕円曲線の有理点の加法をなるべく減らす点にある. $\mathcal{E}$ の素数の積を M_E とし, 中国剰余定理により, 次の整数 S, $0 \leq S < M_E$, を求めておく: $S \equiv t_\ell \pmod{\ell}$ for $\ell \in \mathcal{E}$ 以下に簡単のため, 例外的な処理を除いた一般の形を記す.

(II-1) $E(GF(p))$ 上の点 P を random に取る.

(II-2) $R = (p + 1 - S)P$, $Q = M_E P$ を計算する.

(II-3) t の候補 t_{can} を生成し以下を行う.

bQ を計算し, $bQ = R$ かどうかを判定する. ここで, $t_{can} = b \times M_E + S$ である.

$bQ = R$ であれば, t_{can} を候補として残し, そうでなければ候補からはずす.

(II-4) 候補が唯一ならば, それが正しい t となる. そうでない場合は (II-1) に戻り P を取り直す.

t の候補生成については, 各 $\ell \in \mathcal{A}$ について, $\mathcal{T}_\ell$ からひとつづつ元を選び, 中国剰余定理により生成する. この場合, Hasse の定理 (3) が criterion として無用な候補を削るのに極めて有効である. また, 有理点の加法演算の効率化も不可欠である.

3.2. Atkin 素数, Elkies 素数における isogeny cycle の選択

本節の冒頭で述べたように, 効率化のための素数選択の戦略には (1) Atkin 素数の選択, (2) Elkies 素数の場合の isogeny cycle の計算をするかどうか, の 2 点がある. 今回の実装では, 素数を 2 より小さい順に取り, 以後の計算が以前に選んだ素数の選択に影響を及ぼさない「逐次」型のものを実験した. 選択の戦略を単純に表現する整数の組 (B_E, B_A), B_E は 選択する Elkies 素数の積の設定する上限, B_A は選択する Atkin 素数の積の設定する上限, を考える. (Morain 等の報告 [6], [5] でも使われている.) すなわち上限を越えた場合にはその素数は選択しないという設計パラメータである. 極端な例をあげると, $(B_E, B_A) = (\infty, 0)$ は Elkies 素数のみを使う方法, $(B_E, B_A) = (0, \infty)$ は Atkin 素数のみを使う方法となる.

残念ながら 160 bit 素数 p に対する (B_E, A_E) の最適な値の理論解析はまだできていない. 今回の実装では実験的に B_A を 10^3 から 10^6 程度までの範囲で動かし, $B_A = 10^5$ の設

定に至った. (modular polynomial が $\ell \leq 97$ ということで, 正確には $(\infty, 10^5)$ という設定にはならない.)

3.2.1. Atkin 素数の選択

Atkin 素数の選択は (II) の実行速度に依存する. (II) を高速に実行するには, Atkin 素数として $\#\mathcal{T}_\ell$ が小さいものが望ましい. 採用する ℓ における $\#\mathcal{T}_\ell$ の上限 B_T も設計パラメータとなる. すなわち, B_T を越えた場合には採用しないという戦略である. 実験では 160 bit 素数 p の場合の B_T として、$B_T = 16$ を選んでいる.

3.2.2. isogeny cycle の計算

Elkies 素数として使用された素数 ℓ の isogeny cycle を利用するかどうかはその計算コストに依存する. $\phi(P) = sP$, $P \in C^* \cap E[\ell^k]$, なる固有値 s の計算に (通常の乗法を用いると)

$$O(\log^3(p)\deg(g_{\ell^k})^2 + |s|\log^2(p)\deg(g_{\ell^k})^2)$$

binary steps 必要となる. 160 bit 素数 p での効率的計算には, 新たなパラーメータとして $\deg(g_{\ell^k})$ に対する上限 B_G を設定することになる. つまり, $\deg(g_{\ell^k}) > B_G$ なる ℓ では isogeny cycle を使わないということである. 実際 k は 高々 2 までと設定するしかない. ($\ell = 2, 3$ を除く) また ℓ の大きさにも上より制限が付くことになる. 以下でこのパラメータの概略を説明する.

一般には $\deg(g_{\ell^k}) = (\ell - 1)\ell^{k-1}/2$ であったが, 次の補題より更に小さい因子が取れる.

補題 5 g_ℓ により決まる $\phi(P) = s_0 P$ なる固有値 s_0 の $GF(\ell)^*$ における位数を d_0 とする. ここで $P \in C^* \cap E[\ell]$ である. そして, d として d_0 が奇数ならば d_0, d_0 が偶数なら $d_0/2$ とする. この時, g_ℓ は $GF(p)$ 上 d 次の因子を持つ. また $\hat{g}_\ell$ も $GF(p)$ 上 d 次の因子を持つ.

s_0 はすでに計算されているので d も分かる. d が小さい場合には $\hat{g}_\ell$ を因数分解することで d 次の多項式 h_ℓ が得られ, 関数合成 $h_{\ell^2} = h_\ell(k_1)$ は次数が $d\ell$ の g_{ℓ^2} の因子になる. そこで g_{ℓ^2} の代わりに h_{ℓ^2} を使えば、計算時間が

$$O(\log^3(p)d^2\ell^2 + |s|\log^2(p)d^2\ell^2)$$

へと向上することになる. ここで $\hat{g}_\ell$ の因数分解という計算が新たに必要となるが, この計算量は $O(\ell^2\log^3(p))$ なので, トータルでは向上することがわかる.

今回の実装においては, isogeny cycle に関して以下に設定した.

(1) $B_G = 150$

(2) $|s|$ が大きくなる場合を避けるため $\ell \leq 31$ に制限.

注意: 用意した Φ_ℓ は $\ell \leq 97$ までという制限があるので, その半分が Elkies 素数になっても $4\sqrt{p}$ にはいたらない. この意味で決定的方法 (Elkies 素数のみ使う) では一部しか計算できないことになる. したがって効率化の意味で導入した Atkin 素数および isogeny cycle は計算できる曲線を増やすということに貢献したことになる.

4. 実装結果

以下の環境で、10個のパラメタについて、有理点群位数を計算するのに要した時間を測定した。

(1) 使用マシン: PentiumPro 200MHz、128MB memory

(2) 数式処理プログラム: Risa/Asir(Unix版)

(3) 素数 p: 160 bit

(4)modular polynomial: $\ell \leq 97$ まで使用

項番	計算時間(秒)	candidates(個)	isogeny cycle使用
1	1146	70000	無
2	681	9000	無
3	1803	73000	有
4	741	24000	無
5	897	49000	無
6	699	18000	無
7	645	4000	無
8	554	12000	無
9	911	24000	無
10	634	700	無

1つのパラメタの計算時間は、最良の場合554秒(9分14秒)、平均871秒(14分31秒)となっている。

isogeny cyclesを使用しないで済んだ場合、20分以内(ほとんどの場合15分以内)で計算は終了している。Atkin素数による候補数を 10^5 程度に押さえた場合、$\ell \leq 97$ のElkies素数及び有効なAtkin素数が十分存在すれば、160bit素体上楕円曲線の位数は約15分以内で計算できる。

それでは足りない場合にisogeny cycleを使用するため、最悪30分程度かかるケースが出てくる。modular polynomialを、より大きな ℓ に対して用意することにより、一層の高速化を図ることができると考えられる。

5. おわりに

楕円暗号で使用する楕円曲線のパラメタを選択する方式の中で、Schoofアルゴリズムを用いた方法が最良のものと考えられる。我々は、SEA法、isogeny cyclesなどを利用し、標数が160bit素数である素体上定義された楕円曲線の場合に、Schoofアルゴリズムを実用的な時間内に実現することができた。今後は、modular polynomialの代わりとして係数膨張の起きない多項式の使用、Schoofオリジナル方法の組み合わせ等により、更なる性能向上を目指す。

参 考 文 献

[1] Atkin, A.O., The number of points on an elliptic curve modulo a prime, preprint, 1988.

[2] Atkin, A.O., Morain, F., Elliptic curves and primality proving, *Math. Comp.* 61 (1993) 29–68.

[3] Chao, J., Tanada, K., Tsujii, S., Design of elliptic curves with controllable lower boundary of extension degree for reduction attacks, In *CRYPTO '94*, Y.Desmedt, Ed., Lecture Notes in Computer Science, 839, pp.50–55, 1994.

[4] Charlap, L.S., Coley, R., Robbins, D.P., Enumeration of rational points on elliptic curves over finite fields, preprint, 1991.

[5] Couveignes, J.-M., Morain, F., Schoof's algorithm and isogeny cycles, In *ANT-I*, L.Adleman and M.-D.Huang, Eds., Lecture Notes in Computer Science, 877, pp.43–58, 1994.

[6] Couveignes, J.-M., Dewaghe, L., Morain, F., Isogeny cycles and the Schoof-Elkies-Atkin algorithm, LIX/RR/96/03, 1996.

[7] Elkies, N.D., Explict isogenies, preprint, 1991.

[8] 伊豆哲也, Risa/Asir による modular polynomial の計算, 京都大学数理解析研究所講究録「数式処理における理論と応用の研究」掲載予定.

[9] Lay, G.-J., Zimmer, H.G., Constructing elliptic curves with given group order over large finite fields, In *ANT-I*, L.Adleman and M.-D.Huang, Eds., Lecture Notes in Computer Science, 877, pp.250–263, 1994.

[10] Lenstra Jr., H.W., Factoring integers with elliptic curves, *Annals of Mathematics* 126 (1987) pp.649–673.

[11] Lercher, R., Morain, F., Counting the number of points on elliptic curves over finite fields: strategy and performances, In *EURO-CRYPTO '95*, L.C.Guillou and J.-J.Quisquater, Eds., Lecture Notes in Computer Science, 921, pp.79–94, 1995.

[12] Menezes, A., Elliptic curve public key cryptosystems, Kluwer Academic Publishers, Boston, 1993.

[13] Menezes, A., Okamoto, T., Vanstone, S.E., Reducing elliptic curves logarithms to logarithms in a finite field, In *STOC '91*, ACM Press, New York, pp.80–89, 1991.

[14] Miyaji, A., Elliptic curves over F_p suitable for cryptosystems, In *AUSCRYPTO '92*, J.Seberry and Y.Zhengs, Eds., Lecture Notes in Computer Science, 718, pp.479–491, 1992.

[15] Morain, F., Calcul du nombre de points sur une courbe elliptique dans un corps fini: aspects algorithmiques, *J. Théor. Nombres Bordeaux* 7 (1995) 255–282.

[16] Noro, M., Takeshima,T., Risa/Asir – a computer algebra system, in: *Proceedings of ISSAC '92*, ACM Press, New York, 1992, pp.387–396.

[17] Satoh, T., Araki, K., Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves, preprint, 1997.

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

[19] Schoof, R., Counting points on elliptic curves over finite fields, *J. Théor. Nombres Bordeaux* 7 (1995) 219–254.

[20] Shoup, V., A new polynomial factorization algorithm and its implementation, *J. Symbolic Computation.* 20 (1995) 363–397.

[21] Silverman, J.H., Advanced topics in the arithmetic of elliptic curves, Graduate Texts in Mathematics Vol.151, Springer-Verlag, 1994.

[22] Smart, N.P., The discrete logarithm problem on elliptic curves of trace one, preprint, 1997.