ディジタル署名 EdDSA で使われている曲線の安全性に関する調査及び評価

安田 雅哉（立教大学理学部）

2020 年 11 月 24 日

目次

第 1 章　本報告書の目的と構成概要 2
1.1　目的 2
1.2　構成概要 2

第 2 章　調査結果・評価結果の概要 3

第 3 章　一般の楕円曲線と Edwards 曲線における加法演算の調査 5
3.1　一般の楕円曲線における加法演算と座標表現 5
3.2　Edwards 曲線とその加算公式 8

第 4 章　楕円曲線暗号に対する攻撃法調査 18
4.1　一般の楕円曲線に適用可能な攻撃法 18
4.2　特殊な曲線にのみ有効な攻撃法 25
4.3　ECDLP に対する攻撃手法のまとめ 28

第 5 章　EdDSA で利用される曲線の安全性評価と効率性に関する考察 29
5.1　EdDSA で利用される曲線に関する解説 29
5.2　EdDSA で利用される曲線に対する攻撃計算量評価 34
5.3　ECDSA と比較した場合の曲線としての効率性に関する考察 38

参考文献 42

第 1 章

本報告書の目的と構成概要

1.1 目的

Edwards 曲線ディジタル署名アルゴリズム（Edwards-curve Digital Signature Algorithm, EdDSA）とは，効率的な点の加算公式を持つ特殊な楕円曲線である Edwards 曲線またはそのツイスト曲線を利用したディジタル署名である．近年のディジタル署名 EdDSA の市場への普及を鑑み，EdDSA がディジタル署名として十分な安全性を有するか否かを判断するために，本報告書では EdDSA で利用される曲線の安全性に関する調査を行うことを目的とする．

1.2 構成概要

Edwards 曲線を利用したディジタル署名 EdDSA で利用される曲線の安全性に関わる脆弱性について，公開されている攻撃方法の有無を調査し，存在する場合はその影響の範囲などについてまとめるなど，対象となる曲線の安全性評価を実施する．本報告書の構成概要は以下である：

- 第 2 章で，本報告書における調査結果・評価結果の概要を述べる．
- 第 3 章では，一般の楕円曲線における加算公式をまとめると共に，ディジタル署名 EdDSA で利用される（ツイスト）Edwards 曲線における加算公式をまとめる．
- 第 4 章では，ディジタル署名を含む楕円曲線暗号の安全性を支える楕円曲線離散対数問題（Elliptic Curve Discrete Logarithm Problem, ECDLP）に対する既存の攻撃手法を調査した結果をまとめる．ただし，既存の攻撃手法の調査については，2020 年 8 月末までに公開された攻撃手法を調査対象とした．
- 第 5 章では，EdDSA で利用される（ツイスト）Edwards 曲線の具体的なパラメータ選択法について解説すると共に，暗号攻撃の観点からこれらの曲線がどの程度安全なのか解析評価する．また，通常の楕円曲線を利用するディジタル署名 ECDSA と比べて EdDSA がどの程度効率的なのか考察する．

第 2 章

調査結果・評価結果の概要

■EdDSA で利用される曲線に関する解説　ディジタル署名 EdDSA では Edwards 曲線と呼ばれる特殊な楕円曲線やそのツイスト曲線を利用する．Edwards 曲線は xy 座標平面内の方程式 $x^2+y^2=1+dx^2y^2$ $(d\neq 0,1)$ で定義される曲線で，その曲線上の点の加算と 2 倍算を効率的に計算することができる．RFC8032 [31] によると，EdDSA では（古典計算機による）暗号攻撃に対して約 128 ビットのセキュリティレベルの Ed25519 と約 224 ビットのセキュリティレベルの Ed448 の 2 種類の実装のどちらかを安全性要件に応じて利用するよう推奨されている．また，Ed25519 では “Curve25519”，Ed448 では “Curve448” と呼ばれるツイスト Edwards 曲線パラメータを利用する．これらの曲線では，可能性のある暗号攻撃を避けるため，元の曲線とそのツイスト曲線の両方の位数が $4r$ または $8r$（ただし，r は巨大素数）の形となるように係数パラメータが選択されている．（ツイスト Edwards 曲線は位数 4 のねじれ点を持つので，4 以上の余因子を必ず持つことに注意する．）特に，Curve25519 は高速実装に適した 255 ビット素数 $p=2^{255}-19$ による素体 $\mathbb{F}_p$ を基礎体に持つ．

■EdDSA で利用される曲線に関して公開されている攻撃・脆弱性の調査　ディジタル署名を含む楕円曲線暗号の安全性は楕円曲線離散対数問題 (ECDLP) の求解困難性に基づく．ECDLP に対する攻撃法は，Pollard の ρ 法や指数計算法などの任意の楕円曲線に適用できる汎用攻撃アルゴリズムと，MOV 攻撃法や SSSA 攻撃などの特殊な楕円曲線にのみ適用可能な特殊攻撃アルゴリズムに大別される（攻撃法のまとめは 4.3 節を参照）．EdDSA で利用される Curve25519 や Curve448 では，MOV 攻撃法や SSSA 攻撃法が有効とならないような曲線パラメータが選択されているため，汎用攻撃アルゴリズムの中で最良の ρ 法が EdDSA に対する最良の攻撃法である．

■EdDSA で利用される曲線の安全性評価　EdDSA に対する最良の攻撃法である ρ 法攻撃は誕生日の逆理に基づく確率的アルゴリズムであるが，Edwards 曲線上の ECDLP に対する ρ 法は通常の楕円曲線とほぼ同じ振る舞いをすることが本報告書内の実験結果（後述の図 5.2）から分かる．これより，Edwards 曲線上の位数 r を持つ ECDLP の ρ 法による平均攻撃計算量は，誕生日の逆理から

$$\frac{\sqrt{\pi r}}{2}\cdot\mathbf{Ed}_{\mathrm{add}}\approx 0.8862\sqrt{r}\cdot\mathbf{Ed}_{\mathrm{add}}$$

と見積もれる．ここで，$\mathbf{Ed}_{\mathrm{add}}$ は Edwards 曲線上の点の加算コストとする．（ただし，攻撃者有利な条件として，Edwards 曲線上の逆元計算による $\sqrt{2}$ 倍の高速化も考慮した.）具体的には，Curve25519 の位数は $r \approx 2^{252}$ より，Curve25519 における ECDLP を攻撃するには，平均的に約 $0.8862 \cdot 2^{126} = 2^{125.8257}$ 回の点の加算が必要となる．また，Curve448 の位数は $r \approx 2^{446}$ より，Curve448 における ECDLP を攻撃するには，平均的に約 $0.8862 \cdot 2^{223} = 2^{222.8257}$ 回の点の加算が必要となる．一方，128 ビットセキュリティレベルの ECDSA での利用が推奨されている P-256 曲線に関して，P-256 の位数が $r \approx 2^{256}$ より，P-256 における ECDLP を攻撃するには，平均的に約 $0.8862 \cdot 2^{128} = 2^{127.8257}$ 回の点の加算が必要となる．これより，P-256 と同程度のセキュリティレベルに設定されている Curve25519 と安全性比較すると，Curve25519 上の方が平均的に約 4 倍少ない回数の楕円加算で攻撃できる．さらに，ツイスト Edwards 曲線は効率的な点の加算公式を持つため，P-256 よりも Curve25519 上の方がより効率的に点の加算が可能であり，ECDLP をより高速に攻撃できる．例えば，P-256 よりも Curve25519 の方が最大 2 倍高速に楕円加算ができたと想定すると，P-256 よりも Curve25519 における ECDLP の方が平均的に最大 8 倍高速に攻撃できる．ただし，Curve25519 における ECDLP を攻撃するには，少なくとも $2^{125.8257}$ 回の楕円加算が必要であり，ほぼ 128 ビットのセキュリティレベルを持つと結論付けれる．

■ECDSA と比較した場合の曲線としての効率性に関する考察　通常の楕円曲線と Edwards 曲線上の点の加算と 2 倍算の計算コスト比較は表 3.1 にまとめた．具体的には，基礎体上の乗算コスト $\mathbf{M}$ と 2 乗算コスト $\mathbf{S}$ の比が $\mathbf{S}/\mathbf{M} = 0.8$ の場合，通常の楕円曲線で標準的に利用する射影座標表現における点の加算コストは $10.8\mathbf{M}$ で 2 倍算コストは $9.8\mathbf{M}$ であるのに対し，Edwards 曲線の射影座標表現における点の加算コストは $9.8\mathbf{M}$ で 2 倍算コストは $6.2\mathbf{M}$ である．これより，同一の基礎体を利用した場合，射影座標表現では Edwards 曲線の方が点の加算で約 9.4%，点の 2 倍算で約 36.7% 効率的に計算できる．また，楕円曲線を利用したディジタル署名では，署名生成時に楕円曲線の点 P のスカラー倍算 nP を行い，署名検証時には楕円曲線の点 P_1, P_2 の複数スカラー倍算 $n_1P_1 + n_2P_2$ を主に行う．表 3.1 と同じように，表 5.1 に座標表現による楕円曲線上のスカラー倍算 nP の計算コスト比較をまとめた．表 5.1 より，同一の基礎体を利用した場合，スカラー倍算 nP に関しては Edwards 曲線の方が最大 33% 程度効率的に行うことができる．一方，表 5.2 に座標表現による楕円曲線上の複数スカラー倍算 $n_1P_1 + n_2P_2$ の計算コスト比較をまとめた．表 5.2 より，同一の基礎体を利用した場合，複数スカラー倍算 $n_1P_1 + n_2P_2$ に関しても Edwards 曲線の方が最大 28% 程度効率的に行うことができることが分かる．特に，EdDSA で利用される Curve25519 においては，高速実装に適した基礎体 $\mathbb{F}_p$ を選択しており，基礎体上の演算の高速化分を考慮すれば，更に効率的に（複数）スカラー倍算を行うことが可能となる．実際，図 5.3 に P-256 曲線による ECDSA と Curve25519 による EdDSA のハードウェア実装による処理時間の比較を示す．ただし，処理時間は鍵生成・署名生成・署名検証の合計時間で，その単位は cycles 数とする（詳細は [13] を参照）．ハードウェアの実装方法により処理時間が大きく異なるため参考程度ではあるが，図 5.3 より Curve25519 による EdDSA の方が最大 2 倍程度高速であることが分かる．

第 3 章

一般の楕円曲線と Edwards 曲線における加法演算の調査

楕円曲線暗号は Koblitz [32] と Miller [37] がほぼ同時期かつ独立に考案した暗号技術で，射影平面内の斉次 3 次曲線として定義される楕円曲線上の点集合における特殊な加算法に基づき暗号化・復号を行う暗号方式である．本章では，一般の楕円曲線における加法演算をまとめると共に，本報告書の調査対象である EdDSA で利用される Edwards 曲線における加法演算をまとめる．

3.1 一般の楕円曲線における加法演算と座標表現

本節では，一般の楕円曲線の加法演算をまとめる．特に，楕円曲線暗号で利用される代表的な座標表現とその表現における加法演算についてまとめる．

3.1.1 楕円曲線における加法演算

標数 $p \geq 5$ の体 k 上の楕円曲線は（短い）Weierstrass 方程式

$$E : y^2 = x^3 + ax + b \quad (a, b \in k,\ \Delta(E) = 4a^3 + 27b^2 \neq 0) \tag{3.1}$$

で表せる[*1]．無限遠点 ∞ を含めた楕円曲線 E 上の点の集合全体は，次で定義する加法演算において可換群をなす（証明は [51, 56] などを参照）：

(i) 無限遠点 ∞ と無限遠点を含めた E 上の任意の点 P との加算について $P + \infty = \infty + P = P$ と定める．（つまり，無限遠点 ∞ が群における零元の役割を果たす.）

(ii) 無限遠点ではない E 上の点 $P = (x, y)$ の逆元を $-P = (x, -y)$ と定める．(一方，無限遠点 ∞ の逆元は ∞ 自身と定める.)

(iii) 無限遠点ではない E 上の 2 つの点 $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ が $P_1 \neq -P_2$ を満たすとする．このとき，2 つの点 P_1 と P_2 の加算点 $P_3 = P_1 + P_2 = (x_3, y_3)$ の座標を以下で定める：

[*1] $\Delta(E)$ は楕円曲線の判別式であり，$\Delta(E) = 0$ の場合は曲線が特異点を持つ.

- $P_1 \neq P_2$ の場合[*2]：

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \\ y_3 = m(x_1 - x_3) - y_1. \end{cases} \left(m = \frac{y_2 - y_1}{x_2 - x_1} \right), \tag{3.2}$$

- $P_1 = P_2$ の場合：$y_1 = 0$ なら $P_3 = \infty$ と定める．一方，$y_1 \neq 0$ なら，P_1 の 2 倍点 $P_3 = 2P_1 = (x_3, y_3)$ の座標を以下で定める：

$$\begin{cases} x_3 = m^2 - 2x_1 \\ y_3 = m(x_1 - x_3) - y_1. \end{cases} \left(m = \frac{3x_1^2 + a}{2y_1} \right), \tag{3.3}$$

特に，無限遠点ではない E 上の 2 つの点 $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ の座標 x_i, y_i $(i = 1, 2)$ が体 k の元であれば，P_1 と P_2 の加算点 $P_3 = (x_3, y_3)$ の座標 x_3, y_3 も体 k の元である．ここで，体 k の座標を持つ楕円曲線 E 上のすべての点に無限遠点を含めた集合を

$$E(k) = \left\{ (x, y) \in k^2 : y^2 = x^3 + ax + b \right\} \cup \{\infty\}$$

とする．このとき，上記で定めた楕円曲線上の加法演算により，集合 $E(k)$ は可換群となる．

3.1.2　楕円曲線における代表的な座標表現とその加算公式

楕円曲線の点の表現を変更することで，楕円曲線上の加法演算を効率的に行うことができる．ここでは，楕円曲線における代表的な座標系とその点の表現での加法演算の代表例を紹介する．

2 つの正の整数 c, d を用いて，集合 $k^3 \setminus \{(0,0,0)\}$ における同値関係を次のように定める：2 つの元 $(X_1, Y_1, Z_1), (X_2, Y_2, Z_2) \in k^3 \setminus \{(0,0,0)\}$ に対して，

$$(X_1, Y_1, Z_1) \sim (X_2, Y_2, Z_2) \Longleftrightarrow X_1 = \lambda^c X_2, Y_1 = \lambda^d Y_2, Z_1 = \lambda Z_2 \ (\exists \lambda \in k^\times)$$

と定める．元 (X, Y, Z) を含む同値類を $(X : Y : Z) = \left\{ \left(\lambda^c X, \lambda^d Y, \lambda Z \right) : \lambda \in k^\times \right\}$ と表し，**射影点**と呼ぶ[*3]．すべての射影点の集合を**射影平面**と呼び，$\mathbb{P}^2(k)$ と表す．特に，射影平面 $\mathbb{P}^2(k)$ の部分集合

$$\mathbb{P}^2(k)^* = \left\{ (X : Y : Z) \in \mathbb{P}^2(k) : Z \neq 0 \right\}$$

と座標平面上の点集合と 1 対 1 に対応する．具体的には, 座標平面上の点 (x, y) は射影点 $(x : y : 1)$ に対応する．逆に，$Z \neq 0$ なる射影点 $(X : Y : Z)$ は座標平面上の点 $(X/Z^c, Y/Z^d)$ に対応させる．この座標平面上と射影平面における対応を利用することで，座標平面上の代数方程式 (3.1) で定義される楕円曲線を射影平面内の曲線として楕円曲線を表すことができる．以下で，射影平面内で定義される楕円曲線とその加法演算の代表例を紹介する：

[*2] 幾何学的なイメージ図については，[51, Figure 3.3 in Chapter III] を参照．また，通常の Weierstrass 方程式で定義される楕円曲線上の点の加算公式については，[51, Section III.2] を参照．

[*3] 逆に言い換えると，元 (X, Y, Z) は射影点 $(X : Y : Z)$ の代表元である．

■（標準）射影座標表現　$c = d = 1$ とする．この場合，座標平面上の Weierstrass 方程式 (3.1) で定義される楕円曲線は，（標準）射影平面内の（斉次）3 次曲線

$$E : Y^2Z = X^3 + aXZ^2 + bZ^3 \tag{3.4}$$

として表せる．特に，$Z = 0$ を満たす E 上の射影点は $(0 : 1 : 0)$ のみであり，楕円曲線の無限遠点 ∞ を射影点 $(0 : 1 : 0)$ に対応させておく．これにより，楕円曲線の点の加算が定まる．具体的に，楕円曲線 E 上の 2 つの射影点を $P_1 = (X_1 : Y_1 : Z_1)$ と $P_2 = (X_2 : Y_2 : Z_2)$ とし，その加算点を $P_3 = P_1 + P_2 = (X_3 : Y_3 : Z_3)$ とする．（特に，各射影点 P_i は方程式 (3.4) を満たす．）このとき，Weierstrass 方程式で表現された楕円曲線の加算公式 (3.2) と 2 倍算公式 (3.3) より，射影点 $P_3 = P_1 + P_2$ の各射影座標 X_3, Y_3, Z_3 は以下で定まる（ただし，楕円曲線上の射影点 $P = (X : Y : Z)$ の逆元は $-P = (X : -Y : Z)$ となる．）：

- $P_1 \neq \pm P_2$ の場合

$$\begin{cases} u = Y_2Z_1 - Y_1Z_2, \quad v = X_2Z_1 - X_1Z_2, \quad w = u^2Z_1Z_2 - v^3 - 2v^2X_1Z_2, \\ X_3 = vw, \quad Y_3 = u(v^2X_1Z_2 - w) - v^3Y_1Z_2, \quad Z_3 = v^3Z_1Z_2. \end{cases}$$

- $P_1 = P_2$ の場合

$$\begin{cases} t = aZ_1^2 + 3X_1^2, \quad u = Y_1Z_1, \quad v = uX_1Y_1, \quad w = t^2 - 8v, \\ X_3 = 2uw, \quad Y_3 = t(4v - w) - 8Y_1^2u^2, \quad Z_3 = 8u^3. \end{cases}$$

- $P_1 = -P_2$ の場合は $P_3 = P_1 + P_2 = (0 : 1 : 0)$ となる．

特に，Weierstrass 方程式の表現による公式 (3.2) と (3.3) では割り算の操作が必要なのに対し，上記の射影座標による表現では割り算が一切必要ない．また，計算の順序を工夫することで，射影座標による点の加算 $P_1 + P_2$ では（体 k における演算として）12 回の乗算と 2 回の 2 乗算を必要とし，点の 2 倍算 $2P_1$ では 5 回の乗算と 6 回の 2 乗算を必要とする [11].

■Jacobi 座標表現　$c = 2, d = 3$ とする．この場合，座標平面上の Weierstrass 方程式 (3.1) で定義される楕円曲線は，射影平面内の方程式

$$E : Y^2 = X^3 + aXZ^4 + bZ^6$$

として表せる．特に，$Z = 0$ を満たす E 上の射影点は $(1 : 1 : 0)$ が代表的で，楕円曲線の無限遠点 ∞ を射影点 $(1 : 1 : 0)$ に対応させておく．これにより，楕円曲線の点の加算が定まる．具体的に，楕円曲線 E 上の 2 つの射影点を $P_1 = (X_1 : Y_1 : Z_1)$ と $P_2 = (X_2 : Y_2 : Z_2)$ とし，その加算点を $P_3 = P_1 + P_2 = (X_3 : Y_3 : Z_3)$ とする．このとき，Weierstrass 方程式で表現された楕円曲線の加算公式 (3.2) と 2 倍算公式 (3.3) より，射影点 $P_3 = P_1 + P_2$ の各射影座標 X_3, Y_3, Z_3 は以下で定まる（ただし，楕円曲線上の射影点 $P = (X : Y : Z)$ の逆元は $-P = (X : -Y : Z)$ となる．）：

- $P_1 \neq \pm P_2$ の場合

$$\begin{cases} r = X_1Z_2^2, \quad s = X_2Z_1^2, \quad t = Y_1Z_2^3, \quad u = Y_2Z_1^3, \quad v = s - r, \quad w = u - t, \\ X_3 = -v^3 - 2rv^2 + w^2, \quad Y_3 = -tv^3 + (rv^2 - X_3)w, \quad Z_3 = vZ_1Z_2. \end{cases}$$

- $P_1 = P_2$ の場合

$$\begin{cases} v = 4X_1Y_1^2, \quad w = 3X_1^2 + aZ_1^4, \\ X_3 = -2v + w^2, \quad Y_3 = -8Y_1^4 + (v - X_3)w, \quad Z_3 = 2Y_1Z_1. \end{cases}$$

- $P_1 = -P_2$ の場合は $P_3 = P_1 + P_2 = (1 : 1 : 0)$ となる.

上述の標準射影座標と同様に，Jacobi 座標による表現では体 k における割り算が一切必要ない．また，計算の順序を工夫することで，Jacobi 座標による点の加算 $P_1 + P_2$ では 11 回の乗算と 5 回の 2 乗算を必要とし，点の 2 倍算 $2P_1$ では 1 回の乗算と 8 回の 2 乗算を必要とする [11].

3.2　Edwards 曲線とその加算公式

標数が 2 以外の体を k とする．xy 座標平面における方程式

$$C : x^2 + y^2 = c^2(1 + dx^2y^2) \quad (c, d \in k,\ cd(c^4d - 1) \neq 0) \tag{3.5}$$

で定義される曲線を（一般化された）**Edwards 曲線**という[*4]．この曲線 C は，変数変換

$$\begin{cases} u = \dfrac{-2c(w - c)}{x^2} \quad (w = (c^2dx^2 - 1)y) \\ v = \dfrac{4c^2(w - c) + 2c(c^4d + 1)x^2}{x^3} \end{cases}$$

を通して，uv 座標平面における Weierstrass 方程式

$$v^2 = (u - c^4d - 1)(u^2 - 4c^4d)$$

で定まる楕円曲線と同型である [56, Proposition 2.18]．（特に，対応する楕円曲線の判別式が 0 以外となるためには条件 $c^4d \neq 1$ が必要である.）この楕円曲線との同型により，曲線 C 上の点集合は群をなす．具体的には，曲線 C 上の点集合における群法則において点 $(0, c)$ は零元であり，曲線 C 上の 2 点 $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ の加算点は

$$P_1 + P_2 = \left(\frac{x_1y_2 + x_2y_1}{c(1 + dx_1x_2y_1y_2)},\ \frac{y_1y_2 - x_1x_2}{c(1 - dx_1x_2y_1y_2)} \right) \tag{3.6}$$

で与えられる．また，曲線 C 上の点 (x, y) の逆元は $(-x, y)$ で与えられる．特に，この加算公式では 2 倍算 $2P$ と異なる 2 点の加算 $P_1 + P_2$ の区別なく計算できる.

[*4] Edwards [20] が $d = 1$ の場合の曲線を紹介した後，その一般形である方程式 (3.5) が [9] で紹介されている.

3.2.1　射影座標表現による加算公式

加算演算 (3.6) を効率的に行うために，3.1.2 節で説明した（標準）射影座標表現を Edwards 曲線 C に適用する．方程式 (3.5) で定義される Edwards 曲線 C 上の点 (x, y) を射影平面 $\mathbb{P}^2$ 内の斉次 4 次曲線

$$(X^2 + Y^2)Z^2 = c^2(Z^4 + dX^2Y^2) \tag{3.7}$$

上の射影点 $(X : Y : Z)$ に対応させる．ただし，任意の $\lambda \neq 0$ に対し $(X : Y : Z) = (\lambda X : \lambda Y : \lambda Z)$ を満たし，$Z \neq 0$ の射影点 $(X : Y : Z)$ を $x = \frac{X}{Z}, y = \frac{Y}{Z}$ に対応させる．このとき，Edwards 曲線上の群の零元は射影点 $(0 : c : 1)$ で，$(X : Y : Z)$ の逆元は $(-X : Y : Z)$ で与えられる．以下に，この射影座標表現による Edwards 曲線上の加算公式の具体的な計算手順を示す [9, Section 4]：

■加算　方程式 (3.7) で定まる射影座標表現による Edwards 曲線上の 2 つの射影点 $(X_1 : Y_1 : Z_1)$, $(X_2 : Y_2 : Z_2)$ の加算点 $(X_3 : Y_3 : Z_3)$ は以下の手順で計算できる：

$$\begin{cases} A = Z_1 \cdot Z_2;\ B = A^2;\ C = X_1 \cdot X_2;\ D = Y_1 \cdot Y_2;\ E = dC \cdot D; \\ F = B - E;\ G = B + E;\ X_3 = A \cdot F \cdot ((X_1 + Y_1) \cdot (X_2 + Y_2) - C - D)\,; \\ Y_3 = A \cdot G \cdot (D - C);\ Z_3 = cF \cdot G. \end{cases} \tag{3.8}$$

体 k 上の 1 回の乗算コストを $\mathbf{M}$，2 乗算コストを $\mathbf{S}$，定数 c, d を掛ける計算コストをそれぞれ $\mathbf{C}, \mathbf{D}$，加減算コストを $\mathbf{add}$ とする．このとき，上記の計算コストは $10\mathbf{M} + 1\mathbf{S} + 1\mathbf{C} + 1\mathbf{D} + 7\mathbf{add}$ である．特に，$X_1Y_2 + X_2Y_1$ は $(X_1 + Y_1)(X_2 + Y_2) - X_1X_2 - Y_1Y_2$ と表し，X_1X_2 や Y_1Y_2 などの項を演算中に使いまわすことで計算の効率化を図っている．

特に $Z_2 = 1$ の場合は一回の乗算 $A = Z_1 \cdot Z_2$ の計算コストを省ける．ゆえに，この場合の計算コストは $9\mathbf{M} + 1\mathbf{S} + 1\mathbf{C} + 1\mathbf{D} + 7\mathbf{add}$ である．

■2 倍算　ここでは $(X_1 : Y_1 : Z_1) = (X_2 : Y_2 : Z_2)$ の場合を考える．この場合，曲線 C の方程式 (3.5) を利用することで，$c(1 + dx_1^2y_1^2)$ を $(x_1^2 + y_1^2)/c$，$c(1 - dx_1^2y_1^2)$ を $(2c^2 - (x_1^2 + y_1^2))/c$ とそれぞれ表せる．これより，(3.6) による 2 倍算公式は

$$2(x_1, y_1) = \left(\frac{2x_1y_1}{c(1 + dx_1^2y_1^2)},\ \frac{y_1^2 - x_1^2}{c(1 - dx_1^2y_1^2)} \right) = \left(\frac{2x_1y_1c}{x_1^2 + y_1^2},\ \frac{(y_1^2 - x_1^2)c}{2c^2 - (x_1^2 + y_1^2)} \right) \tag{3.9}$$

とかける．これより，方程式 (3.7) で定まる射影座標表現による Edwards 曲線上の射影点 $(X_1 : Y_1 : Z_1)$ の 2 倍点 $(X_3 : Y_3 : Z_3)$ は以下の手順で計算できる：

$$\begin{cases} B = (X_1 + Y_1)^2;\ C = X_1^2;\ D = Y_1^2;\ E = C + D;\ H = (cZ_1)^2; \\ J = E - 2H;\ X_3 = c(B - E) \cdot J;\ Y_3 = cE \cdot (C - D);\ Z_3 = E \cdot J. \end{cases} \tag{3.10}$$

この計算コストは $3\mathbf{M} + 4\mathbf{S} + 3\mathbf{C} + 6\mathbf{add}$ である．（ただし，$2H$ は $H + H$ と計算する．）特に，$2X_1Y_1$ を $(X_1 + Y_1)^2 - X_1^2 - Y_1^2$ と表し，$(X_1 + Y_1)^2$，X_1^2 や Y_1^2 の項を演算中に使いまわすことで計算の効率化を図っている．

表 3.1　様々な座標表現による楕円曲線上の加算と 2 倍算の計算コスト比較 [9, Section 5]
（各座標表現による具体的な計算手順とその計算コストについては [11] を参照）

座標表現	加算		加算 ($Z_2 = 1$ の時)		2 倍算	
	計算コスト	$\mathbf{S}/\mathbf{M}=0.8$	計算コスト	$\mathbf{S}/\mathbf{M}=0.8$	計算コスト	$\mathbf{S}/\mathbf{M}=0.8$
(1)	$12\mathbf{M}+2\mathbf{S}$	$13.6\mathbf{M}$	$9\mathbf{M}+2\mathbf{S}$	$10.6\mathbf{M}$	$5\mathbf{M}+6\mathbf{S}$	$9.8\mathbf{M}$
(2)	$11\mathbf{M}+5\mathbf{S}$	$15\mathbf{M}$	$7\mathbf{M}+4\mathbf{S}$	$10.2\mathbf{M}$	$1\mathbf{M}+8\mathbf{S}$	$7.4\mathbf{M}$
(3)	$10\mathbf{M}+3\mathbf{S}$	$12.4\mathbf{M}$	$8\mathbf{M}+3\mathbf{S}$	$10.4\mathbf{M}$	$2\mathbf{M}+6\mathbf{S}$	$6.8\mathbf{M}$
(4)	$13\mathbf{M}+2\mathbf{S}$	$14.6\mathbf{M}$	$11\mathbf{M}+2\mathbf{S}$	$12.6\mathbf{M}$	$3\mathbf{M}+4\mathbf{S}$	$6.2\mathbf{M}$
(5)	$12\mathbf{M}$	$12\mathbf{M}$	$10\mathbf{M}$	$10\mathbf{M}$	$7\mathbf{M}+1\mathbf{S}$	$7.8\mathbf{M}$
(6)	$12\mathbf{M}+5\mathbf{S}$	$16\mathbf{M}$	$8\mathbf{M}+4\mathbf{S}$	$11.2\mathbf{M}$	$2\mathbf{M}+5\mathbf{S}$	$6\mathbf{M}$
(7)	$11\mathbf{M}+6\mathbf{S}$	$15.8\mathbf{M}$	$7\mathbf{M}+4\mathbf{S}$	$10.2\mathbf{M}$	$2\mathbf{M}+7\mathbf{S}$	$7.6\mathbf{M}$
(8)	$10\mathbf{M}+1\mathbf{S}$	$10.8\mathbf{M}$	$9\mathbf{M}+1\mathbf{S}$	$9.8\mathbf{M}$	$3\mathbf{M}+4\mathbf{S}$	$6.2\mathbf{M}$

(1) **Projective:** 無限遠点 ∞ が零元の楕円曲線 $y^2 = x^3 + ax + b$ 上の点 (x, y) を $Y^2Z = X^3 + aXZ^2 + bZ^3$ 上の点 $(X : Y : Z)$ に対応させる．任意の $\lambda \neq 0$ に対し $(X : Y : Z) = (\lambda X : \lambda Y : \lambda Z)$ を満たし，$Z \neq 0$ で $x = \frac{X}{Z}$, $y = \frac{Y}{Z}$ に対応させる．（3.1.2 節を参照）

(2) **Jacobi:** 無限遠点 ∞ が零元の楕円曲線 $y^2 = x^3 + ax + b$ 上の点 (x, y) を $Y^2 = X^3 + aXZ^4 + bZ^6$ 上の点 $(X : Y : Z)$ に対応．任意の $\lambda \neq 0$ に対し $(X : Y : Z) = (\lambda^2 X : \lambda^3 Y : \lambda Z)$ を満たし，$Z \neq 0$ において $x = \frac{X}{Z^2}$, $y = \frac{Y}{Z^3}$ に対応させる．（3.1.2 節を参照）

(3) **Jacobi quartic:** 点 $(0, 1)$ が零元の楕円曲線 $y^2 = x^4 + 2ax^2 + 1$ 上の点 (x, y) を $Y^2 = X^4 + 2aX^2Z^2 + Z^4$ 上の点 $(X : Y : Z)$ に対応させる．任意の $\lambda \neq 0$ に対し $(X : Y : Z) = (\lambda X : \lambda^2 Y : \lambda Z)$ を満たし，$Z \neq 0$ で $x = \frac{X}{Z}, y = \frac{Y}{Z^2}$ に対応．

(4) **Jacobi intersection:** 点 $(0, 1, 1)$ が零元の楕円曲線 $s^2 + c^2 = 1$, $as^2 + d^2 = 1$ 上の点 (s, c, d) を $S^2 + C^2 = Z^2$, $aS^2 + D^2 = Z^2$ 上の点 $(S : C : D : Z)$ に対応にさせる．任意の $\lambda \neq 0$ に対し $(S : C : D : Z) = (\lambda S : \lambda C : \lambda D : \lambda Z)$ を満たし，$Z \neq 0$ において $s = \frac{S}{Z}$, $c = \frac{C}{Z}$, $d = \frac{D}{Z}$ に対応させる．

(5) **Hessian:** 無限遠点 ∞ が零元の楕円曲線 $x^3 + y^3 + 1 = 3axy$ 上の点 (x, y) を $X^3 + Y^3 + Z^3 = 3aXYZ$ 上の点 $(X : Y : Z)$ に対応させる．任意の $\lambda \neq 0$ に対し $(X : Y : Z) = (\lambda X : \lambda Y : \lambda Z)$ を満たし，$Z \neq 0$ において $x = \frac{X}{Z}$, $y = \frac{Y}{Z}$ に対応．

(6) **Doubling-oriented Doche/Icart/Kohel:** 無限遠点 ∞ が零元の楕円曲線 $y^2 = x^3 + ax^2 + 16ax$ 上の点 (x, y) を $Y^2 = X^3Z + aX^2Z^2 + 16aXZ^3$ 上の点 $(X : Y : Z : Z^2)$ に対応させる．任意の $\lambda \neq 0$ に対し $(X : Y : Z : Z^2) = (\lambda X : \lambda^2 Y : \lambda Z : \lambda^2 Z^2)$ を満たし，$Z \neq 0$ において $x = \frac{X}{Z}$, $y = \frac{Y}{Z^2}$ に対応させる．

(7) **Tripling-oriented Doche/Icart/Kohel:** 無限遠点 ∞ が零元の楕円曲線 $y^2 = x^3 + 3a(x + 1)^2$ 上の点 (x, y) を $Y^2 = X^3Z + 3a(X + Z)^2Z^2$ 上の点 $(X : Y : Z : Z^2)$ に対応させる．任意の $\lambda \neq 0$ に対し $(X : Y : Z : Z^2) = (\lambda^2 X : \lambda^3 Y : \lambda Z : \lambda^2 Z^2)$ を満たし，$Z \neq 0$ において $x = \frac{X}{Z^2}$, $y = \frac{Y}{Z^3}$ に対応させる．

(8) **Edwards:** 射影座標表現による $c = 1$ の Edwards 曲線上の加算公式．（3.2.1 節を参照）

■他の座標表現による楕円曲線上の加算コストとの比較　表 3.1 に，上述した射影座標表現による Edwards 曲線上の加算 (3.8)・2 倍算 (3.10) と他の座標表現による様々な加算公式の計算コストの比較表を示す．（詳細は [9, Section 5] を参照．また，各座標表現による具体的な計算手順とその計算コストについては [11] を参照.）今回の比較では，計算コストが小さい定数 c, d との掛け算コスト $\mathbf{C}, \mathbf{D}$ や体 k 上の加減算コスト **add** は無視し，乗算コスト $\mathbf{M}$ と 2 乗算コスト $\mathbf{S}$ のみで比較する．表 3.1 から，2 乗算と乗算のコスト比率が $\mathbf{S}/\mathbf{M} = 0.8$ の時，他の座標表現に比べて射影座標表現による Edwards 曲線上の加算の計算コストが最も小さいことが分かる．また，2 倍算に関しても Edwards 曲線上の計算コストはかなり小さいことが分かる（ただし，最も小さいわけではない）．このように計算コストの観点から Edwards 曲線の利用は非常に有用である．

3.2.2　反転座標表現による加算公式

簡単のため，これ以降は $c = 1$ の場合の Edwards 曲線

$$E_d : x^2 + y^2 = 1 + dx^2y^2 \quad (d \in k,\ d \neq 0, 1) \tag{3.11}$$

を扱う．ここでは，Edwards 曲線上の点 (x, y) を射影平面 $\mathbb{P}^2$ 内の斉次 4 次曲線

$$(X^2 + Y^2)Z^2 = X^2Y^2 + dZ^4 \tag{3.12}$$

上の射影点 $(X : Y : Z)$ に対応させる．具体的には，任意の $\lambda \neq 0$ に対し $(X : Y : Z) = (\lambda X : \lambda Y : \lambda Z)$ を満たすが，前述の射影座標表現を反転させて，$XYZ \neq 0$ を満たす射影点 $(X : Y : Z)$ を $x = \frac{Z}{X},\ y = \frac{Z}{Y}$ に対応させる．ただし，対応する射影点に関する制限条件 $XYZ \neq 0$ により，$xy = 0$ を満たす Edwards 曲線 (3.11) 上の点 (x, y) はこの反転座標では表現できないことに注意する必要がある．具体的には，$xy = 0$ を満たす Edwards 曲線上の零元 $(0, 1)$，位数 2 の点 $(0, \pm 1)$，位数 4 の点 $(\pm 1, 0)$ は反転座標では表せないため，以下で紹介する反転座標表現における Edwards 曲線上の加算公式ではこれらの特別な点は除外する（詳細は [10, Section 4] を参照）：

■加算　反転座標表現による Edwards 曲線 (3.12) 上の制約条件 $X_iY_iZ_i \neq 0$ $(i = 1, 2)$ を満たす 2 つの射影点を $(X_1 : Y_1 : Z_1), (X_2 : Y_2 : Z_2)$ とし，その加算点を $(X_3 : Y_3 : Z_3)$ とする．このとき，Edwards 曲線上の加算公式 (3.6) と反転座標表現により，

$$\left(\frac{Z_1}{X_1}, \frac{Z_1}{Y_1}\right) + \left(\frac{Z_2}{X_2}, \frac{Z_2}{Y_2}\right) = \left(\frac{(X_2Y_1 + X_1Y_2)Z_1Z_2}{X_1X_2Y_1Y_2 + dZ_1^2Z_2^2},\ \frac{(X_1X_2 - Y_1Y_2)Z_1Z_2}{X_1X_2Y_1Y_2 - dZ_1^2Z_2^2}\right) = \left(\frac{Z_3}{X_3}, \frac{Z_3}{Y_3}\right)$$

が成り立つ．ただし，

$$\begin{cases} X_3 = (X_1X_2 - Y_1Y_2)(X_1X_2Y_1Y_2 + dZ_1^2Z_2^2) \\ Y_3 = (X_2Y_1 + X_1Y_2)(X_1X_2Y_1Y_2 - dZ_1^2Z_2^2) \\ Z_3 = (X_1X_2 - Y_1Y_2)(X_2Y_1 + X_1Y_2)Z_1Z_2 \end{cases}$$

とする．また，反転座標表現による加算点 $(X_3 : Y_3 : Z_3)$ の効率的な計算手順は以下である：

$$\begin{cases} A = Z_1 \cdot Z_2;\ B = dA^2;\ C = X_1 \cdot X_2;\ D = Y_1 \cdot Y_2;\ E = C \cdot D; \\ H = C - D;\ I = (X_1 + Y_1) \cdot (X_2 + Y_2) - C - D; \\ X_3 = (E + B) \cdot H;\ Y_3 = (E - B) \cdot I;\ Z_3 = A \cdot H \cdot I. \end{cases} \tag{3.13}$$

この計算コストは $9\mathbf{M} + 1\mathbf{S} + 1\mathbf{D} + 7\mathbf{add}$ で，前述した射影座標表現による加算の計算コスト $10\mathbf{M} + 1\mathbf{S} + 1\mathbf{D} + 7\mathbf{add}$ より少ない．（ここでは，Edwards 曲線の係数として $c = 1$ と設定しているため，定数 c を掛ける計算コストは $\mathbf{C} = 0$ であることに注意する.）

■2 倍算　反転座標表現による Edwards 曲線 (3.12) 上の制約条件 $X_1Y_1Z_1 \neq 0$ を満たす射影点を $(X_1 : Y_1 : Z_1)$ とし，その 2 倍点を $(X_3 : Y_3 : Z_3)$ とする．このとき，Edwards 曲線上の 2 倍算公式 (3.9) と反転座標表現により，

$$2\left(\frac{Z_1}{X_1}, \frac{Z_1}{Y_1}\right) = \left(\frac{2X_1Y_1}{X_1^2 + Y_1^2},\ \frac{X_1^2 - Y_1^2}{X_1^2 + Y_1^2 - 2dZ_1^2}\right) = \left(\frac{Z_3}{X_3}, \frac{Z_3}{Y_3}\right)$$

が成り立つ．ただし，

$$\begin{cases} X_3 = (X_1^2 + Y_1^2)(X_1^2 - Y_1^2) \\ Y_3 = 2X_1Y_1(X_1^2 + Y_1^2 - 2dZ_1^2) \\ Z_3 = 2X_1Y_1(X_1^2 - Y_1^2) \end{cases}$$

とする．また，反転座標表現による 2 倍点 $(X_3 : Y_3 : Z_3)$ の効率的な計算手順は以下である：

$$\begin{cases} A = X_1^2;\ B = Y_1^2;\ C = A + B;\ D = A - B;\ E = (X_1 + Y_1)^2 - C; \\ Z_3 = D \cdot E;\ X_3 = C \cdot D;\ Y_3 = E \cdot (C - 2dZ_1^2). \end{cases} \tag{3.14}$$

この計算コストは $3\mathbf{M} + 4\mathbf{S} + 1\mathbf{D} + 6\mathbf{add}$ である．（ただし，$2dZ_1^2$ は $dZ_1^2 + dZ_1^2$ と計算する.）前述した射影座標表現による 2 倍算の計算コスト $3\mathbf{M} + 4\mathbf{S} + 6\mathbf{add}$ と比べて，この計算コストは定数 d を掛ける計算コスト $\mathbf{D}$ が 1 回分多いことが分かる．

以上の議論から，射影座標表現に比べて，反転座標表現による Edwards 曲線上の加算は乗算コスト $\mathbf{M}$ が 1 回分少ない一方，2 倍算では定数 d を掛ける計算コスト $\mathbf{D}$ が一回分多いというトレードオフの関係にある（後述の表 3.2 を参照）．

3.2.3　ツイスト Edwards 曲線

方程式 (3.11) で定義される Edwards 曲線の集合を拡張するために，ここでは Edwards 曲線の（2 次）ツイストを紹介する．方程式

$$E_{a,d} : ax^2 + y^2 = 1 + dx^2y^2 \quad (a \neq 0,\ d \neq 0,\ a \neq d) \tag{3.15}$$

で定義される曲線を**ツイスト Edwards 曲線**という [5]．特に，$a = 1$ の場合の曲線 $E_{1,d}$ は通常の Edwards 曲線 (3.11) である．変数変換 $\bar{x} = \sqrt{a}x,\ \bar{y} = y$ により，ツイスト Edwards 曲線 $E_{a,d}$ は

通常の Edwards 曲線 $E_{1,d/a} : \bar{x}^2 + \bar{y}^2 = 1 + (d/a)\bar{x}^2\bar{y}^2$ と拡大体 $k(\sqrt{a})$ 上で同型である[*5]．（特に，元 a が体 k 上平方である場合は，これら 2 つの曲線は体 k 上で同型である.）この同型により，ツイスト Edwards 曲線上の点集合は群をなす[*6]．具体的に，ツイスト Edwards 曲 $E_{a,d}$ 上の加算公式は

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2} \right) \tag{3.16}$$

で与えられる．実際，変数変換 $\bar{x} = \sqrt{a}x,\ \bar{y} = y$ を経由して，通常の Edwards 曲線 $E_{1,d/a}$ 上の加算公式 (3.6) と一致する．以下で，通常の Edwards 曲線と同様に，効率的な加算計算を可能とするツイスト Edwards 曲線上の座標表現をまとめておく（詳細は [5, Section 6] を参照）：

■ツイスト Edwards 曲線における射影座標表現　3.2.1 節と同様に，ツイスト Edwards 曲線 $E_{a,d}$ 上の点 (x, y) を射影平面 $\mathbb{P}^2$ 内の斉次 4 次曲線

$$(aX^2 + Y^2)Z^2 = Z^4 + dX^2Y^2 \tag{3.17}$$

上の射影点 $(X : Y : Z)$ に対応させる．特に，$\lambda \neq 0$ に対し，$(X : Y : Z) = (\lambda X : \lambda Y : \lambda Z)$ を満たし，$Z \neq 0$ で $x = \frac{X}{Z}, y = \frac{Y}{Z}$ に対応させる．この射影座標表現により，ツイスト Edwards 曲線 $E_{a,d}$ 上の加算公式 (3.16) は下記のように計算できる：

- 加算：方程式 (3.17) で定義される射影座標表現によるツイスト Edwards 曲線上の 2 つの射影点 $(X_1 : Y_1 : Z_1)$, $(X_2 : Y_2 : Z_2)$ の加算点を $(X_3 : Y_3 : Z_3)$ とする．このとき，加算点 $(X_3 : Y_3 : Z_3)$ は次の計算手順で求まる：

$$\begin{cases} A = Z_1 \cdot Z_2;\ B = A^2;\ C = X_1 \cdot X_2;\ D = Y_1 \cdot Y_2;\ E = dC \cdot D; \\ F = B - E;\ G = B + E;\ X_3 = A \cdot F \cdot ((X_1 + Y_1) \cdot (X_2 + Y_2) - C - D); \\ Y_3 = A \cdot G \cdot (D - aC);\ Z_3 = F \cdot G. \end{cases} \tag{3.18}$$

  この計算コストは $10\mathbf{M} + 1\mathbf{S} + 2\mathbf{D} + 7\mathbf{add}$ である．（ただし，$2\mathbf{D}$ は定数 a と d を掛ける 2 回の計算コストとする.）
- 2 倍算：射影点 $(X_1 : Y_1 : Z_1)$ の 2 倍点 $(X_3 : Y_3 : Z_3)$ は次の計算手順で求まる：

$$\begin{cases} B = (X_1 + Y_1)^2;\ C = X_1^2;\ D = Y_1^2;\ E = aC;\ F = E + D;\ H = Z_1^2; \\ J = F - 2H;\ X_3 = (B - C - D) \cdot J;\ Y_3 = F \cdot (E - D);\ Z_3 = F \cdot J. \end{cases} \tag{3.19}$$

  この計算コストは $3\mathbf{M} + 4\mathbf{S} + 1\mathbf{D} + 7\mathbf{add}$ である．（ただし，$\mathbf{D}$ は定数 a を掛ける 1 回の計算コストとし，$2H$ は $H + H$ と計算する.）

[*5] さらに，ツイスト Edwards 曲線 $E_{a,d}$ は $E_{d,a}$ の 2 次ツイストである．実際，有理写像 $(x, y) \longmapsto \left(x, \frac{1}{y}\right)$ は $E_{a,d}$ と $E_{d,a}$ の間の双有理同値を与える．より一般には，$d'/a' = a/d$ であれば，$E_{a,d}$ は $E_{a',d'}$ の 2 次ツイストである [5, Section 2].

[*6] ツイスト Edwards 曲線を楕円曲線と見なすとき，その j-不変量は $\dfrac{16(a^2 + 14ad + d^2)^3}{ad(a - d)^4}$ である．

表 3.2　座標表現による通常/ツイスト Edwards 曲線上の加算と 2 倍算の計算コスト比較

Edwards 曲線の種類	座標表現	加算	2 倍算
方程式 (3.11) で定義される通常 Edwards 曲線	射影座標	$10\mathbf{M}+1\mathbf{S}+1\mathbf{D}$ 計算手順 (3.8)	$3\mathbf{M}+4\mathbf{S}$ 計算手順 (3.10)
	反転座標	$9\mathbf{M}+1\mathbf{S}+1\mathbf{D}$ 計算手順 (3.13)	$3\mathbf{M}+4\mathbf{S}+1\mathbf{D}$ 計算手順 (3.14)
方程式 (3.15) で定義されるツイスト Edwards 曲線	射影座標	$10\mathbf{M}+1\mathbf{S}+2\mathbf{D}$ 計算手順 (3.18)	$3\mathbf{M}+4\mathbf{S}+1\mathbf{D}$ 計算手順 (3.19)
	反転座標	$9\mathbf{M}+1\mathbf{S}+2\mathbf{D}$ 計算手順 (3.21)	$3\mathbf{M}+4\mathbf{S}+2\mathbf{D}$ 計算手順 (3.22)

■ツイスト Edwards 曲線における反転座標表現　3.2.2 節と同様に，ツイスト Edwards 曲線 $E_{a,d}$ 上の点 (x,y) を射影平面 $\mathbb{P}^2$ 内の斉次 4 次曲線

$$(X^2+aY^2)Z^2=X^2Y^2+dZ^4 \tag{3.20}$$

上の射影点 $(X:Y:Z)$ に対応させる．上記の射影座標表現とは異なり，$XYZ\neq 0$ を満たす射影点 $(X:Y:Z)$ を $x=\frac{Z}{X}$, $y=\frac{Z}{Y}$ に対応させる．(3.2.2 節で説明したように，下記で紹介する加算公式では $XYZ=0$ を満たす特殊な射影点 $(X:Y:Z)$ は除外する.)

- 加算：方程式 (3.20) で定義される反転座標表現によるツイスト Edwards 曲線上の 2 つの射影点 $(X_1:Y_1:Z_1)$, $(X_2:Y_2:Z_2)$ の加算点を $(X_3:Y_3:Z_3)$ とする．このとき，加算点 $(X_3:Y_3:Z_3)$ は次の計算手順で求まる：

$$\begin{cases} A=Z_1\cdot Z_2;\ B=dA^2;\ C=X_1\cdot X_2;\ D=Y_1\cdot Y_2;\ E=C\cdot D;\\ H=C-aD;\ I=(X_1+Y_1)\cdot(X_2+Y_2)-C-D;\\ X_3=(E+B)\cdot H;\ Y_3=(E-B)\cdot I;\ Z_3=A\cdot H\cdot I. \end{cases} \tag{3.21}$$

  この計算コストは $9\mathbf{M}+1\mathbf{S}+2\mathbf{D}+7\mathbf{add}$ である．(ただし，$2\mathbf{D}$ は定数 a と d を掛ける 2 回の計算コストとする.)
- 2 倍算：射影点 $(X_1:Y_1:Z_1)$ の 2 倍点 $(X_3:Y_3:Z_3)$ は次の計算手順で求まる：

$$\begin{cases} A=X_1^2;\ B=Y_1^2;\ U=aB;\ C=A+U; D=A-U;\\ E=(X_1+Y_1)^2-A-B;\ X_3=C\cdot D;\ Y_3=E\cdot(C-2dZ_1^2); Z_3=D\cdot E. \end{cases} \tag{3.22}$$

  この計算コストは $3\mathbf{M}+4\mathbf{S}+2\mathbf{D}+6\mathbf{add}$ である．(ただし，$2\mathbf{D}$ は定数 a と $2d$ を掛ける 2 回の計算コストとする.)

■座標表現による通常/ツイスト Edwards 曲線上の計算コストの比較　表 3.2 に，座標表現による通常/ツイスト Edwards 曲線上の加算と 2 倍算の計算コストの比較表を示す（[5, Section 7] も参照）. 表 3.2 から通常/ツイストの Edwards 曲線共に，加算は一回の乗算コスト $\mathbf{M}$ の分だけ反転座標表現の方が効率的である一方，2 倍算に関しては一回の定数を掛ける計算コスト $\mathbf{D}$ の分だけ射影座標表現の方が効率的であることが分かる．また，ツイスト Edwards 曲線よりも通常 Edwards 曲線の方が加算・2 倍算共に一回の定数を掛ける計算コスト $\mathbf{D}$ の分だけ効率的であることが分かる．

3.2.4　Montgomery 曲線とツイスト Edwards 曲線との関係

ここでは，楕円曲線の点のスカラー倍算を効率的に計算できる Montgomery 曲線を紹介すると共に，ツイスト Edwards 曲線との関係をまとめる．

■Montgomery 曲線　uv 座標平面における方程式

$$E_{A,B}^{(M)}:\ Bv^2 = u^3 + Au^2 + u \quad (A, B \in k,\ B(A^2-4) \neq 0) \tag{3.23}$$

で定義される曲線を **Montgomery 曲線**という（詳細は [22, Section 9.12.1] や [19, Section 13.2.3] などを参照）．単純な変数変換 $x = Bu,\ y = B^2v$ により，方程式 (3.23) は Weierstrass 方程式

$$y^2 = x^3 + ABx^2 + B^2x$$

に変換できるため, Montgomery 曲線は楕円曲線の一種であることが分かる．（特に, 上記の楕円曲線の判別式が $16B^6(A^2-4)$ であるため, 対応する楕円曲線が特異点を持たないように Montgomery 曲線の係数に関する条件 $B(A^2-4) \neq 0$ が必須である.）ゆえに，対応する楕円曲線の加算公式を用いることで，Montgomery 曲線上の点集合は群をなす．特に，Montgometry 曲線においては点 P のスカラー倍算を効率的に計算できる．（標準）射影座標表現により，$P = (U_1 : V_1 : W_1)$, $nP = (U_n : V_n : W_n)$ と表す．このとき，スカラー倍点 $(n+m)P = nP + mP$ の射影座標 $(U_{n+m} : V_{n+m} : W_{n+m})$ は次の計算式から，逐次的に求めることが可能である（下記の逐次的な計算において V_n の値は一切必要ない）：

- $n \neq m$ の場合

$$\begin{cases} U_{m+n} = W_{m-n}\left((U_m - W_m)(U_n + W_n) + (U_m + W_m)(U_n - W_n)\right)^2, \\ W_{m+n} = U_{m-n}\left((U_m - W_m)(U_n + W_n) - (U_m + W_m)(U_n - W_n)\right)^2. \end{cases}$$

  加減算コストを無視すると，この計算コストは $4\mathbf{M} + 2\mathbf{S}$ である．

- $n = m$ の場合

$$\begin{cases} 4U_nW_n = (U_n + W_n)^2 - (U_n - W_n)^2, \\ U_{2n} = (U_n + W_n)^2(U_n - W_n)^2, \\ W_{2n} = 4U_nW_n\left((U_n - W_n)^2 + ((A+2)/4)(4U_nW_n)\right). \end{cases}$$

  計算手順を工夫すると，この計算コストは $3\mathbf{M} + 2\mathbf{S}$ である（加減算コストは無視する）.

これらの計算式から，次のスカラー倍算における Montgomery 乗法公式を得る [17, Chapter 4]：

$$\begin{cases} U_{2n} = (U_n - W_n)^2(U_n + W_n)^2, \\ W_{2n} = \left((U_n+W_n)^2-(U_n-W_n)^2\right)\left((U_n+W_n)^2+\dfrac{A-2}{4}\left((U_n+W_n)^2-(U_n-W_n)^2\right)\right), \\ U_{2n+1} = \left((U_n - W_n)(U_{n+1} + W_{n+1}) + (U_n + W_n)(U_{n+1} - W_{n+1})\right)^2 W_1, \\ W_{2n+1} = \left((U_n - W_n)(U_{n+1} + W_{n+1}) + (U_n + W_n)(U_{n+1} - W_{n+1})\right)^2 U_1. \end{cases} \tag{3.24}$$

■ツイスト Edwards 曲線との関係　Montgomery 曲線全体とツイスト Edwards 曲線全体との間には次の関係がある [5, Theorem 3.2]（また，[17, Chapter 4] も参照）：

定理 3.2.1. 標数が 2 でない体を k とする．このとき，以下が成り立つ：

(i) すべての k 上のツイスト Edwards 曲線 $E_{a,d}$ は Montgomery 曲線 $E^{(M)}_{A,B}$ に体 k 上双有理同値[*7] である．ただし，対応する Montgomery 曲線 $E^{(M)}_{A,B}$ の定義方程式の係数は

$$A = \frac{2(a+d)}{a-d}, \quad B = \frac{4}{a-d}$$

とする．具体的には，k 上定義された 2 つの有理写像

$$E_{a,d} \longrightarrow E^{(M)}_{A,B}; \quad (x,y) \longmapsto \left(\frac{1+y}{1-y}, \frac{1+y}{(1-y)x}\right)$$
$$E^{(M)}_{A,B} \longrightarrow E_{a,d}; \quad (u,v) \longmapsto \left(\frac{u}{v}, \frac{u-1}{u+1}\right)$$

は双有理同値を与える．

(ii) 逆に，すべての k 上の Montgomery 曲線 $E^{(M)}_{A,B}$ はツイスト Edwards 曲線 $E_{a,d}$ に体 k 上双有理同値である．ただし，対応するツイスト Edwards 曲線 $E_{a,d}$ の定義方程式の係数は

$$a = \frac{A+2}{B}, \quad d = \frac{A-2}{B}$$

とする．（具体的には，上記 (i) と同様の有理写像により双有理同値となる．）

3.2.2 節で紹介したように，すべての（ツイスト）Edwards 曲線は位数 4 のねじれ点を持つ．（ツイスト Edwards 曲線との双有理同値性から，すべての Montgomery 曲線も位数 4 のねじれ点を持つ．）一方で，位数 4 のねじれ点を持つ楕円曲線に関して以下が知られている [9, Theorem 2.1]：

定理 3.2.2. 標数が 2 でない体 k 上の楕円曲線 E が k-有理な位数 4 のねじれ点を持つとする．このとき，以下が成り立つ：

(i) 通常 Edwards 曲線 $x^2 + y^2 = 1 + dx^2y^2$ が E の 2 次ツイストと k 上双有理同値になるような元 $d \in k \setminus \{0, 1\}$ が存在する．

[*7] 双有理同値とは，ほとんど至るところ定義された 2 つの有理写像が存在し，互いに逆写像の関係にあるときをいう．

(ii) もし楕円曲線 E が k-有理な位数 2 のねじれ点を唯 1 つ持つなら，通常 Edwards 曲線 $x^2 + y^2 = 1 + dx^2y^2$ が E の 2 次ツイストと k 上双有理同値になる非平方元 $d \in k$ が存在する．さらに，k が有限体であるとき，通常 Edwards 曲線 $x^2 + y^2 = 1 + dx^2y^2$ が E と k 上双有理同値になる非平方元 $d \in k$ が存在する．

第 4 章

楕円曲線暗号に対する攻撃法調査

楕円曲線暗号の安全性を支える**楕円曲線離散対数問題（Elliptic Curve Discrete Logarithm Problem, ECDLP）**とは次の計算問題である：「有限体 $\mathbb{F}_q$ 上の楕円曲線 E とその曲線上の 2 点 $S, T \in E(\mathbb{F}_q)$ が与えられたとき，$T = dS$ を満たす整数 d を見つけよ．ただし，基点 S の位数は素数 r とし[*1]，目標点 T は基点 S が生成する巡回群 $\langle S \rangle$ の元とする.」特に，法 r において ECDLP の解 d は一意的に定まる．また，暗号の安全性のため，楕円曲線暗号では基点 S の位数 r が巨大な素数となるような曲線パラメータを用いる．（具体的には，条件 $r \approx q$ を満たす曲線パラメータを利用する.）さらに，楕円曲線暗号では，標数が巨大な素数 p の素体 $\mathbb{F}_p$ または標数 2 の拡大体 $\mathbb{F}_{2^n}$ 上定義された楕円曲線がよく利用される．特に，標数 2 の素数次拡大体 $\mathbb{F}_{2^n}$ 上定義される Koblitz 曲線 $y^2 + xy = x^3 + ax^2 + 1$ $(a \in \mathbb{F}_2)$ も利用されることがある[*2]．本章では，EdSA で利用される曲線の攻撃評価のため，主に素体上の ECDLP に対する攻撃手法を調査する.

4.1 一般の楕円曲線に適用可能な攻撃法

本節では，一般の楕円曲線上の ECDLP に適用可能な攻撃手法についてまとめる．（本節の内容をまとめるに際し，ECDLP に対する攻撃法に関するサーベイ論文 [23] を主に参考にした.）

4.1.1 Shanks の Baby-Step Giant-Step（BSGS）法

BSGS 法の基本的な計算手順

BSGS 法の基本的なアイデアは，楕円曲線 E 上の基点 S の位数 r に対し $m = \lceil \sqrt{r} \rceil$ とおくことにより，ECDLP の解を $d = d_0 + d_1 m$ $(0 \leq d_0, d_1 < m)$ と一意的に表すことである [49] [*3]．さ

[*1] 基点 S の位数 r が合成数の場合は Pohlig-Hellman 法 [40] により r の冪を含めた各素因子に関する ECDLP に帰着できるため，簡単のため本章では基点 S の位数 r は素数とする．(Pohlig-Hellman 法の詳細については [27, Section 4.1.1] や [15, Section V.1] を参照.)

[*2] 楕円曲線暗号の安全性検証のため，Certicom ECC Challenge [45] では素体・2 冪体上の楕円曲線に加え Koblitz 曲線に関する 3 種類の ECDLP インスタンス $(E/\mathbb{F}_q, S, T, r)$ を公開している．(また，各 ECDLP インスタンスに対して，初めて求解された際の懸賞金が用意されている.)

[*3] 整数 d を m で割った際の商が d_1 で余りが d_0 とみなせ，m の選択の仕方から整数の組 (d_0, d_1) は一意的である.

らに，整数の組 (d_0, d_1) を探索するために，以下の計算手順を行う：

1. まず，楕円曲線 E 上の点 a_0S $(a_0 = 0, 1, 2, \ldots m-1)$ を計算し（この計算を “baby steps” と呼ぶ），それらすべての点を以下の集合リスト L に格納する：

$$L = \{a_0S \in E(\mathbb{F}_q) : a_0 = 0, 1, 2, \ldots, m-1\}$$

2. 次に，基点 S の m 倍点 $R = mS \in E(\mathbb{F}_q)$ を計算し，楕円曲線 E 上の点 $T - a_1R$ $(a_1 = 0, 1, 2, \ldots, m-1)$ を順次計算しながら（この計算を “giant steps”と呼ぶ），リスト L に含まれるか確認する．実際，$T = (d_0 + d_1m)S$ より，リスト L に含まれる点 d_0S が点 $T - d_1R$ に一致するので，$d = d_0 + d_1m$ を満たす組 $(d_0.d_1)$ を探索することができる．

この計算手順では，$O(\sqrt{r})$ 回の楕円曲線 E 上の演算を必要とし，$O(\sqrt{r})$ 個の楕円曲線の点を格納する必要がある．Pollard による逐次点に関する生成法 [42, Section 3] により BSGS 法の平均計算量を削減でき，その平均計算量は $\frac{4}{3}\sqrt{r}$ 回の楕円曲線上の演算が必要である．

逆元計算による BSGS 法の高速化

楕円曲線上の点の逆元計算を利用して BSGS 法を改良することができる．具体的には，無限遠点 ∞ ではない楕円曲線上の点が $P = (x, y)$ で表される場合，その逆元は $-P = (x, -y)$ と簡単に計算できることと，2 点 $\pm P$ の x 座標が同じであることを利用する．より具体的には，$m' = \lceil\sqrt{2r}\rceil$ とおき，求めたい ECDLP の解を $d = d'_0 + d'_1m'$ $(-m'/2 \le d'_0 \le m'/2,\ 0 \le d'_1 < r/m' \approx \sqrt{r/2} \approx m'/2)$ と一意的に表す．整数の組 (d'_0, d'_1) を探索するために，以下の計算手順を行う：

1. まず，楕円曲線 E 上の a_0S の形の点の x 座標の集合リスト

$$L' = \{x(a_0S)\ :\ a_0 = 0, 1, 2, \ldots, \lfloor m'/2 \rfloor\}$$

を作成する．ただし，無限遠点 ∞ では楕円曲線上の点 P の x 座標を $x(P)$ と表す．今回求めたい整数 d'_0 は区間 $[-m'/2, m'/2]$ に含まれるが，リスト L' には非負整数 a_0 に関するスカラー倍点 a_0S $(a_0 = 0, 1, 2, \ldots)$ の x 座標の情報だけを格納する．これにより，今回のリスト L' のサイズがおよそ $m'/2 \approx \sqrt{r/2}$ となり，上記で説明したリスト L のサイズより $\sqrt{2}$ 倍程度小さくなっていることに注意する（つまり，$\#L \approx \sqrt{r}, \#L' \approx \sqrt{r/2}$ である）．
2. 次に, 上記で説明した基本的な BSGS 法と同じように, 基点 S の m' 倍点 $R' = m'S \in E(\mathbb{F}_q)$ を計算し，楕円曲線 E 上の点 $T - a_1R'$ $(a_1 = 0, 1, 2, \ldots, \lfloor r/m' \rfloor)$ を順次計算しながら，各点の x 座標がリスト L' に含まれるか確認する．実際, $T = (d'_0 + d'_1m')S$ より，点 $T - d'_1R'$ の x 座標はリスト L' に含まれ $x(d'_0S) \in L'$ に一致する．これより，整数の組 (d'_0, d'_1) を探索することができる．

この改良版の BSGS 法で必要な楕円曲線上の平均演算回数は $\frac{4}{3}\sqrt{r/2} \approx 0.943\sqrt{r}$ で, 通常の BSGS 法に比べて $\sqrt{2}$ 倍の高速化が実現できる．（BSGS 法タイプのアルゴリズムに関するより詳細な計算量や改良方式については，[23, Section 4] を参照.）

4.1.2　Pollard の ρ 法

Pollard の ρ 法は ECDLP を含む一般の離散対数問題に対する攻撃法である [41]．上記で説明した BSGS 法は膨大なデータを格納する必要があるのに対し（$O(\sqrt{r})$ の空間計算量），ρ 法では非常に少ないデータの格納だけで ECDLP を求解できる．さらに，大規模な並列化計算も比較的簡単に実現できるため，100 ビット以上の巨大な位数 r を持つ ECDLP の求解にはすべて並列化 ρ 法が利用されている．実際，現時点での各種曲線における ECDLP 求解の世界記録は以下で，すべて並列化 ρ 法により解かれている（特に，ECDLP の解読計算量は位数 r の大きさに依存する）：

- 素体においては，112 ビット素数 p の有限体 $\mathbb{F}_p$ 上定義された 111.78 ビットの位数 r を持つ ECDLP インスタンスが 2009 年に解かれた [16] [*4]．（一般に楕円曲線暗号で利用される曲線ではないが，114 ビット素体上で 113.20 ビットの位数 r を持つペアリング暗号で利用される Barreto-Naehrig 曲線に関する ECDLP インスタンスが 2017 年に解かれている [34].）
- 2 冪体においては，127 ビットの拡大体 $\mathbb{F}_{2^{127}}$ 上定義された 117.35 ビットの位数 r を持つ ECDLP インスタンスが 2016 年に解かれた [7],
- 2 冪体上の Koblitz 曲線においては，113 ビットの拡大体 $\mathbb{F}_{2^{113}}$ 上定義された 112.00 ビットの位数 r を持つ Koblitz 曲線 $y^2 + xy = x^3 + x^2 + 1$ に関する ECDLP インスタンスが 2014 年に解かれた [57]．（解読されたわけではないが，Certicom ECC Challenge [45] で公開されている 131 ビットの拡大体 $\mathbb{F}_{2^{131}}$ 上定義された Koblitz 曲線に関する ECDLP インスタンス ECC2K-130 の求解に必要な計算量が [3] で見積もられている.）

反復関数の選択

ECDLP インスタンス $(E/\mathbb{F}_q, S, T, r)$ に対し，ρ 法では巡回群 $\langle S\rangle$ 上の自己写像 $f : \langle S\rangle \longrightarrow \langle S\rangle$ を $f(X) = X + aS + bT,\ X \in \langle S\rangle$ となる整数 a, b が効率的に計算できる関数を固定する．この f を反復関数（iteration function）と呼ぶ．例えば，自然数 L に対し，適当な写像 $H : \langle S\rangle \longrightarrow \{1, 2, \ldots, L\}$ をとる[*5]．ここでは，L を分割数，H を分割関数と呼ぶ．次に，基点 S と目標点 T を用いて，L 個の組 $(R_i, a_i, b_i),\ i = 1, 2, \ldots, L$ を用意する．ただし，$a_i, b_i \in [0, r)$ はランダム整数とし，$R_i = a_iS + b_iT$ を楕円曲線 E 上の点とする．このとき，任意の点 $X \in \langle S\rangle$ に対し，

$$f(X) = X + R_i, \quad i = H(X) \tag{4.1}$$

と定めると反復関数が構成できる[*6]（その他の関数は [53, 54] を参照）．一方，Koblitz 曲線上の ECDLP に対しては，後述の自己同型写像による高速化に適した反復関数を選択する必要がある[*7]．

[*4] 具体的には，“secp112r1”[43] と呼ばれる楕円曲線に関する ECDLP インスタンスが，ゲーム機 PlayStation3 による計算機クラスターにより求解された．(SEC2 [43] を含めた様々な楕円曲線パラメータについては [12] を参照.)

[*5] 具体的には，点 $P \in \langle S\rangle$ に対し $H(P) = x(P) \pmod{L} + 1$ と定めれば良い.

[*6] 構成の仕方から，$f(X) = X + R_i = X + a_iS + b_iT$ を満たす整数の組 (a_i, b_i) は既知である.

[*7] Koblitz 曲線上の ECDLP に適した反復関数の構成は少し複雑で，詳しくは [24, 57, 59] を参照.

ρ 法の基本的な計算手順

反復関数 f を用いて，ρ 法では次の計算手順で ECDLP を求解する：

1. ランダムに 2 つの整数 $u_0, v_0 \in [0, r)$ をとり，初期点 $X_0 = u_0S + v_0T \in \langle S \rangle$ を生成する．
2. 次に，$X_{i+1} = f(X_i)$ $(i = 0, 1, 2, \dots)$ により逐次的に巡回群 $\langle S \rangle$ の元 X_i を生成していく．特に，反復関数の性質から，$X_i = u_iS + v_iT$ を満たす整数の組 (u_i, v_i) が効率的に計算できる．（実際，初期点 X_0 が $u_0S + v_0T$ という基点 S と目標点 T の線形和の形なので，反復関数 f の性質から X_i も $u_iS + v_iT$ という点 S, T の線形和の形で表せる．）
3. 巡回群 $\langle S \rangle$ の位数は r（つまり有限集合）だから，点 X_i を順に計算していくと，やがて既に得られた点と等しくなる．（この現象を**衝突**と呼ぶ[*8]．）衝突した点 $X_i = X_j$ $(i \neq j)$ を考えると，関係式 $u_iS + v_iT = u_jS + v_jT$ が成り立つ．これより，$v_i \not\equiv v_j \pmod r$ なら[*9]，

$$d \equiv (u_i - u_j) \cdot (v_j - v_i)^{-1} \pmod r$$

となる整数 d は ECDLP の解である．

ρ 法の計算量と並列化

反復関数 f が巡回群 $\langle S \rangle$ 上でランダムな振る舞いをする場合[*10]，衝突が起こるまでに必要な点列 $\{X_0, X_1, X_2, \dots\}$ の個数の期待値は誕生日の逆理（birthday paradox）より，

$$\sqrt{\frac{\pi r}{2}} \approx 1.2533\sqrt{r} \tag{4.2}$$

である（証明については，例えば [51, Section XI.5] を参照）．よって，ρ 法を用いて ECDLP を解くのにかかる計算時間の期待値は

$$\sqrt{\frac{\pi r}{2}} \cdot t(f) \tag{4.3}$$

である．ただし，$t(f)$ は反復関数 f の計算時間とする．（式 (4.1) の反復関数を用いた場合，$t(f)$ は楕円曲線上の点の加算の計算時間である．）また，ρ 法は比較的簡単に並列化でき，M 台の計算機を用いれば M 倍の高速化を実現することができる [55]．具体的には，M 台の計算機が独立に初期点 $X_0^{(k)}$, $k = 1, 2, \dots, M$ を生成し，同じ反復関数 f を用いて点列 $\{X_0^{(k)}, X_1^{(k)}, X_2^{(k)}, \dots\}$ を生成していく．このとき，M 台の計算機が独立に生成する点列 $\{X_0^{(k)}, X_1^{(k)}, X_2^{(k)}, \dots\}$, $k = 1, 2, \dots, M$ において衝突を探索することで，ECDLP の解が得られる．

[*8] ちなみに，「ρ 法」という名前は，衝突を引き起こす点列 $\{X_0, X_1, \dots\}$ がギリシャ文字の “ρ” の形のように点在することに由来している．（例えば [27, Figure 4.1] の図を参照．）

[*9] 点列 $\{X_0, X_1, X_2, \dots\}$ が巡回群 $\langle S \rangle$ 上ほぼランダムに点在する限り，$v_i \equiv v_j$ $(i \neq j)$ となることは実用上ほぼない．つまり，$v_i \equiv v_j$ $(i \neq j)$ となる自明な衝突の発生確率は実用上無視できる．

[*10] 正確には，反復関数 f によって生成される点列 $\{X_0, X_1, X_2, \dots\}$ が巡回群 $\langle S \rangle$ 上ランダムに点在する場合を指す．また，式 (4.1) の反復関数 f を利用する場合は，分割数 L を 16 以上に設定すると f がほぼランダムに振る舞うことが実験的に示されている [54]．（また，[60, Section 3.2] の実験報告も参照．）

代表的な ρ 法の改良法

ここでは，ECDLP 求解における ρ 法の改良方法を 2 つ紹介する：

■識別点の導入と衝突探索　識別点（distinguished points）を導入することで，衝突を探索するための空間計算量を劇的に削減できる．まず，楕円曲線の点が識別点であるかの基準を定める．（例えば，楕円曲線の点の x 座標の最初の t ビットがすべて 0 であるものを識別点と定める.）複数の計算機が独立に点列を生成する際，識別点が現れたら，その点を衝突探索サーバに送信する．衝突探索サーバ上では識別点だけを格納しており，識別点間の衝突を探索する．巡回群 $\langle S\rangle$ における識別点の割合を $0 < \theta < 1$ とする．M 台の計算機で ρ 法を計算する際，識別点での衝突が起こるまでに各計算機が生成する点列 $\{X_0^{(k)}, X_1^{(k)}, X_2^{(k)}, \dots\}$ の個数の期待値は

$$\frac{1}{M}\sqrt{\frac{\pi r}{2}} + \frac{1}{\theta}$$

である．特に，$\frac{1}{\theta}$ は識別点とは限らない衝突が起こった後に識別点での衝突が起こるまでに必要な点列の個数の期待値である．

■自己同型写像による高速化　楕円曲線上の自己同型写像による ρ 法の高速化が [24, 58] により提案されている（[27, Section 4.1.2] も参照）．効率的に計算可能な巡回群 $\langle S\rangle$ 上の位数 t の自己同型写像 ψ が存在すると仮定する[*11]．ここで，巡回群 $\langle S\rangle$ 上の同値関係を $P \sim Q \iff P = \psi^j(Q)$ $(\exists j \in [0, t))$ と定める．この同値類による集合全体を $\langle S\rangle/\sim$ とし，$[P]$ を点 P を含む同値類とする．自己同型写像 ψ による ρ 法の高速化の基本的なアイデアは，同値類による集合 $\langle S\rangle/\sim$ 上の反復関数を構成することである．具体的には，巡回群 $\langle S\rangle$ 上の反復関数 f に対して，$\bar{f}([P]) = f([P])$ と定めることで同値類による集合 $\langle S\rangle/\sim$ 上の反復関数 $\bar{f}$ を構成する．このとき，$\#\left(\langle S\rangle/\sim\right) \approx r/t$ より，反復関数 $\bar{f}$ で生成された点列間で衝突が起こるまでに必要な点の個数の期待値はおよそ $\sqrt{\frac{\pi r}{2t}}$ である．つまり，位数 t の自己同型写像 ψ による ρ 法の高速化により，およそ $\sqrt{t}$ 倍少ない個数で衝突を引き起こすことが期待できる．

- すべての楕円曲線上では逆元計算による位数 2 の自己同型写像 ψ が存在する（つまり，$\psi(P) = -P$）．自己同型写像 ψ は効率的に計算できるため，ρ 法を $\sqrt{2}$ 倍高速化できる[*12]．
- Koblitz 曲線 E 上では Frobenius 写像

  $$\phi : E(\mathbb{F}_{2^n}) \longrightarrow E(\mathbb{F}_{2^n}), \quad (x, y) \longmapsto (x^2, y^2), \ \infty \longmapsto \infty$$

  による位数 n の自己同型写像 ϕ があり，効率的に計算することができる．上記に説明した逆元計算による位数 2 の自己同型写像 ϕ との合成写像を考えることで，Koblitz 曲線上では位数 $2n$ の自己同型写像が考えることができるので，およそ $\sqrt{2n}$ 倍の高速化が可能となる．

[*11] 自己同型写像 ψ の位数とは楕円曲線上の自己同型写像全体がなす群 $\mathrm{Aut}(E)$ の元としての位数を指す．

[*12] しかし，実用的に $\sqrt{2}$ 倍の高速化を実現するには，適切な反復関数 f の選択や自明な衝突の回避対処など様々な工夫が必要である [14].

注意 4.1.1. 反復関数の構成方法から，ρ 法は楕円曲線上の加算の繰り返しが本質的演算であり，加算演算操作自体には多くのメモリーを必要としないことに加え，衝突探索部は識別点を利用することで省メモリ化が可能である．このことから，ρ 法は大規模分散計算に向けた実装が可能な攻撃アルゴリズムであり，これまでの ECDLP の解読実験のようにインターネット上の有志による不統一な計算環境でも，比較的うまく機能させることができる．ただし，ρ 法を用いて ECDLP を求解する際には，反復関数の選択や数多くの実装パラメータに依存して，解読計算量が変化する可能性がある（具体的なアルゴリズム並びに理論検討・実験評価の詳細については [60, 63] を参照）．

4.1.3 指数計算法（Index Calculus Method）

指数計算法は，ECDLP に限らず一般の離散対数問題（Discrete Logarithm Problem, DLP）を効率的に求解する方法である．以下では，まず ECDLP を指数計算法で求解するために必要な Semaev の summation 多項式について説明する．

Semaev の summation 多項式

Semaev の summation 多項式は楕円曲線に付随する多変数多項式で，具体的には次のように構成する [48]．標数が 2, 3 以外の体 k 上の楕円曲線 $E : y^2 = x^3 + ax + b$ において，$m = 2, 3$ における summation 多項式 S_m を

$$\begin{aligned} S_2(x_1, x_2) =& x_1 - x_2 \\ S_3(x_1, x_2, x_3) =& (x_1 - x_2)^2 x_3^2 - 2\left\{(x_1 + x_2)(x_1x_2 + a) + 2b\right\} x_3 \\ &+ \left\{(x_1x_2 - a)^2 - 4bx_1x_2\right\} \end{aligned}$$

と定める．また，$m \geq 4$ における summation 多項式は

$$S_m(x_1, \ldots, x_m) = \mathrm{Res}_x\left(S_{m-j}(x_1, \ldots, x_{m-j-1}, x), S_{j+2}(x_{m-j}, x_{m-j+1}, \ldots, x_m, x)\right)$$

と再帰的に定める．ただし，Res_x は変数 x における終結式とし，j を $1 \leq j \leq m-3$ を満たす整数とする．任意の $m \geq 2$ に対し，$S_m(x_1, x_2, \ldots, x_m)$ は既約かつ対称な多変数多項式で，各変数において 2^{m-2} の次数を持つ．また，summation 多項式は次の性質を持つ[*13]：体 k の代数閉体 $\bar{k}$ における m 個の元 $\alpha_1, \ldots, \alpha_m$ に対して，$S_m(\alpha_1, \ldots, \alpha_m) = 0$ であることと，$P_i = (\alpha_i, \beta_i) \in E(\bar{k})$ $(1 \leq i \leq m)$ かつ $P_1 + \cdots + P_m = \infty$ を満たす m 個の元 $\beta_1, \ldots, \beta_m \in \bar{k}$ が存在することは同値である[*14]．

ECDLP に対する指数計算法の計算手順

ECDLP インスタンス $(E/\mathbb{F}_q, S, T, r)$ に対する指数計算法の基本的な計算手順は以下である（ただし，簡単のため $E(\mathbb{F}_q) = \langle S \rangle$ と仮定する）：

[*13] 逆に，この性質を持つように summation 多項式は構成されている．

[*14] 特に，summation 多項式の零点 $(\alpha_1, \ldots, \alpha_m) \in \bar{k}^m$ は $P_1 + \cdots + P_m = \infty$ を満たす点 P_i の x-座標を与える．

1.（因子基底の選択ステップ）群 $E(\mathbb{F}_q)$ の部分集合 F を選択する．集合 F は**因子基底**（factor base）と呼ばれる．例えば，ランダムに選択した有限体 $\mathbb{F}_q$ の部分集合 V に対し，V の元を x-座標に持つ楕円曲線 E 上の点集合 $F = \{(x, y) \in E(\mathbb{F}_q) : x \in V\}$ を因子基底と定めればよい．（この他にも様々な定め方があるが，ここでは最も基本的な因子基底を取る．）
2.（関係式の生成ステップ）
   （a）ランダムに 2 つの整数 u, v を選び，楕円曲線上の点 $R = uS + vT \in E(\mathbb{F}_q)$ を計算する．
   （b）次に，点 R を因子基底 F の元の和で分解する．分解に成功した場合，関係式

$$uS + vT = \sum_{P_i \in F} e_i P_i$$

   を満たす整数の組 (u, v) とベクトル (e_i) を関係行列の行として格納する．具体的には，固定した自然数 m に対し，summation 多項式を含めた変数 $x_1, \ldots, x_m$ に関する連立方程式

$$\begin{cases} S_{m+1}(x_1, \ldots, x_m, x(R)) = 0 \\ F(x_1) = 0 \\ \vdots \\ F(x_m) = 0 \end{cases} \tag{4.4}$$

   を考える．ただし，$F(x)$ は集合 V のすべての元を根に持つ一変数多項式

$$F(x) = \prod_{\alpha \in V} (x - \alpha)$$

   とする．この連立方程式を解いて[*15]，解 $(\alpha_1, \ldots, \alpha_m) \in \mathbb{F}_q^m$ が得られた場合，summation 多項式の性質から $P_1 + \cdots + P_m = \pm R$ を満たす因子基底 F の元 $P_1, \ldots, P_m$ が必ず存在する[*16]．（因子基底の取り方により，連立方程式の解を得られないこともある．）
   （c）少なくとも $\#F$ 個の異なる関係式を見つけるまで，この操作を続ける．
3.（線形代数ステップ）上記のステップで構成した関係行列に対する線形代数操作により[*17]，$\lambda S + \mu T = 0$ を満たす等式を見つける．特に，法 r において整数 μ が可逆であれば[*18]，ECDLP の解として

$$d \equiv -\frac{\lambda}{\mu} \pmod{r}$$

   が得られる．

ECDLP に対する指数計算法における研究課題と研究進展

上記の計算手順において，ステップ 2-(b) の点 R の因子基底 F の元の和の分解が最も計算コストが大きく，summation 多項式の対称性や S_3 などの次数が小さい summation 多項式を組み合わ

[*15] 連立方程式を解くのにグレブナー基底計算アルゴリズムを適用するのが一般的である．
[*16] より具体的には，すべての $1 \leq i \leq m$ に対して点 P_i の x-座標 $x(P_i)$ は α_i に一致する．
[*17] 具体的には，関係行列に対して Gauss の消去法を適用すれば良い．
[*18] 可逆でない場合は，別の関係式を生成し，関係行列に追加することで関係行列を構成し直せばよい．

せた連立方程式を利用する改良がある（詳しくは [23, Section 9] を参照）．しかし，ECDLP に対する指数計算法においては，次の 2 点が主な研究課題である：

- 連立方程式を解くためのグレブナー基底計算アルゴリズムの評価・解析が困難なため，指数計算法の計算量見積もりが非常に難しい.（さらに，指数計算法の計算量は因子基底の取り方やそのサイズに依存するが，最良の因子基底の取り方が現状明らかではない.）
- 暗号で利用される実用サイズの ECDLP を効率的に解くアルゴリズムが整備されていない.（実際，ECDLP に対する指数計算法に関しては，小さなサイズの実験データしかない.）

以下で，これらの研究課題に対する近年の研究進展紹介する：

■素体 $\mathbb{F}_p$ の場合　2016 年，Petit ら [39] は因子基底の元による点の分解を効率的に行える素数 p の条件を提案した．具体的には，$p-1$ が滑らかな場合（特に 2 冪の場合），グレブナー基底計算アルゴリズムで点の分解に関する連立方程式を効率的に解けるような因子基底の取り方を提案した．しかしその後，Peit らが提案した指数計算法にとって優位な条件下ですら，20〜30 ビットの小さな標数 p において BSGS 法や ρ 法の方が効率的であることが実験的に示された [33]．また近年では，連立方程式 (4.4) に対するグレブナー基底計算アルゴリズム内の振る舞いを解析することにより，上記で説明した基本的な指数計算法は全数探索法よりも悪い計算量を持つことが示された [61].

■2 冪体の場合　2 冪体上の ECDLP に対しては，Weil descent と呼ばれる技法と組み合わせた指数計算法に関する研究が数多くある（詳しくは [23, 62] を参照[*19]）．しかし，Weil descent を利用した指数計算法よりも，ρ 法の方が計算効率が良いと [62] で結論付けられている．また近年では，合成次数の拡大体の ECDLP に対し，因子基底の選択を工夫することで，全数探索よりも効率的解く方法が提案された [29]．（しかし，これも BSGS 法や ρ 法より効率的ではないことに注意する.）

注意 4.1.2 (Xedni 計算法)**.** 指数計算法の変種として，有限体上の楕円曲線 E と 2 点 S, T をある代数体上の楕円曲線と有理点に持ち上げてから ECDLP を解く Xedni 計算法 [50] がある[*20]．しかし，[30] の解析によると，基礎体の標数のサイズが大きくなるごとに Xedni 計算法は ECDLP の求解に失敗し，実用サイズの ECDLP には適用できないと報告されている．（また，比較的小さな標数でさえ，Xedni 計算法が成功する確率は高くないことが実験的に示されている.）

4.2　特殊な曲線にのみ有効な攻撃法

前節で紹介した攻撃法はすべての楕円曲線に適用可能な攻撃法である一方，本節では特殊な楕円曲線にのみ有効な攻撃法をまとめておく．（本節の内容をまとめるにあたり，[51, Section XI.6] や [56, Sections 5.3, 5.4] などを主に参考にした.）また，本節では素体 $\mathbb{F}_p$ 上の ECDLP に着目する．

[*19] 本章の序文で述べたように，素体上の ECDLP が主な調査対象であるため，2 冪体については詳しくは説明しない.
[*20] ちなみに，“Xedni”は英単語 “Index”のアルファベットを逆から並べた造語である.

4.2.1　Menezes-Okamoto-Vanstone（MOV）攻撃法

MOV 攻撃法は楕円曲線上のペアリングを利用して ECDLP を乗法群上の DLP に帰着させる攻撃法である [36] [*21]．まず，ECDLP の帰着先の乗法群を定める上で重要な役割を果たすパラメータを定義する．ECDLP インスタンス $(E/\mathbb{F}_p, S, T, r)$ に対して，楕円曲線 E における位数 r のねじれ群

$$E[r] = \{P \in E(\overline{\mathbb{F}}_p) \mid rP = \infty\}$$

を考える．条件 $E[r] \subseteq E(\mathbb{F}_{p^m})$ を満たす最小の自然数 m を**埋め込み次数**（embedding degree）と呼ぶ．特に，$r \nmid p-1$ の条件下で[*22]，埋め込み次数 m は条件 $r \mid p^m - 1$ を満たす最小の自然数と一致する（証明については，例えば [51, Lemma 6.2 in Chapter XI] を参照）．

数学的準備

ガロア群 $G = \mathrm{Gal}(\overline{\mathbb{F}}_p/\mathbb{F}_{p^m})$ に対して，G-加群の短完全系列

$$0 \longrightarrow E[r] \longrightarrow E(\overline{\mathbb{F}}_p) \xrightarrow{r} E(\overline{\mathbb{F}}_p) \longrightarrow 0$$

から（上記の “r” は r 倍算を意味する），ガロアコホモロジーによる群の長完全系列

$$\begin{array}{ccccccccc} 0 & \longrightarrow & E[r]^G & \longrightarrow & E(\mathbb{F}_{p^m}) & \xrightarrow{r} & E(\mathbb{F}_{p^m}) & & \\ & \xrightarrow{\delta} & H^1(G, E[r]) & \longrightarrow & H^1(G, E(\overline{\mathbb{F}}_p)) & \xrightarrow{r} & H^1(G, E(\overline{\mathbb{F}}_p)) & \longrightarrow & \cdots \end{array}$$

が得られる．これより，単射準同型写像

$$\delta_E : E(\mathbb{F}_{p^m})/rE(\mathbb{F}_{p^m}) \longrightarrow H^1(G, E[r]) = \mathrm{Hom}(G, E[r])$$

が得られる．ただし，$\delta_E(P) = [G \ni \sigma \longmapsto \sigma(Q) - Q \in E[r]] \in \mathrm{Hom}(G, E[r])$ とする（$P = rQ$ なる点 Q を固定）．さらに，単射準同型写像 δ_E と Weil ペアリング $e_r : E[r] \times E[r] \longrightarrow \mu_r$ を組み合わせて，双線形写像

$$\kappa : E(\mathbb{F}_{p^m})/rE(\mathbb{F}_{p^m}) \times E[r] \longrightarrow \mathrm{Hom}(G, \mu_r) \simeq \mathbb{F}_{p^m}^{\times}/(\mathbb{F}_{p^m}^{\times})^r$$

が得られる．ただし，μ_r を 1 の r 乗根の集合 $\left\{\alpha \in \overline{\mathbb{F}}_p : \alpha^r = 1\right\}$ とし，

$$\kappa(P, Q) = [G \ni \sigma \longmapsto e_r(\delta_E(P), Q) \in \mu_r] \in \mathrm{Hom}(G, \mu_r)$$

とする．この双線形写像 κ は Tate-Lichtenbaum ペアリングと呼ばれる（詳細は [51, Section XI.9] を参照）．特に，双線形写像 κ は非退化で，Miller アルゴリズムにより効率的に計算可能である（詳細は [19, Chapter 16] を参照）．

[*21] 元々は Menezes-岡本-Vanstone が Weil ペアリングを利用した攻撃法を提案し [36]，その後 Frey-Rück が Tate-Lichitenbaum ペアリングを利用した攻撃法を提案した [21]．ここでは，後者の Tate ペアリング-Lichitenbaum を利用した攻撃法を説明する．

[*22] 楕円曲線暗号における ECDLP では，2 つの巨大素数 p, r は $p \approx r$ を満たすので，$r \mid p-1$ となることはない．

計算手順と適用曲線

上記で構成した非退化な双線形写像 κ を用いて，ECDLP インスタンス $(E/\mathbb{F}_p, S, T, r)$ に対する MOV 攻撃法では以下の計算手順を行う：

1. $E[r] \subseteq E(\mathbb{F}_{p^m})$ を満たす埋め込み次数 m を計算する．
2. $\kappa(P,S) \neq 1$ となる点 $P \in E(\mathbb{F}_{p^m})$ を 1 つ固定する[*23]．
3. $\zeta_1 = \kappa(P,S)$ と $\zeta_2 = \kappa(P,T)$ を計算する．
4. 乗法群 $\mathbb{F}_{p^m}^{\times}$ において $\zeta_1^d = \zeta_2$ を満たす整数 d を見つける[*24]．（つまり，乗法群 $\mathbb{F}_{p^m}^{\times}$ における DLP を解く．）

埋め込み次数 m が大きいと，乗法群 $\mathbb{F}_{p^m}^{\times}$ における DLP を求解することが困難になる．ゆえに，埋め込み次数が十分小さい ECDLP インスタンスに限り MOV 攻撃法は有効である．特に，E が超特異楕円曲線のとき，$m = 2$ であり MOV 攻撃法が有効である．

4.2.2　Semaev-Smart-Satoh-Araki（SSSA）攻撃法

有限体 $\mathbb{F}_p$ 上の楕円曲線 E の $\mathbb{F}_p$-有理点全体がなす群の位数 $\#E(\mathbb{F}_p)$ がちょうど p となるとき，楕円曲線 E を **anomalous** と呼ぶ．Semaev [47]，Smart [52]，佐藤-荒木 [46] はほぼ同時期かつ独立に anomalous 楕円曲線上の ECDLP を非常に効率的に求解するアルゴリズムを提案した．ここでは，その攻撃アルゴリズムを SSSA 攻撃法と呼ぶ．SSSA 攻撃法のアイデアは，anomalous 楕円曲線上の ECDLP を加法群 $\mathbb{F}_p^+$ 上の DLP に帰着させることである[*25]．

数学的準備

ここでは，まず anomalous とは限らない有限体 $\mathbb{F}_p$ 上の楕円曲線 E を考える．SSSA 攻撃法では，有限体 $\mathbb{F}_p$ 上の楕円曲線 E の方程式の各係数を p-進数体 $\mathbb{Q}_p$ に持ち上げることで $\mathbb{Q}_p$ 上の楕円曲線 $\widetilde{E}$ を構成する．このとき，群の短完全列

$$0 \longrightarrow \ker\pi \longrightarrow \widetilde{E}(\mathbb{Q}_p) \xrightarrow{\pi} E(\mathbb{F}_p) \longrightarrow 0$$

を考える．ただし，π は還元写像とする．また，$\mathcal{E}$ を $\widetilde{E}$ に付随する形式群とすると，群の同型

$$\ker\pi \simeq \mathcal{E}(p\mathbb{Z}_p) = p\mathbb{Z}_p, \quad (x,y) \longmapsto -\frac{x}{y}$$

が存在する．ここで，任意の持ち上げ写像 $u : E(\mathbb{F}_p) \longrightarrow \widetilde{E}(\mathbb{Q}_p)$ に対して[*26]（この写像は群としてはではなく単なる集合としての写像），合成写像

$$\lambda(u) : E(\mathbb{F}_p) \xrightarrow{u} \widetilde{E}(\mathbb{Q}_p) \xrightarrow{N} \ker\pi \simeq \mathcal{E}(p\mathbb{Z}_p) = p\mathbb{Z}_p \xrightarrow{\bmod p^2} p\mathbb{Z}_p/p^2\mathbb{Z}_p = \mathbb{F}_p^+$$

[*23] 双線形写像 κ の非退化性より，$\kappa(P,S) \neq 1$ を満たす点 P は必ず存在する．

[*24] 写像 κ の双線形性から，$\zeta_1^d = \zeta_2$ を満たす整数 d は ECDLP の解と一致する．

[*25] 記号 $\mathbb{F}_p^+$ は有限体 $\mathbb{F}_p$ を加法群とみなすことを意味する．また，加法群 $\mathbb{F}_p^+$ 上の DLP の求解は非常に簡単である．

[*26] 具体的には，Hensel の補題による持ち上げ写像を構成すれば良い．

を考える．ただし，$N = \#E(\mathbb{F}_p)$ に対し，上記の “N”は $\mathbb{Q}_p$ 上の楕円曲線 $\widetilde{E}$ における N 倍写像とする．特に，楕円曲線 E が anomalous のとき（つまり，$N = p$ のとき），持ち上げ写像 u により定まる写像 $\lambda(u)$ は群の同型写像か零写像である（証明については，例えば [51, Proposition 6.5 in Chapter XI] を参照）．

計算手順

以下に，anomalous 楕円曲線上の ECDLP インスタンス $(E/\mathbb{F}_p, S, T, p)$ に対する SSSA 攻撃法の計算手順を示す[*27]：

1. 写像 $\lambda(u)$ が同型写像となる持ち上げ写像 u を固定する[*28].
2. 加法群 $\mathbb{F}_p^+$ 上の 2 つの元 $s = \lambda(u)(S),\ t = \lambda(u)(T)$ を計算する.
3. 加法群 $\mathbb{F}_p^+$ 上の DLP を解くことにより，$t = ds$ を満たす整数 d を見つける．（写像 $\lambda(u)$ が群の同型写像により，整数 d は ECDLP の解と一致する.）

4.3　ECDLP に対する攻撃手法のまとめ

本章で説明した ECDLP に対する攻撃手法の特徴を，以下の表にまとめておく：

攻撃手法	特徴
BSGS 法	位数 r の ECDLP を $O(\sqrt{r})$ 回の楕円曲線上の演算回数で求解できる攻撃法. ただし，$O(\sqrt{r})$ 個の楕円曲線の点を格納する必要があり，暗号で利用される実用サイズの ECDLP を解くのは空間計算量の観点から困難である.
ρ 法	BSGS 法と同様，位数 r の ECDLP を $O(\sqrt{r})$ の計算量で求解できる.BSGS 法と異なり，識別点の導入により少ない空間計算量で求解できると共に，大規模並列計算も比較的容易である．MOV 法や SSSA 法の攻撃対象となる特殊な曲線を除き，**一般の ECDLP を解く現在最良の攻撃法**である．（実際，100 ビット以上の位数 r の ECDLP はすべて並列化 ρ 法で求解されている.）
指数計算法	ECDLP を含む一般の DLP を解く方法. しかし現状では，暗号で利用される実用サイズの ECDLP を効率的に解くアルゴリズムが整備されていない.
MOV 攻撃法	楕円曲線上のペアリングを利用して，ECDLP を乗法群 $\mathbb{F}_{p^m}^\times$ 上の DLP に帰着する方法. 帰着先の乗法群 $\mathbb{F}_{p^m}^\times$ のサイズが小さい場合のみ有効（例えば，超特異楕円曲線の場合は $m = 2$ で非常に有効）.
SSSA 攻撃法	有限体 $\mathbb{F}_p$ 上の anomalous 曲線における ECDLP を加法群 $\mathbb{F}_p^+$ 上の DLP に帰着する非常に有効な攻撃法.

[*27] 写像 $\lambda(u)$ が $\log(p)$ の多項式時間で計算可能なので，SSSA 攻撃法は多項式時間アルゴリズムである.

[*28] 実用的に，巨大な素数 p に対しては，ほとんどの場合 $\lambda(u)$ は同型写像となる.

第 5 章

EdDSA で利用される曲線の安全性評価と効率性に関する考察

公開鍵暗号において Edwards 曲線ディジタル署名アルゴリズム（Edwards-curve Digital Signature Algorithm, EdDSA）とは，3.2.3 節で説明したツイスト Edwards 曲線を利用した Schnorr によるディジタル署名の変種の一つである．EdDSA は他の楕円曲線上のディジタル署名方式におけるいくつかの問題点を回避すると共に[*1]，ツイスト Edwards 曲線上の点の加算・2 倍算公式により効率的な暗号化処理が可能なように設計されている（詳細は [6] を参照）．本章では，EdDSA で利用される曲線に関して解説すると共に，暗号攻撃の観点からこれらの曲線がどの程度安全なのか解析評価する．さらに，通常の ECDSA と比べて EdDSA がどの程度効率的なのかを考察する．

5.1 EdDSA で利用される曲線に関する解説

RFC 8032 [31] によると，EdDSA では（古典計算機による）暗号攻撃に対して約 128 ビットのセキュリティレベルの “Ed25519” と約 224 ビットのセキュリティレベルの “Ed448” の 2 種類の実装のどちらかを適用先の安全性要件に応じて利用するよう推奨されている[*2]．特に，Ed25519 では “Curve25519” [4]，Ed448 では “Curve448” [26] [*3]と呼ばれるツイスト Edwards 曲線パラメータを利用するよう推奨されている（その他の候補曲線のパラメータについては [8] を参照）．本節では，Curve25519 と Curve448 のツイスト Edwards 曲線パラメータとその特徴について解説する（具体的な曲線パラメータについては，RFC 7748 [35] も参考にした）．

5.1.1 Curve25519 の曲線パラメータと特徴

ここでは，Ed25519 で利用する Curve25519 の曲線パラメータとその特徴について解説する．

[*1] サイドチャンネル攻撃耐性などの安全性を考慮している（詳しくは [31, Section 8] を参照）．

[*2] 具体的には，128 ビットのセキュリティレベルで十分である限り，Ed25519 の利用が推奨されている．

[*3] RFC 7748 [35] では “Curve448” と呼んでいるが，原論文 [26] では “Ed448-Goldilocks” という名称で曲線パラメータを提案している．（特に，Curve448 を定める基礎体の標数 $p = 2^{448} - 2^{224} - 1$ を Goldilocks 素数と呼ぶ．）

Curve25519 の曲線パラメータ

Curve25519 は，255 ビット素数 $p = 2^{255} - 19$ による素体 $\mathbb{F}_p$ 上の方程式 (3.15) で定義されるツイスト Edwards 曲線 $E_{a,d}$ である．ただし，係数 a, d は

$$\begin{cases} a = -1 \\ d = -\dfrac{121665}{121666} \pmod p \\ \quad = 37095705934669439343138083508754565189542113879843219016388785533085940283555 \end{cases}$$

とする．また，約 128 ビットのセキュリティレベルの署名アルゴリズム Ed25519 を構成するためのツイスト Edwards 曲線上の基点 $S = (x(S), y(S)) \in E_{a,d}(\mathbb{F}_p)$ は

$$\begin{cases} x(S) = 15112221349535400772501151409588531511454012693041857206046113283949847762202 \\ y(S) = 46316835694926478169428394003475163141307993866256225615783033603165251855960 \end{cases}$$

で与えられている．さらに，Curve25519 上の $\mathbb{F}_p$-有理点がなす群の位数は

$$\begin{aligned} \#E_{a,d}(\mathbb{F}_p) &= 2^3 \cdot r \\ r &= 2^{252} + 27742317777372353535851937790883648493 \text{ (素数)} \end{aligned} \tag{5.1}$$

で，基点 S の位数は素数 r である．（つまり，Curve25519 は余因子 8 を持つ．）

他の曲線との関係

Curve25519 を定めるツイスト Edwards 曲線 $E_{a,d}$ の係数 $a = -1$ が素体 $\mathbb{F}_p$ 上平方なので[*4]，3.2.3 節で説明したようにツイスト曲線 $E_{a,d}$ は通常 Edwards 曲線

$$E_{1,d/a} : \bar{x}^2 + \bar{y}^2 = 1 + \frac{121665}{121666}\bar{x}^2\bar{y}^2$$

と $\mathbb{F}_p$ 上同型である[*5]．さらに，定理 3.2.1 節で説明したように，この通常 Edwards 曲線 $E_{1,d/a}$ は Montgomery 曲線

$$v^2 = u^3 + 486662u^2 + u \tag{5.2}$$

と $\mathbb{F}_p$ 上双有理同値である．具体的な変数変換は

$$\bar{x} = \frac{\sqrt{486664}u}{v}, \quad \bar{y} = \frac{u-1}{u+1}$$

[*4] RFC [31] によると，係数 a は $\mathbb{F}_p$ における平方元で，具体的には $p \equiv 1 \pmod 4$ の場合は $a = -1$，$p \equiv 3 \pmod 4$ の場合は $a = 1$ と設定するよう推奨している．

[*5] 具体的には，$\bar{x} = \sqrt{-1}x$，$\bar{y} = y$ と変数変換すれば良い（$\sqrt{-1} \in \mathbb{F}_p$ に注意）．

で，特に 486664 が $\mathbb{F}_p$ における平方元であることに注意する．また，逆の変換変換は

$$u = \frac{1+\bar{y}}{1-\bar{y}}, \quad v = \frac{\sqrt{486664}u}{\bar{x}}$$

で与えられる．

Curve25519 の曲線パラメータの選択法

ここでは，Curve25519 の曲線パラメータがどのような理由で選択されているのかを説明する．まず，Montgomery 曲線に関する下記の定理を紹介する [4, Theorem 2.1] [*6]：

定理 5.1.1. $p \geq 5$ を素数とし，$A^2 - 4 \pmod{p}$ が平方でない整数を A とする．有限体 $\mathbb{F}_p$ 上の Montgomery 曲線 $E : v^2 = u^3 + Au^2 + u$ に対し，写像 $X_0 : E(\mathbb{F}_{p^2}) \longrightarrow \mathbb{F}_{p^2}$ を次のように定める：

$$X_0(\infty) = 0, \quad X_0((u,v)) = u$$

このとき，任意の整数 n と元 $q \in \mathbb{F}_p$ に対して，$X_0(Q) = q$ を満たすすべての点 $Q \in E(\mathbb{F}_{p^2})$ に対し $X_0(nQ) = s$ を満たす元 $s \in \mathbb{F}_p$ が唯一存在する．

Curve25519 を定めるツイスト Edwards 曲線 $E_{a,d}$ と双有理同値になる Montgomery 曲線 (5.2) が上記の定理の条件を満たすように，素数 $p = 2^{255} - 19$ と $A = 486662$ が選択されている．より具体的な選択理由は以下である（詳細は [4] を参照）：

■標数 p の選択理由　安全性を考慮しつつ，$32 \cdot 8 = 256$ ビットより小さい 2 冪に近い素数を選択している．（公開鍵情報を 32 ビットのワードに変換するために，$32k$ の型のビットが適している．）標数 p の候補として，$2^{255} + 95$, $2^{255} - 19$, $2^{255} - 31$, $2^{254} + 79$, $2^{253} + 51$, $2^{253} + 39$ があるが，19 が $31, 39, 51, 79, 95$ よりも小さいため $p = 2^{255} - 19$ が選択されている．

■係数 A の選択理由　様々な可能性のある暗号攻撃を避けるため[*7]，楕円曲線とそのツイストの両方の位数が $4r$ または $8r$（ただし r は巨大素数）の形となるように係数 A を選択している[*8]．また，Montgomery 型の楕円曲線 E 上の点のスカラー倍演算を効率的に行えるように，$(A-2)/4$ が十分小さい整数となることが望ましい（詳しくは Montgomery 乗法公式 (3.24) を参照）．これらの条件を満たす係数 A の候補として，小さい順に 358990, 464586, 486662 がある．$A = 358990$ と 464586 の場合は上記の r の値が 252 ビットよりほんの小さいのに対し，$A = 486662$ の場合はそうはならない（正確な位数 (5.1) を参照）．そのため，$A = 486662$ が選択されている．

[*6] この定理における性質は Diffie-Hellman 鍵共有などの暗号方式の構成 [4] に有用である一方，暗号攻撃に利用した文献は今のところ見当たらない．

[*7] 楕円曲線上のツイストを利用した暗号攻撃などを想定している．

[*8] 実際，Curve25519 の（2 次）ツイスト曲線の位数は

$$2^2 \cdot 14474011154664524427946373126085988481603263447650325797860494125407373907997$$

である．つまり，4 を余因子として持つ．

Curve25519 上の点の加算

RFC 8032 [31] では，Curve25519 を定めるツイスト Edwards 曲線 $E_{a,d}$ 上の点の加算・2 倍算においては拡張版の射影座標表現の利用を推奨している．具体的には，ツイスト Edwards 曲線 $E_{a,d}$ 上の点 (x, y) を射影空間 $\mathbb{P}^3$ 内の斉次 2 次曲線 $aX^2 + Y^2 = Z^2 + dT^2$ 上の拡張版の射影点 $(X : Y : T : Z)$ に対応させる．特に, 任意の $\lambda \neq 0$ に対し, $(X : Y : T : Z) = (\lambda X : \lambda Y : \lambda T : \lambda Z)$ を満たし，$Z \neq 0$ で $x = \frac{X}{Z}, y = \frac{Y}{Z}, xy = \frac{T}{Z}$ に対応させる．（特に，ツイスト Edwards 曲線上の零元 $(0, 1)$ は射影点 $(0 : 1 : 0 : 1)$ に対応し，拡張版の射影点 $P = (X : Y : T : Z)$ の逆元は $-P = (-X : Y : -T : Z)$ で与えられる.) 以下に, 拡張版の射影座標表現によるツイスト Edwards 曲線上の加算公式の具体的な計算手順を示す [31, Section 5.1.4]（詳細は [11, 28] を参照）：

■加算　拡張版の射影座標表現によるツイスト Edwards 曲線上の 2 つの射影点 $(X_1 : Y_1 : T_1 : Z_1)$, $(X_2 : Y_2 : T_2 : Z_2)$ の加算点 $(X_3 : Y_3 : T_3 : Z_3)$ は次の計算手順で求まる：

$$\begin{cases} A = (Y_1 - X_1) \cdot (Y_2 - X_2);\ B = (Y_1 + X_1) \cdot (Y_2 + X_2);\ C = 2dT_1 \cdot T_2; \\ D = 2Z_1 \cdot Z_2;\ E = B - A;\ F = D - C;\ G = D + C;\ H = B + A; \\ X_3 = E \cdot F;\ Y_3 = G \cdot H;\ T_3 = E \cdot H;\ Z_3 = F \cdot G; \end{cases}$$

この計算コストは $8\mathbf{M} + 8\mathbf{add}$ である．（ただし，定数との掛け算コストは無視する.）表 3.2 から拡張版の射影座標表現による加算は他の座標表現よりも効率的であることが分かる．

■2 倍算　拡張版の射影座標表現によるツイスト Edwards 曲線上の射影点 $(X_1 : Y_1 : T_1 : Z_1)$ の 2 倍点 $(X_3 : Y_3 : T_3 : Z_3)$ は次の計算手順で求まる：

$$\begin{cases} A = X_1^2;\ B = Y_1^2;\ C = 2Z_1^2;\ H = A + B; \\ E = H - (X_1 + Y_1)^2;\ G = A - B;\ F = C + G; \\ X_3 = E \cdot F;\ Y_3 = G \cdot H;\ T_3 = E \cdot H;\ Z_3 = F \cdot G; \end{cases}$$

この計算コストは $4\mathbf{M} + 4\mathbf{S} + 5\mathbf{add}$ である．（ただし，定数との掛け算コストは無視する.）加算の場合とは異なり，拡張版の射影座標表現による 2 倍算は他の座標表現より必ずしも効率的とは限らないことが表 3.2 から分かる．

5.1.2　Curve448 の曲線パラメータと特徴

ここでは，約 224 ビットのセキュリティレベルの署名アルゴリズム Ed448 で利用する Curve448 の曲線パラメータとその特徴について解説する．

Curve448 の曲線パラメータ

Curve448 は，448 ビット素数 $p = 2^{448} - 2^{224} - 1$ による素体 $\mathbb{F}_p$ 上の方程式 (3.11) で定義される通常 Edwards 曲線 E_d である．ただし，係数は $d = -39081$ とする．また，署名アルゴリズム

Ed448 を構成するための通常 Edwards 曲線上の基点 $S = (x(S), y(S)) \in E_d(\mathbb{F}_p)$ は

$$\begin{cases} x(S) = & 224580040295924300187604334099896036246789641632564134246125461686950415467406032909029192869357953282578032075146446173674602635247710 \\ y(S) = & 298819210078481492676017930443930673437544040154080242095928241372331506189835876003536878655418784733982303233503462500531545062832660 \end{cases}$$

で与えられている．さらに，Curve448 上の $\mathbb{F}_p$-有理点がなす群の位数は

$$\begin{aligned} \#E_d(\mathbb{F}_p) =& 2^2 \cdot r \\ r =& \ 2^{446} - 138180668098951153520073867485154268803366924748821786098945475038853 \end{aligned} \quad (5.3)$$

(素数)

で，基点 S の位数は素数 r である．（つまり，Curve448 は余因子 4 を持つ.）

Curve448 の曲線パラメータの選択法

ここでは，Curve448 の曲線パラメータがどのような理由で選択されているのかを説明する.

■標数 p の選択理由　有限体上の四則演算を効率的に行えるよ 3 項式の Solinas 素数[*9] である $p = 2^{448} - 2^{224} - 1$ を選択している．特に，素数 p は [26] で Goldilocks 素数と呼ばれ，$\phi = 2^{224}$ に対し法 p による整数を $a + b\phi$ の形で表すことにより，高速な Karatsuba 乗算などが可能となる [26, Section 3.2].（ただし，a, b は ϕ 未満の非負整数とする.）特に，$224 = 32 \cdot 7 = 28 \cdot 8 = 56 \cdot 4$ により，28 ビット・32 ビットや 56 ビットなどの様々なサイズのワードを利用した有限体 $\mathbb{F}_p$ 上の高速演算が可能である.

■Edwards 曲線の選択理由　Curve448 を定める有限体 $\mathbb{F}_p$ 上の通常 Edwards 曲線 E_d の中で，係数パラメータ d の絶対値が十分小さく，E_d とそのツイストの位数が共に $4r$（ただし r は巨大素数）の形になるよう $d = -39081$ を選択している（正確な位数は (5.3) を参照[*10]）．特に，選択された Edwards 曲線 E_d は SafeCurves [12] における安全性要件[*11]をすべて満たしている.

Curve448 上の点の加算

RFC 8032 [31] では，Curve448 を定める通常 Edwards 曲線 E_d 上の点の加算・2 倍算においては，3.2.1 節で説明した射影座標表現の利用を推奨している.

[*9] $2^k - 2^\ell \pm \cdots \pm 1$ の形の素数を Solinas 素数と呼ぶ.

[*10] また，Curve448 を定める通常 Edwards 曲線 E_d のツイストの位数は

$$4 \cdot \left(2^{446} + 338093476319795454673879205005611543518120263611892369342742379277\right)$$

で，4 を余因子として持つ.

[*11] 安全性要件として，ρ 法による解読計算量，一般の離散対数問題に帰着した場合の解読計算量，虚数乗法における判別式の大きさ，ツイスト楕円曲線における安全性などの項目がある.

5.2　EdDSA で利用される曲線に対する攻撃計算量評価

本節では，EdDSA で利用される Curve25519 と Curve448 の 2 つの曲線上の ECDLP に対して，現時点で知られている最良の攻撃法を選択すると共に，その攻撃計算量を評価する．

5.2.1　最良の攻撃法の選択

ここでは，EdDSA で利用される Curve25519 と Curve448 の 2 つの曲線に対して，現時点で知られている最良の攻撃法を選択する．まず，Curve25519 と Curve448 の 2 つの曲線に対して，MOV 攻撃法や SSSA 攻撃法の特殊攻撃法が有効になり得るのか議論する：

■MOV 攻撃法に対して　4.2.1 節で説明したように，MOV 攻撃法では ECDLP を乗法群 $\mathbb{F}_{p^m}^{\times}$ 上の DLP に帰着するが（ただし，m は埋め込み次数とする[*12]），帰着先の乗法群のサイズが小さい場合のみ有効である．具体的には，楕円曲線暗号の安全性のため，IEEE P1363 [38] では 30 以上，ANSI X9.62 [1] では少なくとも 100 以上[*13]の埋め込み次数を推奨している．また，より十分な安全性を担保するため，ECC Brainpool [18] では

$$m = \frac{r-1}{100} \tag{5.4}$$

以上の埋め込み次数を推奨している．（ただし，r は ECDLP における基点の位数とする．）さらに，SafeCurves [12] では，乗法群上の DLP 解法に関するこれまでの研究進展から IEEE P1363 や ANSI X9.62 の基準では安全性に対する懸念があるとする一方，ECC Brainpool の基準は過度な安全性レベルとしている．しかし，十分な安全性担保の観点から，SafeCurves では ECC Brainpool の基準を推奨している[*14]．一方，本報告書の評価対象である Curve25519 における埋め込み次数は

$$m = \frac{r-1}{6} = 1206167596222043702328864427173832373476186059896651267666991823047575708498$$

であり，Curve448 における埋め込み次数は

$$m = \frac{r-1}{2} = 908548405369508613186654759860005667942051700859147575351862748975730019807697928580978776458461879816551468545458311523868779298248889$$

[*12] 具体的には，ECDLP インスタンス $(E/\mathbb{F}_p, S, T, r)$ に対して，埋め込み次数 m は条件 $r \mid p^m - 1$ を満たす最小の自然数である．

[*13] ANSI X9.62 (1998) では埋め込み次数の基準を 20 以上としていた（SEC1 [44] も参照）．

[*14] SafeCurves [12] における ECDLP の安全性要件として，ρ 法による攻撃計算量や MOV 攻撃法や SSSA 攻撃法による帰着攻撃以外に，楕円曲線における虚数乗法の判別式の絶対値の大きさや曲線生成に関する詳細な説明の有無に関する項目がある．しかし，これらの項目は ECDLP に対する直接的な攻撃と関係が薄いため，本報告書では取り上げない事とした．（ちなみに，EdDSA で利用される Curve25519 と Curve448 の 2 つの曲線は共に，SafeCurves の安全性要件をすべて満たしている．）

である．つまり，Curve25519 と Curve448 の 2 つの曲線は共に ECC Brainpool の基準 (5.4) よりも大きな埋め込み次数を持つので，これらの 2 つの曲線に対しては MOV 攻撃法は有効ではない．

■SSSA 攻撃法に対して　前節の曲線パラメータ説明で述べたように，Curve25519 と Curve448 の 2 つの曲線は共に anomalous ではないので，そもそも SSSA 攻撃法は適用できない．

上記の議論より，EdDSA で利用される Curve25519 と Curve448 の 2 つの曲線に対しては，MOV 攻撃法や SSSA 攻撃法の特殊な攻撃法は有効ではない．一方，4.3 節のまとめから，特殊な攻撃法を除けば，ρ 法が現時点での最良の攻撃法である．

5.2.2　EdDSA で利用される曲線に対する ρ 法による攻撃計算量評価

ここでは，EdDSA で利用される Curve25519 と Curve448 の 2 つの曲線に対する ρ 法による攻撃計算量評価を行う．

Edwards 曲線に対する ρ 法による攻撃実験

ρ 法は誕生日の逆理に基づく確率的な攻撃アルゴリズムであるため，攻撃対象となる曲線の違いにより計算量が大きく変化する可能性がある．ここでは，攻撃実験可能なサイズの Edwards 曲線に対する ρ 法攻撃の振る舞いを実験的に解析する．

■攻撃実験用の Edwards 曲線パラメータ生成　ρ 法の攻撃実験用に，40 ビット程度の素数 p による有限体 $\mathbb{F}_p$ 上の（ツイストなしの）Edwards 曲線 E_d を生成する[*15]．特に，Curve25519 と Curve448 と同じように，曲線とそのツイストの位数が共に $4r$（ただし r は素数）の形となる Edwards 曲線を選択する．3.2 節で説明したように，Edwards 曲線 E_d は Weierstrass 方程式

$$\begin{aligned} C_d:\ y^2 &= (x-d-1)(x^2-4d) \\ &= x^3-(d+1)x^2-4dx+4d(d+1) \end{aligned} \tag{5.5}$$

で定まる楕円曲線に同型であることに注意する[*16]．さらに，Edwards 曲線 E_d 上の点の加算は，Weierstrass 方程式 (5.5) で定まる楕円曲線 C_d 上の点の加算と一致するため，2 つの曲線 E_d と C_d に対する ρ 法攻撃の振る舞いも一致する．本実験では，楕円曲線に関する機能を豊富に持つ数式処理ソフトウェア SageMath [25] を利用するので[*17]，Weierstrass 方程式 (5.5) で定める曲線 C_d 上の ECDLP インスタンスを生成し，ρ 法による攻撃実験を行う．

攻撃実験用に生成した具体的な ECDLP インスタンス $(C_d/\mathbb{F}_p, S, T, r)$ は以下である（下記の ECDLP インスタンスを作成する際に利用した Sage スクリプトを図 5.1 に示しておく）：

[*15] Edwards 曲線 E_d を定める方程式は (3.11) である．

[*16] Edwards 曲線と同型なので，曲線 C_d も位数 4 のねじれ点を必ず持つ．

[*17] SageMath では Weierstrass 方程式で定まる楕円曲線に関して非常に多くの関数を利用することができる．

```
p = next_prime(2^40); print("p = ", p)
while(1):
    d = randint(0, p)
    E = EllipticCurve(GF(p), [0, -d-1, 0, -4*d, 4*d*(d+1)])
    t = E.trace_of_frobenius()
    r = ZZ((p+1-t)/4)  # p+1-t: order of E
    rr = ZZ((p+1+t)/4) # p+1+t: order of twist of E
    if is_prime(r) == True and is_prime(rr) == True:
        print("d = ", d); print("r = ", r)
        P = E.gens()[0]
        S = 4*P; print("S = ", S)
        T = randint(0, r)*S; print("T = ", T)
        break
```

図 5.1　ECDLP インスタンス (5.6) の生成用 Sage スクリプト

$$\begin{cases} p = 1099511627791\ (\approx 2^{40}) \\ d = 416824108313 \\ S = (176845795840, 980148948053) \in C_d(\mathbb{F}_p) \\ T = (697615400391, 406954616758) \in C_d(\mathbb{F}_p) \\ r = 274878050303 \end{cases} \tag{5.6}$$

■Edwards 曲線に対する ρ 法攻撃の実験データ　上記で生成した Edwards 曲線上の ECDLP インスタンス (5.6) に対して，ρ 法攻撃アルゴリズムを実装し，攻撃実験を行った．ρ 法攻撃アルゴリズムの実装においては，分割数を $L = 16$ に設定した (4.1) で定義された反復関数 f を用いた[*18]．図 5.4 に ECDLP に対する ρ 法攻撃アルゴリズムの Sage スクリプトを示しておく．（ただし，簡単のため，本攻撃実験では楕円曲線上の逆元写像を利用した ρ 法の高速化手法は用いなかった．）また，図 5.2 に，ρ 法攻撃において衝突が起こるまでに必要した楕円曲線の点列の個数に関するヒストグラムを示す．具体的には，ρ 法における初期点をランダムに入れ替え，10,000 回の攻撃実験から得られた度数分布を図 5.2 に示した．（ただし，横軸は誕生日の逆理に基づく期待値 (4.2) で正規化した．）図 5.2 から，横軸がほぼ期待値 (4.2) において分布の山の頂点となることが分かる．さらに，分布の山のすその広がり方は，一般の楕円曲線上に対する ρ 法攻撃における分布図 [60, Figures 3, 4] とほぼ同じであることが分かる．これより，Edwards 曲線上の ρ 法攻撃で必要となる楕円曲線の点列の個数は，一般の楕円曲線上の個数とほぼ同じであると結論付けることができる[*19]．

[*18] 4.1.2 節で説明したように，分割数 $L \geq 16$ で反復関数 (4.1) は楕円曲線上ランダムな振る舞いをする．

[*19] ρ 法攻撃で必要となる楕円曲線の点列の個数に関する度数分布は，楕円曲線を定める基礎体のサイズに依存しないことが [60] の実験で示されている．これより，EdDSA で利用される Edwards 曲線に対しても，図 5.2 とほぼ同様の度数分布が得られると期待できる．

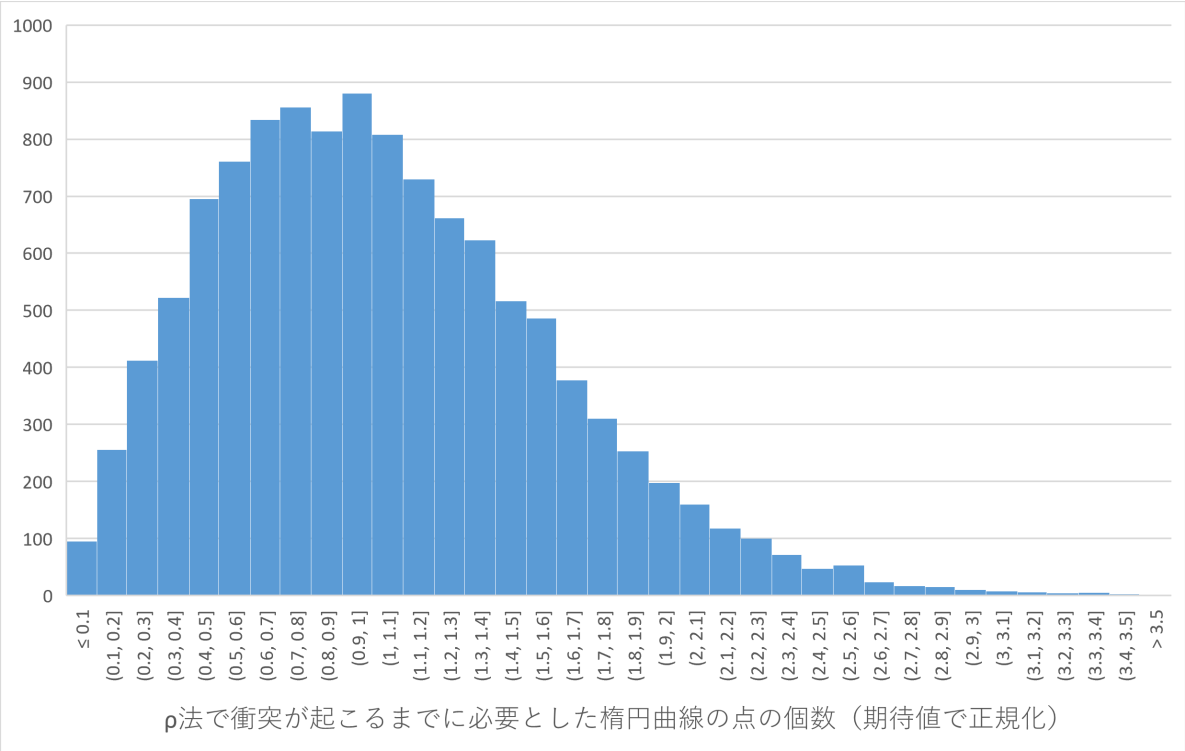

図 5.2　Edwards 曲線上の ECDLP インスタンス (5.6) に対する ρ 法攻撃において，衝突が起こるまでに必要とした楕円曲線の点列の個数に関するヒストグラム（ただし，横軸は誕生日の逆理に基づく期待値 (4.2) で正規化した）

Curve25519 と Curve448 に対する ρ 法による攻撃計算量評価

前述の実験結果と式 (4.3) から，Edwards 曲線上の位数 r を持つ ECDLP に対する ρ 法による平均攻撃計算量は

$$\frac{\sqrt{\pi r}}{2} \cdot t(f) \approx 0.8862\sqrt{r} \cdot t(f) \tag{5.7}$$

と見積もれる．ただし，攻撃者有利な条件として，逆元計算による $\sqrt{2}$ 倍の高速化も考慮した．また，$t(f)$ は反復関数 f の計算時間で，反復関数の定め方から本質的には Edwards 曲線上の加算コストに一致する．具体的には，Curve25519 の位数は $r \approx 2^{252}$ より，Curve25519 における ECDLP を攻撃するには，平均的に約 $0.8862 \cdot 2^{126} = 2^{125.8257}$ 回の点の加算が必要となる．また，Curve448 の位数は $r \approx 2^{446}$ より，Curve448 における ECDLP を攻撃するには，平均的に約 $0.8862 \cdot 2^{223} = 2^{222.8257}$ 回の点の加算が必要となる．一方，128 ビットセキュリティレベルの ECDSA での利用が推奨されている P-256 曲線に関して，P-256 の位数が $r = 2^{256}$ より P-256 における ECDLP を攻撃するには，平均的に約 $0.8862 \cdot 2^{128} = 2^{127.8257}$ 回の点の加算が必要となる（P-256 の具体的な曲線パラメータは [1] を参照）．これより，P-256 と同程度のセキュリティレベルに設定されている Curve25519 と安全性比較すると，Curve25519 上の方が平均的に約 4 倍少な

表 5.1　様々な座標表現による楕円曲線上のスカラー倍算 nP の計算コスト比較 [9, Section 6]
（項目 (1)–(8) は表 3.1 と同じ座標表現. 項目 (8) が Edwards 曲線における射影座標表現）

座標	NAF アルゴリズム		符号付 sliding window 法（幅は 4）	
表現	計算コスト	$\mathbf{S}/\mathbf{M} = 0.8$	計算コスト	$\mathbf{S}/\mathbf{M} = 0.8$
(1)	$8\mathbf{M} + 6.67\mathbf{S} + 1\mathbf{D}$	$13.3\mathbf{M}$	$7.17\mathbf{M} + 6.28\mathbf{S} + 0.98\mathbf{D}$	$12.2\mathbf{M}$
(2)	$3.33\mathbf{M} + 9.33\mathbf{S} + 1\mathbf{D}$	$10.8\mathbf{M}$	$2.85\mathbf{M} + 8.64\mathbf{S} + 0.98\mathbf{D}$	$9.77\mathbf{M}$
(3)	$4.67\mathbf{M} + 7\mathbf{S} + 2.33\mathbf{D}$	$10.3\mathbf{M}$	$3.69\mathbf{M} + 6.48\mathbf{S} + 2.16\mathbf{D}$	$8.87\mathbf{M}$
(4)	$6.67\mathbf{M} + 4.67\mathbf{S} + 0.33\mathbf{D}$	$10.4\mathbf{M}$	$5.09\mathbf{M} + 4.32\mathbf{S} + 0.19\mathbf{D}$	$8.54\mathbf{M}$
(5)	$10.3\mathbf{M} + 1\mathbf{S}$	$11.1\mathbf{M}$	$9.16\mathbf{M} + 0.98\mathbf{S}$	$9.94\mathbf{M}$
(6)	$4.67\mathbf{M} + 6.33\mathbf{S} + 2.33\mathbf{D}$	$9.73\mathbf{M}$	$4.2\mathbf{M} + 5.86\mathbf{S} + 2.16\ \mathbf{D}$	$8.88\mathbf{M}$
(7)	$4.33\mathbf{M} + 8.33\mathbf{S} + 2.33\mathbf{D}$	$11\mathbf{M}$	$3.84\mathbf{M} + 7.99\mathbf{S} + 2.16\mathbf{D}$	$10.2\mathbf{M}$
(8)	$6\mathbf{M} + 4.33\mathbf{S} + 0.33\mathbf{D}$	$9.47\mathbf{M}$	$4.86\mathbf{M} + 4.12\mathbf{S} + 0.194\mathbf{D}$	$8.16\mathbf{M}$

い回数の楕円加算で攻撃できる．さらに，ツイスト Edwards 曲線は効率的な点の加算公式を持つため，P-256 よりも Curve25519 上の方がより効率的に点の加算が可能であり，ECDLP をより効率的に攻撃できる．例えば，P-256 よりも Curve25519 の方が最大 2 倍高速に楕円加算ができたと想定すると，P-256 よりも Curve25519 における ECDLP の方が平均的に最大 8 倍高速に攻撃できる．ただし，Curve25519 における ECDLP を攻撃するには，少なくとも $2^{125.8257}$ 回の楕円加算が必要であり，ほぼ 128 ビットのセキュリティレベルを持つと結論付けれる．

5.3　ECDSA と比較した場合の曲線としての効率性に関する考察

本節では，ECDSA と比較した場合，EdDSA で利用される Edwards 曲線の効率性に関する考察を行う．まず，通常の楕円曲線と Edwards 曲線上の点の加算と 2 倍算の計算コスト比較は表 3.1 にまとめてある．具体的には，基礎体上の乗算コスト $\mathbf{M}$ と 2 乗算コスト $\mathbf{S}$ の比が $\mathbf{S}/\mathbf{M} = 0.8$ の場合，通常の楕円曲線で標準的に利用する射影座標表現における点の加算コストは $10.8\mathbf{M}$ で 2 倍算コストは $9.8\mathbf{M}$ であるのに対し，Edwards 曲線の射影座標表現における点の加算コストは $9.8\mathbf{M}$ で 2 倍算コストは $6.2\mathbf{M}$ である．これより，同一の基礎体の演算を利用した場合，射影座標表現においては Edwards 曲線の方が点の加算で約 9.4%，点の 2 倍算で約 36.7% 効率的に計算できると見積もれる．また，楕円曲線を利用したディジタル署名では，署名生成時に楕円曲線の点 P のスカラー倍算 nP を行い，署名検証時には楕円曲線の点 P_1, P_2 の複数スカラー倍算 $n_1P_1 + n_2P_2$ を主に行う．表 3.1 と同じように，表 5.1 に座標表現による楕円曲線上のスカラー倍算 nP の計算コスト比較をまとめておく（詳細は [9, Section 6] を参照[*20]）．表 5.1 より，同一の基礎体を利用した場合，スカラー倍算 nP に関しては Edwards 曲線の方が最大 33% 程度効率的に行うことができ

[*20] NAF アルゴリズムなどの具体的なスカラー倍算アルゴリズムについては [2, 19] を参照.

表 5.2　様々な座標表現による楕円曲線上の複数スカラー倍算 $n_1P_1 + n_2P_2$ の計算コスト比較 [9, Section 7]（座標表現は表 3.1 と同じ. 項目 (8) が Edwards 曲線における射影座標表現）

座標	JSF アルゴリズム		加速化 ECDSA 検証アルゴリズム	
表現	計算コスト	$\mathbf{S}/\mathbf{M} = 0.8$	計算コスト	$\mathbf{S}/\mathbf{M} = 0.8$
(1)	$10.2\mathbf{M} + 7\mathbf{S} + 1\mathbf{D}$	$15.8\mathbf{M}$	$6.92\mathbf{M} + 3\mathbf{S} + 0.33\mathbf{D}$	$9.32\mathbf{M}$
(2)	$5.25\mathbf{M} + 10\mathbf{S} + 1\mathbf{D}$	$12.8\mathbf{M}$	$4.58\mathbf{M} + 4.67\mathbf{S} + 0.33\mathbf{D}$	$8.32\mathbf{M}$
(3)	$6.25\mathbf{M} + 7.5\mathbf{S} + 2.5\mathbf{D}$	$12.2\mathbf{M}$	$4.92\mathbf{M} + 3.5\mathbf{S} + 1.17\mathbf{D}$	$7.72\mathbf{M}$
(4)	$8.5\mathbf{M} + 5\mathbf{S} + 0.5\mathbf{D}$	$12.5\mathbf{M}$	$6.5\mathbf{M} + 2.33\mathbf{S} + 0.5\mathbf{D}$	$8.37\mathbf{M}$
(5)	$12.5\mathbf{M} + 1\mathbf{S}$	$13.3\mathbf{M}$	$7.83\mathbf{M} + 0.33\mathbf{S}$	$8.1\mathbf{M}$
(6)	$7\mathbf{M} + 7.25\mathbf{S} + 2.5\mathbf{D}$	$12.8\mathbf{M}$	$5.67\mathbf{M} + 3.92\mathbf{S} + 1.17\mathbf{D}$	$8.8\mathbf{M}$
(7)	$6.25\mathbf{M} + 9.5\mathbf{S} + 2.5\mathbf{D}$	$13.8\mathbf{M}$	$4.92\mathbf{M} + 4.83\mathbf{S} + 1.17\mathbf{D}$	$8.78\mathbf{M}$
(8)	$7.75\mathbf{M} + 4.5\mathbf{S} + 0.5\mathbf{D}$	$11.3\mathbf{M}$	$5.75\mathbf{M} + 1.83\mathbf{S} + 0.5\mathbf{D}$	$7.22\mathbf{M}$

ることが分かる．一方，表 5.2 に座標表現による楕円曲線上の複数スカラー倍算 $n_1P_1 + n_2P_2$ の計算コスト比較をまとめておく（詳細は [9, Section 7] を参照[*21]）．表 5.2 より，同一の基礎体を利用した場合，複数スカラー倍算 $n_1P_1 + n_2P_2$ に関しても Edwards 曲線の方が最大 28% 程度効率的に行うことができることが分かる．特に，EdDSA で利用される Curve25519 においては，高速実装に適した基礎体 $\mathbb{F}_p$ を選択しており，基礎体上の演算の高速化分を考慮すれば，更に効率的に（複数）スカラー倍算を行うことが可能となる．実際，図 5.3 に P-256 曲線[*22]による ECDSA と Curve25519 による EdDSA のハードウェア実装による処理時間の比較を示す．ただし，処理時間は鍵生成・署名生成・署名検証の合計時間で，その単位は cycles 数とする（詳細は [13] を参照）. ハードウェアの実装方法により処理時間が大きく異なるため参考程度ではあるが, 図 5.3 より Curve25519 による EdDSA の方が最大 2 倍程度高速であることが分かる.（実際，縦軸が amd64 Zen2 の項目で比べると，図左側の P-256 曲線による ECDSA の処理が約 524288cycles であるのに対し，図右側の Curve25519 による EdDSA の処理が約 262144cycles である.）

[*21] 表 5.2 には JSF アルゴリズムと加速化 ECDSA 検証アルゴリズムの計算コスト比較しかまとめなかったが，[9, Section 7] では他にも Bos-Coster アルゴリズムと point-comb アルゴリズムの計算コスト比較が示されている.

[*22] P-256 曲線の基礎体 $\mathbb{F}_p$ を定める素数 は $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ であり [1, Annex B]，そのサイズは Curve25519 の基礎体サイズと同程度である.

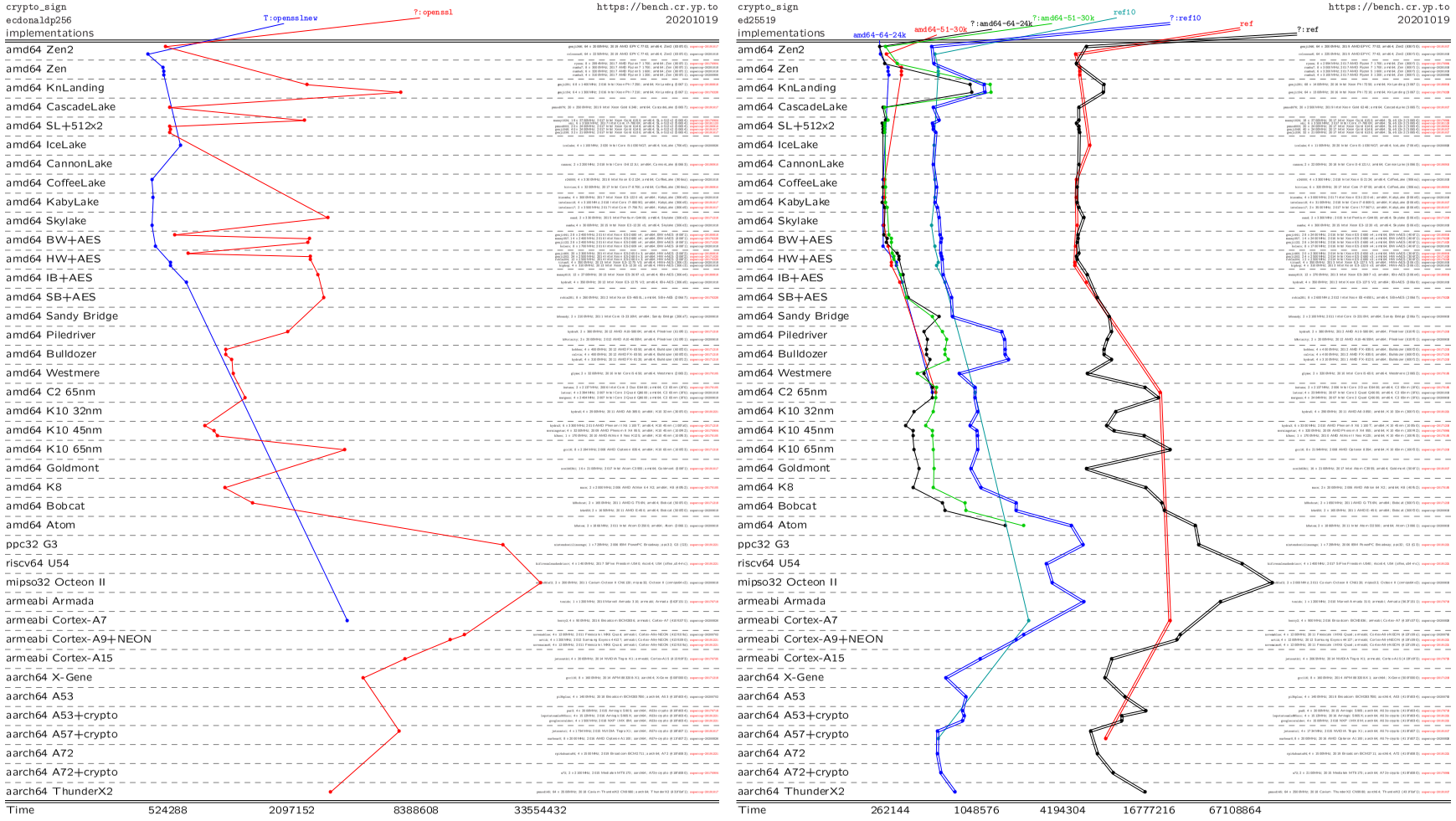

図 5.3　P-256 曲線による ECDSA（図左側）と Curve25519 による EdDSA（図右側）のハードウェア実装による処理時間（cycle 数）の比較（詳細は [13] を参照. 処理時間は鍵生成・署名生成・署名検証の合計時間）

```
def rho(E, S, T, r):
    L = 16 # Partition number of L-adding addition walk
    a = vector(ZZ, L); b = vector(ZZ, L); R = []
    for j in range(L):
        a[j] = randint(0, r)
        b[j] = randint(0, r)
        R.append(a[j]*S + b[j]*T)

    # Generation of an initial point
    u = randint(0, r); v = randint(0, r); P = u*S + v*T

    # Main loop
    List=[]; List1=[]; flag = 0; count = 0
    while (1):
        j = lift(mod(lift(P[0]), L))
        u += a[j]; v += b[j]; P += R[j]; count += 1
        if lift(mod(lift(P[0]), 1000)) == 0: # Distinguished points
            if (P in List1) == True:
                for i in range(len(List)):
                    if List[i][0] == P:
                        P1 = List[i][0]; u1 = List[i][1]; v1 = List[i][2]
                        flag=1; # A collision occurs
                        break
            if flag==1:
                break
            else:
                List.append([P, u, v]); List1.append(P)

    # After a collision between P =u*S + v*T and P1 = u1*S + v1*T
    if v==v1:
        return False
    else:
        k = mod(u1-u, r)*(v-v1).inverse_mod(r); # The solution of ECDLP
        print("count = ", count)    # Number of elliptic additions
        return k
```

図 5.4　ECDLP に対する ρ 法攻撃アルゴリズムの Sage スクリプト

参考文献

[1] American National Standards Institute. American National Standard for Financial Services X9.62-2005, Public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ECDSA), 2005.

[2] Roberto M Avanzi. The complexity of certain multi-exponentiation techniques in cryptography. *Journal of Cryptology*, Vol. 18, No. 4, pp. 357–373, 2005.

[3] Daniel V Bailey, Lejla Batina, Daniel J Bernstein, Peter Birkner, Joppe W Bos, Hsieh-Chung Chen, Chen-Mou Cheng, Gauthier Van Damme, Giacomo de Meulenaer, Luis J Dominguez Perez, et al. Breaking ECC2K-130. *IACR ePrint Archive 2009/541*, 2009.

[4] Daniel J Bernstein. Curve25519: New Diffie-Hellman speed records. In *Public Key Cryptography–PKC 2006*, Vol. 3958 of *Lecture Notes in Computer Science*, pp. 207–228. Springer, 2006.

[5] Daniel J Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted Edwards curves. In *Progress in Cryptology–AFRICACRYPT 2008*, Vol. 5023 of *Lecture Notes in Computer Science*, pp. 389–405. Springer, 2008.

[6] Daniel J Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, Vol. 2, No. 2, pp. 77–89, 2012.

[7] Daniel J Bernstein, Susanne Engels, Tanja Lange, Ruben Niederhagen, Christof Paar, Peter Schwabe, and Ralf Zimmermann. Faster elliptic-curve discrete logarithms on FPGAs. *IACR ePrint Archive 2016/382*, 2016.

[8] Daniel J Bernstein, Simon Josefsson, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. EdDSA for more curves. *IACR ePrint Archive 2015/677*, 2015.

[9] Daniel J Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In *Advances in Cryptology–ASIACRYPT 2007*, Vol. 4833 of *Lecture Notes in Computer Science*, pp. 29–50. Springer, 2007.

[10] Daniel J Bernstein and Tanja Lange. Inverted Edwards coordinates. In *Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes (AAECC 2007)*, Vol. 4851 of *Lecture Notes in Computer Science*, pp. 20–27. Springer, 2007.

[11] Daniel J Bernstein and Tanja Lange. Explicit-formula database (EFD). `https://www.hyperelliptic.org/EFD/`, since 2007.

[12] Daniel J Bernstein and Tanja Lange. SafeCurves: Choosing safe curves for elliptic-curve cryptography. `https://safecurves.cr.yp.to/`, since 2014.

[13] Daniel J Bernstein and Tanja Lange. eBACS: ECRYPT benchmarking of cryptographic systems. `http://bench.cr.yp.to/`, version 2019.08.05.

[14] Daniel J Bernstein, Tanja Lange, and Peter Schwabe. On the correct use of the negation map in the Pollard rho method. In *Public Key Cryptography–PKC 2011*, Vol. 6571 of *Lecture Notes in Computer Science*, pp. 128–146. Springer, 2011.

[15] Ian Blake, Gerald Seroussi, Gadiel Seroussi, and Nigel Smart. *Elliptic Curves in Cryptography*, Vol. 265. Cambridge university press, 1999.

[16] Joppe W Bos, Marcelo E Kaihara, Thorsten Kleinjung, Arjen K Lenstra, and Peter L Montgomery. Solving a 112-bit prime elliptic curve discrete logarithm problem on game consoles using sloppy reduction. *International Journal of Applied Cryptography*, Vol. 2, No. 3, pp. 212–228, 2012.

[17] Joppe W Bos and Arjen K Lenstra. *Topics in Computational Number Theory Inspired by Peter L. Montgomery*. Cambridge University Press, 2017.

[18] ECC Brainpool. ECC Brainpool standard curves and curve generation. `https://www.teletrust.de/fileadmin/files/oid/oid_ECC-Brainpool-Standard-curves-V1.pdf`, 2005.

[19] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. CRC press, 2005.

[20] Harold Edwards. A normal form for elliptic curves. *Bulletin of the American mathematical society*, Vol. 44, No. 3, pp. 393–422, 2007.

[21] Gerhard Frey and Hans-Georg Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of computation*, Vol. 62, No. 206, pp. 865–874, 1994.

[22] Steven D Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012.

[23] Steven D Galbraith and Pierrick Gaudry. Recent progress on the elliptic curve discrete logarithm problem. *Designs, Codes and Cryptography*, Vol. 78, No. 1, pp. 51–72, 2016.

[24] Robert Gallant, Robert Lambert, and Scott Vanstone. Improving the parallelized Pollard lambda search on anomalous binary curves. *Mathematics of Computation*, Vol. 69, No. 232, pp. 1699–1705, 2000.

[25] The Sage Group. SageMath: Open-source mathematical software system. `https://www.sagemath.org/`.

[26] Mike Hamburg. Ed448-Goldilocks, a new elliptic curve. *IACR ePrint Archive 2015/625*, 2015.

[27] Darrel Hankerson, Alfred J Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer Science & Business Media, 2006.

[28] Huseyin Hisil, Kenneth Koon-Ho Wong, Gary Carter, and Ed Dawson. Twisted Edwards curves revisited. In *Advances in Cryptology–ASIACRYPT 2008*, Vol. 5350 of *Lecture Notes in Computer Science*, pp. 326–343. Springer, 2008.

[29] Ming-Deh Huang, Michiel Kosters, Christophe Petit, Sze Ling Yeo, and Yang Yun. Quasi-subfield polynomials and the elliptic curve discrete logarithm problem. *Journal of Mathematical Cryptology*, Vol. 14, No. 1, pp. 25–38, 2020.

[30] Michael J Jacobson, Neal Koblitz, Joseph H Silverman, Andreas Stein, and Edlyn Teske. Analysis of the xedni calculus attack. *Designs, Codes and Cryptography*, Vol. 20, No. 1, pp. 41–64, 2000.

[31] Simon Josefsson and Ilari Liusvaara. RFC 8032: Edwards-curve digital signature algorithm (EdDSA). Internet Engineering Task Force (IETF). `https://tex2e.github.io/rfc-translater/html/rfc8032.html`, 2017.

[32] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, Vol. 48, No. 177, pp. 203–209, 1987.

[33] Momonari Kudo, Yuki Yokota, Yasushi Takahashi, and Masaya Yasuda. Acceleration of index calculus for solving ECDLP over prime fields and its limitation. In *Cryptology and Network Security (CANS 2018)*, Vol. 11124 of *Lecture Notes in Computer Science*, pp. 377–393. Springer, 2018.

[34] Takuya Kusaka, Sho Joichi, Ken Ikuta, Md Al-Amin Khandaker, Yasuyuki Nogami, Satoshi Uehara, Nariyoshi Yamai, and Sylvain Duquesne. Solving 114-bit ECDLP for a Barreto-Naehrig curve. In *Information Security and Cryptology (ICISC 2017)*, Vol. 10779 of *Lecture Notes in Computer Science*, pp. 231–244. Springer, 2017.

[35] A Langley, M Hamburg, and S Turner. RFC 7748: Elliptic curves for security. Internet Engineering Task Force (IETF). `https://tools.ietf.org/pdf/rfc7748.pdf`, 2016.

[36] Alfred J Menezes, Tatsuaki Okamoto, and Scott A Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on information Theory*, Vol. 39, No. 5, pp. 1639–1646, 1993.

[37] Victor S Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology–CRYPTO 1985*, Vol. 218 of *Lecture Notes in Computer Science*, pp. 417–426. Springer, 1985.

[38] Institute of Electrical and Electronics Engineers. IEEE 1363-2000: Standard specifications for public key cryptography. `https://perso.telecom-paristech.fr/guilley/recherche/cryptoprocesseurs/ieee/00891000.pdf`, 2000.

[39] Christophe Petit, Michiel Kosters, and Ange Messeng. Algebraic approaches for the elliptic curve discrete logarithm problem over prime fields. In *Public-Key Cryptography–PKC 2016*, Vol. 9615 of *Lecture Notes in Computer Science*, pp. 3–18. Springer, 2016.

[40] Stephen Pohlig and Martin Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. *IEEE Transactions on information Theory*, Vol. 24, No. 1, pp. 106–110, 1978.

[41] John M Pollard. Monte carlo methods for index computation (mod p). *Mathematics of computation*, Vol. 32, No. 143, pp. 918–924, 1978.

[42] John M Pollard. Kangaroos, monopoly and discrete logarithms. *Journal of cryptology*, Vol. 13, No. 4, pp. 437–447, 2000.

[43] Certicom Research. Standards for efficient crytography 2: Recommended elliptic curve domain parameters. `https://www.secg.org/SEC2-Ver-1.0.pdf`, 2000.

[44] Certicom Research. Standards for efficient crytography, SEC1: Elliptic curve cryptography (version 2.0). `https://www.secg.org/sec1-v2.pdf`, 2009.

[45] Certicom Research. The Certicom ECC Challenge. `https://www.certicom.com/content/certicom/en/the-certicom-ecc-challenge.html`, since 1997.

[46] Takakazu Satoh, Kiyomichi Araki, et al. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Rikkyo Daigaku sugaku zasshi*, Vol. 47, No. 1, pp. 81–92, 1998.

[47] Igor Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Mathematics of computation*, Vol. 67, No. 221, pp. 353–356, 1998.

[48] Igor A Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. *IACR ePrint Archive 2004/31*, 2004.

[49] Daniel Shanks. Class number, a theory of factorization, and genera. In *Symposia in Pure Mathematics*, Vol. 20, pp. 415–440, 1971.

[50] Joseph H Silverman. The xedni calculus and the elliptic curve discrete logarithm problem. *Designs, Codes and Cryptography*, Vol. 20, No. 1, pp. 5–40, 2000.

[51] Joseph H Silverman. *The Arithmetic of Elliptic Curves*, Vol. 106. Springer Science & Business Media, second edition, 2009.

[52] Nigel P Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of cryptology*, Vol. 12, No. 3, pp. 193–196, 1999.

[53] Edlyn Teske. Speeding up Pollard's rho method for computing discrete logarithms. In *Algorithmic Number Theory (ANTS 1998)*, Vol. 1423 of *Lecture Notes in Computer Science*, pp. 541–554. Springer, 1998.

[54] Edlyn Teske. On random walks for Pollard's rho method. *Mathematics of computation*, Vol. 70, No. 234, pp. 809–825, 2001.

[55] Paul C Van Oorschot and Michael J Wiener. Parallel collision search with cryptanalytic

applications. *Journal of cryptology*, Vol. 12, No. 1, pp. 1–28, 1999.

[56] Lawrence C Washington. *Elliptic Curves: Number Theory and Cryptography*. CRC press, second edition, 2008.

[57] Erich Wenger and Paul Wolfger. Solving the discrete logarithm of a 113-bit Koblitz curve with an FPGA cluster. In *Selected Areas in Cryptography (SAC 2014)*, Vol. 8781 of *Lecture Notes in Computer Science*, pp. 363–379. Springer, 2014.

[58] Michael J Wiener and Robert J Zuccherato. Faster attacks on elliptic curve cryptosystems. In *Selected Areas in Cryptography (SAC 1998)*, Vol. 1556 of *Lecture Notes in Computer Science*, pp. 190–200. Springer, 1998.

[59] Masaya Yasuda, Tetsuya Izu, Takeshi Shimoyama, and Jun Kogure. On random walks of Pollard's rho method for the ECDLP on Koblitz curves. *Journal of Math-for-Industry*, Vol. 3, No. 3, pp. 107–112, 2011.

[60] Masaya Yasuda, Takeshi Shimoyama, Jun Kogure, and Tetsuya Izu. Computational hardness of IFP and ECDLP. *Applicable Algebra in Engineering, Communication and Computing*, Vol. 27, No. 6, pp. 493–521, 2016.

[61] Kazuhiro Yokoyama, Masaya Yasuda, Yasushi Takahashi, and Jun Kogure. Complexity bounds on Semaev's naive index calculus method for ECDLP. *Journal of Mathematical Cryptology*, Vol. 14, No. 1, pp. 460–485, 2020.

[62] 篠原直行, 野呂正行, 横山和弘. 楕円曲線上の離散対数問題に関する指数計算法. CRYPTREC-EX-2602-2016: `https://www.cryptrec.go.jp/exreport/cryptrec-ex-2602-2016.pdf`, 2016.

[63] 富士通株式会社, 株式会社富士通研究所. 楕円曲線暗号と RSA 暗号の安全性比較. `https://www.fujitsu.com/jp/group/labs/documents/resources/tech/external-activities/crypto/eccvsrsa-20100820.pdf`, 2010 年 8 月 20 日.