# ディジタル署名 EdDSA の構成の安全性に関する調査および評価

北陸先端科学技術大学院大学

藤崎　英一郎

2020 年 12 月

# エグゼクティブサマリー

本報告書はディジタル署名 EdDSA の安全性に関する評価結果を報告するものである．EdDSA とは Internet Research Task Force (IRTF) の RFC8032 [61] で規定されるディジタル署名のことであり，有限体上のツイスト Edwards 曲線 [34, 11] といわれる楕円曲線上の Schnorr 署名 [65, 66] の署名内部乱数（ノンス）を署名者の秘密情報と署名される平文のハッシュ値に置き換えた確定版（determinisitic）の Schnorr 署名である．EdDSA で推奨される ツイスト Edwards 曲線は IRTF の RFC7748 [60] で規定されるものであり Ed25519，Ed448 と記述される．以下評価結果の概要を述べる．

- **ツイスト Edwards 曲線の安全性について．** Ed25519 や Ed448 は Curve25519 や Curve448 と同型なため安全性の根拠となる離散対数問題に特に問題があると思えない．
- **Schnorr 署名との違い.** 一番大きな違いは署名内部乱数（ノンス）を署名者の秘密情報と平文のハッシュ値で生成し署名を確定的かつ異なる平文に対してノンスを衝突させにくくしたことである．また内部で使うハッシュ関数の出力長をかなり長くし，群の位数で剰余を取っている，Key-prefixing を採用している（後述），さらに群要素チェックが通常の Schnorr 署名より緩くなっているなどがある．
- **証明可能安全性.** Schnorr 署名をもとに EdDSA 署名は構成されているため，Schnorr 署名に対する安全性評価を利用できるが，Ed25519 ではハッシュ関数に SHA512 を使用しているためその内部の Merkle-Damgård 構造によりランダム関数と識別がつきランダムオラクルモデルでの安全性解析が使えない．また generic group model でも同様に安全性解析をするのも障害がある．一方，Ed448 ではランダムオラクルモデルや generic group model での Schnorr 署名の解析結果が利用でき安全性にある程度の理論的根拠を与えることができる．
- **ビット安全性.** Ed448-EdDSA 署名はランダムオラクルモデルでの解析で 112 ビット程度の安全性，generic group model での解析では 224 ビット程度の安全性が期待できる．
- **Key-prefixing.** EdDSA 署名は key-prefixing という署名者自身の公開鍵を平文と連結させ，公開鍵と平文に署名を付けさせる形を取っている．この仕様のため関連鍵攻撃に対して耐性が上がっている．
- **複数署名者での安全性.** 通常の署名方式は署名者が増えると署名者の数に応じて証明可能安全性で保証できるビット安全性は劣化する．EdDSA 署名は複数署名者でのビット安全性が署名者の数に関係せず，単一署名者の Schnorr 署名のビット安全性で抑えることができる．
- **PureEdDSA と HashEdDSA.** EdDSA で署名するとき，PureEdDSA と HashEdDSA というどちらかのオプションを選ぶ必要がある．HashEdDSA では平文を署名する前にハッシュ関数で圧縮してから署名アルゴリズムに入力する．一方，PureEdDSA は平文を直

接署名アルゴリズムに入力する．HashEdDSA では前処理で平文を圧縮しておくことができるため効率が良いが，ハッシュの衝突耐性以上の安全性を持たない．一方，PureEdDSA はハッシュの衝突耐性以上の安全性を持つ可能性がある．

- **ノンスについて．**Ed25519-EdDSA ではノンスの出力が疑似ランダム関数の出力とみることが出来ないため証明可能安全性の意味では証明がつかなくなっている．しかし，現実の攻撃を考えると異なる平文に対するノンスの衝突こそが 1 番に回避しなければならないものでありハッシュ関数でノンスを生成することでこれを回避している．
- **ECDSA 署名との比較．**安全性において EdDSA 署名が ECDSA 署名に劣ると考えられる点は無いと考えられる．単独の署名生成及び検証の計算時間について比べた時，署名される平文がさほど長くない（平文をハッシュする時間が十分短い）場合 EdDSA 署名が ECDSA 署名よりやや少ない．ただし平文が極めて長い場合，両署名の署名生成時間はほぼハッシュ関数の計算時間となってしまうため，2 回平文のハッシュ値を計算しなければいけない EdDSA 署名の署名生成時間は 1 回のハッシュ値生成で済む ECDSA 署名のほぼ 2 倍かかってしまう．この場合署名検証時間は両方式でほぼ同程度である．単独署名者の複数の署名を検証する場合，EdDSA 署名はバッチ検証処理が使え，同一署名者の場合署名検証をかなり高速にできる．一方，ECDSA 署名は EdDSA 署名ほどの高速化技法は知られていない．
- **サイドチャネル攻撃耐性．**ツイスト Edwards 曲線上の演算は加法と 2 倍算を計算式の切り替えなしに行うことが可能である．一方，より高速に計算するために加法と 2 倍算を別の式で切り替えるやり方も RFC8032 には記載されている．共通式を使うとタイミング攻撃や電力解析攻撃に強くなることが期待できるが，これらの攻撃が本格的にできる環境においてはさらなる対策が必要かもしれない．多くの実装に使用される SUPERCOP [70] の EdDSA 署名は（加法と 2 倍算を切り替える高速版を使った上で）スカラー倍算の加法と 2 倍算の呼び出しが（群位数のサイズの）定数回になるよう実装されており，タイミング攻撃と電力解析攻撃に対する本格的な対策が施されている．近年確定ノンスを使う EdDSA 署名のような確定型署名に対する新たなフォルト攻撃も提案されており，組み込みデバイスとして利用するような場合，将来的にはさらなる対策をとる必要があるかもしれない．

以上の評価により，EdDSA 署名は証明可能安全性という枠組みでは不十分であったり十分なビット安全性が保証されなかったりするが，Schnorr 署名という成熟した方式をもとにノンスの生成で既存の攻撃を注意深く回避する配慮がされており現実的には安全であるという結論を得た．

# 目次


1 はじめに 1

1.1 参考仕様書 . . . . . 1

2 準備 2

3 EdDSA の技術仕様 2

3.1 EdDSA パラメータ . . . . . 3

3.2 推奨パラメータ . . . . . 4

3.3 ツイスト Edwards 曲線上の推奨加法演算 . . . . . 6

3.4 EdDSA 署名アルゴリズム . . . . . 9

3.5 PureEdDSA と HashEdDSA . . . . . 10

3.6 Key-Prefixing と関連鍵攻撃 . . . . . 11

3.7 ノンス $r$ の生成 . . . . . 11

3.8 ハッシュ関数 $H$ の出力長 . . . . . 12

3.9 タイミング攻撃と電力解析攻撃対策 . . . . . 12

3.10 推奨パラメータのツイスト Edwards 曲線について . . . . . 13

4 EdDSA の詳細な安全性評価 13

4.1 署名 . . . . . 13

4.2 離散対数 (DL) 問題 . . . . . 14

4.3 Schnorr 署名 . . . . . 14

4.4 Schnorr 署名から EdDSA 署名への変形 . . . . . 15

4.5 存在的偽造不可能性（EUF-CMA 安全性），存在的強偽造不可能性（sEUF-CMA 安全性） . . . . . 16

4.6 複数署名者版の存在的（強）偽造不可能性（m(s)EUF-CMA 安全性） . . . . . 17

4.7 ハッシュ関数の必要条件 . . . . . 20

4.8 ランダムオラクルモデルでの安全性評価 . . . . . 21

4.9 関連鍵攻撃安全性 . . . . . 23

4.10 Generic Group model での安全性評価 . . . . . 24

5 ECDSA 署名 24

5.1 ECDSA 署名の安全性 . . . . . 26

6 EdDSA 署名と ECDSA 署名の計算量比較 26

6.1 署名生成と検証の計算量比較 . . . . . 26
6.2 バッチ署名検証 . . . . . 27
6.3 楕円曲線上の加法演算の計算量比較 . . . . . 27
6.4 計算量比較まとめ . . . . . 29

7 **サイドチャネル攻撃** 30

8 **まとめ** 32

# 図目次

1 EdDSA . . . . . 9
2 共通パラメータ生成 PGen . . . . . 15
3 Schnorr 署名 . . . . . 15
4 Schnorr 署名から EdDSA 署名への変換 . . . . . 16
5 EUF-CMA 試行 . . . . . 17
6 sEUF-CMA 試行 . . . . . 17
7 mEUF-CMA 試行 . . . . . 18
8 msEUF-CMA 試行 . . . . . 18
9 ECDSA 署名 . . . . . 25

# 表目次

1 Schnorr 署名と EdDSA 署名対応表 . . . . . 16
2 単独署名生成と署名検証の計算量比較 . . . . . 26
3 楕円曲線上の加法の方式による比較 . . . . . 29

# 1 はじめに

本報告書ではディジタル署名 EdDSA の安全性評価を行いその結果を報告する．

EdDSA とは Internet Research Task Force (IRTF) の RFC8032 で規定されるディジタル署名のことであり，ツイスト Edwards 曲線 [34] といわれる有限体上の楕円曲線表現を用いた Schnorr 署名 [65, 66] の署名内部乱数（ノンス）を署名者の秘密情報と署名される平文のハッシュ値に置き換えた確定的 (determinisitic) 版の Schnorr 署名である．EdDSA で使用される Edwards 曲線は IRTF の RFC7748 で規定される Edwards 曲線 Ed25519 および Ed448 からなる．RFC8032 の EdDSA は文献 [11, 12] にもとづいて記述されている．

本報告書の構成について述べる．1 章で必要な記法の準備を行い，3 章で EdDSA 署名の仕様について記述する．4 章で安全性評価を記述し，5 章で ECDSA 署名の安全性について知られている事実をまとめた後，6 章で EdDSA 署名と ECDSA 署名の計算量比較を行い，7 章で知られているサイドチャネル攻撃について記述する．最後に 8 章で EdDSA 署名についての見解をまとめる．

## 1.1 参考仕様書

本報告書の EdDSA，推奨曲線，ハッシュ関数，ECDSA の仕様は以下の仕様書の記述を正式なものとした．

- [RFC8032] Internet Research Task Force (IRTF) Request for Comments: 8032. `https://www.rfc-editor.org/info/rfc8032`.
- [RFC7748] Internet Research Task Force (IRTF) Request for Comments: 7748. `https://www.rfc-editor.org/info/rfc7748`.
- [RFC6234] Internet Research Task Force (IRTF) Request for Comments: 6234. `https://www.rfc-editor.org/info/rfc6234` (SHA512).
- [FIPS202] SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions.
  `https://csrc.nist.gov/publications/detail/fips/202/final`
- [SEC1] SEC 1: Elliptic Curve Cryptography (September 20, 2000 Version 1.0)
  `https://www.secg.org/SEC1-Ver-1.0.pdf`
- [SEC2] SEC 2: Recommended Elliptic Curve Domain Parameters (January 27, 2010 Version 2.0)
  `https://www.secg.org/sec2-v2.pdf`

## 2 準備

本報告書で用いる数学的記法を準備する．

$\{0,1\}^*$ で任意の有限ビット列の集合を表す．$x \in \{0,1\}^*$ に対して $|x|$ は $x$ のビット長を意味する．$x, y \in \{0,1\}^*$ に対して $(x, y)$ または $x|y$ は $x$ と $y$ の連結とする．$x \in \{0,1\}^*$ に対して $\mathsf{msb}(x)$ で $x$ を整数の 2 進展開と考えたときの $2^n$（$n := |x| - 1$）の係数を表し，$\mathsf{lsb}(x)$ で $2^0$ の係数を表す．

オクテット列とは 256 進数表現で表された系列である．$8d$ ビット長の $x$ はオクテット列では $d$ オクテット列になる．$\mathsf{octlen}(x)$ で $x$ のオクテット長を表す．

集合 $S$ に対して $\#S$ は集合 $S$ に含まれる元の個数を意味する．$A := B$ で代入もしくは $A$ を $B$ と定義する意味で使う．

確率的アルゴリズム $A$ に対して $y \leftarrow A(x)$ で $A$ が入力 $x$ をとり $y$ を出力する試行 (experiment) と定義する．特に $A$ の内部乱数 $r$ を明示すると $A$ は関数となるので $y = A(x; r)$ と表現できる．集合 $S$ に対して $s \leftarrow S$ で集合 $S$ から一様ランダムに元を選び結果 $s$ を出力した試行 (experiment) と定義する．

関数 $p : \mathbb{N} \to \mathbb{R}^+$ がある定数 $c > 0$ に対して $p(\kappa) = O(\kappa^c)$ のとき $p$ は多項式制限といい $p(\kappa) = \mathrm{poly}(\kappa)$ と表す．関数 $\epsilon : \mathbb{N} \to \mathbb{R}^+$ が $\epsilon(\kappa) = \kappa^{-\omega(1)}$ のとき $\epsilon$ を無視できる関数と定義し，$\epsilon(\kappa) = \mathsf{negl}(\kappa)$ と表す．

$\mathbb{Z}/\ell\mathbb{Z}$ で $\{0, 1, \ldots, \ell-1\}$ を代表元とする剰余類環を表す．加法群 $\mathbb{G}$ の元 $A \in \mathbb{G}$ と整数 $x$ に対して，$x \geq 0$ であれば $xA := \overbrace{A + \cdots + A}^{x}$ を，$x < 0$ であれば $xA := \overbrace{(-A) + \cdots + (-A)}^{|x|}$ をそれぞれ意味する．$A \in \mathbb{G}$ に対して $\langle A \rangle := \{rA \,|\, r \in \mathbb{Z}\}$ は $A$ で生成される巡回群を表す．$\mathbb{G}$ の単位元（零元）は 0 で通常表す．ただし楕円曲線上の群を意識しているときは $O$ を使うこともある．$\ell := \#\mathbb{G}$ を群 $\mathbb{G}$ の位数という．$\langle A \rangle := \mathbb{G}$ で $\mathbb{G}$ の位数が（有限の）$\ell$ であれば $\langle A \rangle = \{rA \,|\, r \in \mathbb{Z}/\ell\mathbb{Z}\}$ で $\ell A = 0$ である．

有限体（素体）$\mathbb{F}_p$ の元を報告書内では $0 \leq x < p - 1$ までの１０進整数で表し，$x, y \in \mathbb{F}_p$（ただし $y \neq 0$）に対して $x/y$ は $yz \equiv x \pmod{p}$ なる $z$ を表している．

## 3 EdDSA の技術仕様

EdDSA 署名の技術仕様を RFC8032 に従って紹介する．EdDSA 署名は平文を署名アルゴリズムに入力する前にハッシュするか否かのオプションがあり，事前にハッシュしないものを PureEdDSA と呼び，事前にハッシュするものを HashEdDSA と呼ぶことになっている．また，ハッシュ関数に署名者と検証者の間で合意した任意のビット列 context を入れるオプションも存在する．

## 3.1 EdDSA パラメータ

- 奇素数 $p$: EdDSA は $\mathbb{F}_p$ 上の（ツイスト）Edwards 曲線を使う．
- 正整数 $b$: $p < 2^{b-1}$ なる正整数．EdDSA の公開鍵長であり 8 の倍数が推奨されている．
- エンコーディング関数 $E' : \mathbb{F}_p \to \{0,1\}^{b-1}$: $\mathbb{F}_p$ の元の $(b-1)$ ビット表現を定める．
- ハッシュ関数 $H : \{0,1\}^* \to \{0,1\}^{2b}$: $2b$ ビット長の出力を出すハッシュ関数．
  - Ed25519 では SHA512 [RFC6234]
  - Ed448 では SHAKE256 [FIPS202]

  が規定されている．
- $(a, d, c, \ell)$: （ツイスト）Edwards 曲線 $E_{(p,a,d)}$ を決定するパラメータ．

$$E_{(p,a,d)} := \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p \,|\, ax^2 + y^2 = 1 + dx^2y^2\}$$

  - $a$ は $\mathbb{F}_p$ 上平方剰余，$d$ は非ゼロの非剰余．すなわち $\left(\frac{a}{p}\right) = 1$, $\left(\frac{d}{p}\right) = -1$.
  - $c = 2$ または 3．$\ell$ は奇素数で，$E_{(p,a,d)}$ の位数 $\#E_{(p,a,d)} = 2^c\ell$ となるような数．
- $n$: $c \le n < b$ なる整数 [*1]．EdDSA の署名秘密鍵の一部 $s$ のビット長は $n+1$.
- ベースポイント $B \in \mathbb{F}_p \times \mathbb{F}_p$: $B \neq (0,1)$ かつ $\ell B = (0,1)$.
- プレハッシュ関数 PH: PureEdDSA と HashEdDSA の場合で異なる．
  - PureEdDSA の場合，$\mathsf{PH}(m) := m$.
  - HashEdDSA の場合，$\mathsf{PH}(m) :=$ “$m$ のハッシュ値”.

（ツイスト）Edwards 曲線の加法は次のように定義される．

$$(x_1, y_1) + (x_2, y_2) := \left(\frac{x_1y_2 + x_2y_1}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2}\right) \tag{1}$$

この演算により $E_{(p,a,d)}$ は $O := (0,1)$ を単位元とする加法群になる．

**注釈 1**　$a = 1$ のとき $E_{(p,a,d)}$ は Edwards 曲線 [34]．$a \neq 1$ も含めたとき $E_{(p,a,d)}$ は一般にツイスト Edwards 曲線と呼ばれる [11, 10].

**注釈 2**　署名生成及び検証の速度面から $p \equiv 1 \pmod 4$ のときは $a = -1$ が，$p \equiv 3 \pmod 4$ のときは $a = 1$ が推奨されている．$\left(\frac{a}{p}\right) = a^{(p-1)/2} \bmod p$ より，$p \equiv 1 \pmod 4$ のとき $\left(\frac{-1}{p}\right) = 1$, $p \equiv 3 \pmod 4$ のとき $\left(\frac{-1}{p}\right) = 1$.

**注釈 3**　ツイスト Edwards 曲線では（他の曲線表現での加法と異なり）任意の $P, Q \in E_{(p,a,d)}$ に対して式 (1) で加法 $P + Q$ が計算できる（完全性）．加法と 2 倍算で式が同じ，零元 $O = (0,1)$ も

[*1] RFC8032 の記述に従ったが $c = n$ は安全性の観点からありえない．

扱える．

**注釈 4**　重い逆数演算を避けるため 3.3, 6 章のような（拡張した）射影座標系に直して計算することが推奨されている．

### 3.1.1　エンコーディング

RFC8032 の 3 章では（ゼロ以上の）整数はビット列としてリトルエンディアンでエンコードする．$E_{(p,a,d)}$ の元 $(x,y)$ を $y$ を $(b-1)$ ビット長にエンコードし，$x<0$ が負なら 1 を連結，$x\geq 0$ なら 0 を連結する．すなわち $E'(y)|b' \in \{0,1\}^b$．ここで $x\in\mathbb{F}_p$ が $x<0$ とは，$E'(x)$ が $E'(-x)$ より辞書的順序で大きいときを意味する．このように Edwards 曲線 $E_{(p,a,d)}$ の元は $b$ ビット長で表現される．このことから $\#E_{(p,a,d)} < 2^b$ なので $\ell < 2^b$ である．実際 $2^{b-c-1} < \ell < 2^{b-c}$ $(c=2,3)$.

なお RFC8032 の推奨パラメータの 5.1 と 5.2 章ではオクテット列とみなしてリトルエンデイアンでエンコードと記述してある．

以後，エンコーディングについては安全性の評価に関係しないので記述は基本省略する．

## 3.2　推奨パラメータ

一般に安全性は推奨パラメータに大きく依存する．RFC8032 では二つのパラメータを推奨パラメータとして紹介している．

### 3.2.1　Ed25519

- $p = 2^{255}-19$
- $b = 256$
- $H$: $H(x) := \mathsf{SHA512}(\mathsf{dom2}(\mathsf{phflag},\mathsf{context}),x)$ [RFC6234] （後述）.
- $c = 2$
- $n = 254$
- $d =$ edwards25519 [RFC7748] で定義された $\mathbb{F}_p$ 上の次の値

$$\frac{-121665}{121666370957059346694393431380835087545651895421138798432190163887855330859402835 55}-1.$$

- $a = -1$
- $B = (x,y)$: edwards25519 [RFC7748] で定義された $(\mathbb{F}_p)^2$ 上の次の値

$$x = 15112221349535400772501151409588531511454012693041857206046113283949847762202$$
$$y = 46316835694926478169428394003475163141307993866256225615783033603165251855960$$

- $\ell = 2^{252} + 27742317777372353535851937790883648493$

- PH: PureEdDSA の場合 $\mathsf{PH}(m) = m$. HashEdDSA の場合 $\mathsf{PH}(m) = H(m)$.

$H(x)$ と $\mathsf{PH}(x)(= H(x))$ の出力長は $2b = 512$ ビット.

■Ed25519, Ed25519ctx, Ed25519ph Ed25519 は 3 つのオプションがある. Ed25519 と Ed25519ctx は PureEdDSA（すなわち $\mathsf{PH}(m) = m$）のパラメータで, Ed25519ph は HashEdDSA（すなわち $\mathsf{PH}(m) = H(m)$）のパラメータである.

- Ed25519: $\mathsf{dom2}(\mathsf{phflag}, \mathsf{context}) = \varepsilon$. すなわち $x$ の直前に何も連結されない. また context も存在しない.
- Ed25519ctx: phflag := 0 で context は署名者と検証者で予め合意された空でないオクテットストリングである.
- Ed25519ph: phflag := 1 で context は署名者と検証者で予め合意されたもので空であってもそうでなくても良い.

$\mathsf{dom2}(x, y)$ は Ed25519 の時は空であり, それ以外の時は

$$\text{“SigEd25519 no Ed25519 collisions”} \,|\, x \,|\, \mathsf{octlen}(x) \,|\, y$$

のオクテット表現列である（$\mathsf{octlen}(x)$ は $x$ のオクテット列長）. $x, y$ は最大 255 オクテットまで許される.

### 3.2.2 Ed448

- $p = 2^{448} - 2^{224} - 1$
- $b = 456$
- $H$: $H(x) := \mathsf{SHAKE256}((\mathsf{dom4}(\mathsf{phflag}, \mathsf{context}), x) \,|\, 114)$ [FIPS202]（後述）.
- $c = 2$
- $n = 447$
- $d = -39081$
- $a = 1$
- $B = (x, y)$: edwards448 [RFC7748] で定義された $(\mathbb{F}_p)^2$ 上の次の値

  $x = 224580040295924300187604334099896036246789641632564134246125461686950415467406032909029192869357953282578032075146446173674602635247710$

  $y = 298819210078481492676017930443930673437544040154080242095928241372331506189835876003536878655418784733982303233503462500531545062832 60$

- $\ell = 2^{446} - 138180668098951153520073867485154268803366924748821786098945475038 85$
- PH: PureEdDSA の場合 $\mathsf{PH}(m) = m$. HashEdDSA の場合 $\mathsf{PH}(m) = \mathsf{SHAKE256}(m \,|\, 64)$（後述）.

SHAKE256$(\cdot|y)$ は SHAKE256 の出力の最初の $y$ オクテットの出力を意味する．よって $H(x)$ は $912(= 114 \cdot 8 = 2 \cdot 456)$ ビット長で，PH$(x)$ は $512(= 64 \cdot 8)$ ビット．

■Ed448，Ed448ph　Ed448 は 2 つのオプションがある．Ed448 は PureEdDSA（すなわち PH$(m) = m$）のパラメータで，Ed448ph は HashEdDSA（すなわち PH$(m) =$ SHAKE256$(m|64)$）のパラメータである．

- Ed448: dom4(phflag, context) $= \varepsilon$．すなわち $x$ の直前に何も連結されない．また context も存在しない．
- Ed448ph: phflag $:= 1$ で context は署名者と検証者で予め合意されたもので空であってもそうでなくても良い．

dom4$(x, y)$ は Ed448 の時は空であり，Ed448ph 時は

$$\text{“SigEd448”} \,|\, x \,|\, \mathsf{octlen}(x) \,|\, y$$

のオクテット表現列である（octlen$(x)$ は $x$ のオクテット列長）．$x, y$ は最大 255 オクテットまで許される．

## 3.3　ツイスト Edwards 曲線上の推奨加法演算

楕円曲線上の実際の演算はアフィン座標系から他の座標系に変更することで $\mathbb{F}_p$ 上の逆算をさけ計算量を削減する．RFC8032 には文献 [10] が示したツイスト Edwards 曲線の射影座標表現での加法と 2 倍算のアルゴリズムを推奨演算として記載している．

以下，有限体 $\mathbb{F}_p$ 上の二つの元の掛け算の計算量を $\mathbf{M}_p$，有限体 $\mathbb{F}_p$ 上の元の 2 乗演算の計算量を $\mathbf{S}_p$，有限体 $\mathbb{F}_p$ 上の元と定数 $d$ との掛け算の計算量を $\mathbf{D}_p$ とする．有限体 $\mathbb{F}_p$ 上の元と定数 $a$ との掛け算は $a = -1$ (Ed25519), $a = 1$ (Ed448) と小さいので無視できる．有限体 $\mathbb{F}_p$ 上の加算や減算の計算量は乗算や 2 乗演算より十分小さいので無視する．

**注釈 5**　ツイスト Edwards 曲線では加法と 2 倍算に共通の公式 (1) が使えるため，下記加法の計算式は 2 倍算にも使える．2 倍算に特化した計算式は加法のものより高速であるが，サイドチャネル攻撃を考慮する環境での署名生成アルゴリズムは，（定数時間実装などの防御 [70] を行わない場合は）2 倍算に対しても加法の計算式を使うのが望ましいと考えられる（署名検証においてその必要はない）．

### 3.3.1　Ed25519 の場合

ツイスト Edwards 曲線 $ax^2 + y^2 = 1 + dx^2y^2$ の点 $(x, y) \in E_{(p,a,d)}$ を $(aX^2 + Y^2)Z^2 = Z^4 + dX^2Y^2$ を満たす（拡張）射影座標 $(X : Y : Z : T)$ で表す．$(x, y) \in E_{(p,a,d)}$ と $(X : Y : Z : T)$

は $x = X/Z$, $y = Y/Z$, $xy = T/Z$ という対応関係があり，$\lambda \in \mathbb{F}_p^\times$ なる全ての $\lambda$ に対して $(X : Y : Z : T) \sim (\lambda X : \lambda Y : \lambda Z : \lambda T)$（同じ点と見なす）．Ed25519 の場合 $a = -1$ であり以下のアルゴリズムは $a = -1$ という特徴に特化して計算量を削減したアルゴリズムになっている．

加法と 2 倍算は以下のアルゴリズムで計算できる．

- 加法 $(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$ の場合，
  1. $A := (Y_1 - X_1) \cdot (Y_2 - X_2)$ [*2]
  2. $B := (Y_1 + X_1) \cdot (Y_2 + X_2)$ [*3]
  3. $C := T_1 \cdot 2d \cdot T_2$
  4. $D := Z_1 \cdot 2Z_2$
  5. $E := B - A$
  6. $F := D - C$
  7. $G := D + C$
  8. $H := B + A$
  9. $X_3 := E \cdot F$
  10. $Y_3 := G \cdot H$
  11. $T_3 := E \cdot H$
  12. $Z_3 := F \cdot G$
- 2 倍算 $(X_3 : Y_3 : Z_3) = 2 \cdot (X_1 : Y_1 : Z_1)$ の場合．
  1. $A := X_1^2$
  2. $B := Y_1^2$
  3. $C := 2 \cdot Z_1^2$
  4. $H := A + B$
  5. $E := H - (X_1 + Y_1)^2$
  6. $G := A - B$
  7. $F := C + G$
  8. $X_3 := E \cdot F$
  9. $Y_3 := G \cdot H$
  10. $T_3 := E \cdot H$
  11. $Z_3 := F \cdot G$

よって加法の計算量は，$9\mathbf{M}_p + 1\mathbf{D}_p$ となる．また 2 倍算の計算量は $4\mathbf{M}_p + 4\mathbf{S}_p$ である．

[*2] $(Y_1 - X_1) \cdot (Y_2 - X_2) = X_1X_2 + Y_1Y_2 - (X_1Y_2 + X_2Y_1)$.
[*3] $(Y_1 + X_1) \cdot (Y_2 + X_2) = X_1X_2 + Y_1Y_2 + X_1Y_2 + X_2Y_1$.

### 3.3.2 Ed448 の場合

ツイスト Edwards 曲線 $ax^2 + y^2 = 1 + dx^2y^2$ の点 $(x, y) \in E_{(p,a,d)}$ を $(aX^2 + Y^2)Z^2 = Z^4 + dX^2Y^2$ を満たす射影座標 $(X : Y : Z)$ で表す．$(x, y) \in E_{(p,a,d)}$ と $(X : Y : Z)$ は $x = X/Z$, $y = Y/Z$ という対応関係があり，$\lambda \in \mathbb{F}_p^\times$ なる全ての $\lambda$ に対して $(X : Y : Z) \sim (\lambda X : \lambda Y : \lambda Z)$（同じ点と見なす）．Ed448 の場合 $a = 1$ であるがツイスト Edwards 曲線全般に適用できるよう Ed25519 のときと異なり $a$ を省略せず記述する．

加法と 2 倍算は以下のアルゴリズムで計算できる．

- $(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$ の場合,
  1. $A := Z_1 \cdot Z_2$
  2. $B := A^2$
  3. $C := X_1 \cdot X_2$
  4. $D := Y_1 \cdot Y_2$
  5. $E := d \cdot C \cdot D$
  6. $F := B - E$
  7. $G := B + E$
  8. $X_3 := A \cdot F \cdot \Big((X_1 + Y_1) \cdot (X_2 + Y_2) - C - D\Big)$ [*4]
  9. $Y_3 := A \cdot G \cdot (D - aC)$
  10. $Z_3 := F \cdot G$
- $(X_3 : Y_3 : Z_3) = 2 \cdot (X_1 : Y_1 : Z_1)$ の場合.
  1. $B := (X_1 + Y_1)^2$
  2. $C := X_1^2$
  3. $D := Y_1^2$
  4. $E := a \cdot C$
  5. $F := E + D$
  6. $H := Z_1^2$
  7. $J := F - 2H$
  8. $X_3 := (B - C - D) \cdot J$
  9. $Y_3 := F \cdot (E - D)$
  10. $Z_3 := F \cdot G$

加法の計算量は $10\mathbf{M}_p + 1\mathbf{S}_p + 1\mathbf{D}_p$ となる．ただし Ed448 の場合 $d = -39081$ と小さいので $\mathbf{D}_p \approx 0$ である．また $a = 1$ なので $E = a \cdot C$ の計算量は無視して良い．すると 2 倍算の計算量は

[*4] $(X_1 + Y_1) \cdot (X_2 + Y_2) - C - D = X_1Y_2 + X_2Y_1$.

$3\mathbf{M}_p + 4\mathbf{S}_p$ となる．

## 3.4 EdDSA 署名アルゴリズム

EdDSA 署名は次のように定義される．EdDSA のシステム共通のパラメータは 3.1 章のように $\mathsf{para} := \{b, p, a, d, c, \ell, n, B, H, E', \mathsf{PH}\}$，ただし

$$E_{(p,a,d)} := \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p \,|\, ax^2 + y^2 = 1 + dx^2y^2\} \tag{2}$$

と定義する．$H : \{0,1\}^* \to \{0,1\}^{2b}$ の出力長は $2b$ ビットである．また

$$p < 2^{b-1}, \quad c < n < b \quad (c = 2 \text{ or } 3), \quad 2^{b-c-1} < \ell < 2^{b-c} \tag{3}$$

が成立している．

このとき EdDSA 署名 $\mathsf{EdDSA} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ は図 1 のようなアルゴリズムの組である．

| $(vk, sk) \leftarrow \mathsf{KGen}(\mathsf{para})$. | $\sigma \leftarrow \mathsf{Sign}_{sk}(vk, m)$. | $\tau \leftarrow \mathsf{Vrfy}(vk, m, \sigma)$. |
|---|---|---|
| input: para | input: $(sk, vk, m)$ | input: $(vk, m, \sigma)$ |
| $s_\epsilon \leftarrow \{0,1\}^b$; | parse $sk = (s, s')$; | parse $\sigma = (R, z)$; |
| $(h_0 \dots h_{2b-1}) := H(s_\epsilon)$; | $r := H(s', m)$; | if $A, R \notin E_{(p,a,d)}$ |
| $s := 2^n + \sum_{i=c}^{n-1} h_i \cdot 2^i$; | $R := rB$; | retun $\tau := 0$; |
| $s' := (h_b \dots h_{2b-1})$; | $e := H(R, A, \mathsf{PH}(m))$; | else |
| $A := sB$; | $z := r + es \bmod \ell$; | $e := H(R, A, \mathsf{PH}(m))$; |
| $vk := A$; | $\sigma := (R, z)$; | if $(2^c z)B = 2^c R + (2^c e)A$ |
| $sk := (s, s')$; | | set $\tau := 1$; |
| return $(vk, sk)$. | return $\sigma$. | else $\tau := 0$; |
| | | return $\tau$. |

図 1　EdDSA

- 鍵生成アルゴリズム KGen は $b$ ビット長の乱数 $s_\epsilon$ を発生させこれをハッシュ関数 $H$ により $2b$ ビット列 $(h_0, \dots, h_{2b-1}) := H(s_\epsilon)$ を生成する．$s$ は $s := 2^n + \sum_{i=c}^{n-1} h_i$ と生成される最上位ビットに 1 が立ち，下位 $c$ ビットが 0 の $(n+1)$ ビット長の秘密情報．$s'$ は $s' := (h_b, \dots, h_{2b-1})$ なる $b$ ビット長の秘密情報であり署名秘密鍵 $sk := (s, s')$ は $(b + n + 1)$ ビット長となる．一方 $vk := A(= sB)$ なので $E_{(p,a,d)}$ の元は $b$ ビットでエンコードできるので署名検証鍵は $b$ ビット長になる．
- 署名生成アルゴリズム Sign は Schnorr 署名（4.3 章）とほぼ同等だが，Schnorr 署名と異なり署名内部乱数 $r$ をランダムに発生させる代わりに秘密情報 $s'$ と署名対象平文 $m$ のハッシュ値として $r := H(s', m)$ のように生成される．その後署名検証鍵 $vk(= A)$ を

$\mathsf{PH}(m)$ と連結 (key-prefixing) させてからハッシュ関数に含め $e := H(R, A, \mathsf{PH}(m))$ を計算する（$R := rB$）．$\mathsf{PH}(m)$ は，PureEdDSA の場合は恒等関数で平文 $m$ そのものであり，HashEdDSA の場合は $H'(m)$ なるあるハッシュ関数 $H'$ の出力値である．最後に $z := r + es \bmod \ell$ と通常の Schnorr 署名と同様の計算を行う．$B$ の作る巡回群 $\langle B \rangle$ の位数 $\ell$ が $(b-c)$ ビット長（$c = 2, 3$）より $r$ も $e$ も $\ell$ の 2 倍程度のビット長であることに注意．

- 署名検証アルゴリズム Vrfy は Schnorr 署名の署名検証と同等だが RFC8032 では署名 $(R, z)$ を検証するのに $zB = R + eA$ のかわりに $(2^c z)B = 2^c R + (2^c e)A$ を用いる．$\langle B \rangle$ は素数位数 $\ell$ の巡回群であり正当な署名者が署名を作った場合 $A, R \in \langle B \rangle$ であるから $zB = R + eA$ も成立しているが $(2^c z)B = 2^c R + (2^c e)A$ の検証式を使う．

署名生成アルゴリズム Sign と署名検証アルゴリズム Vrfy は厳密には para の情報が必要だが最初からアルゴリズムに含まれているものとして省略した．

## 3.5 PureEdDSA と HashEdDSA

$\mathsf{PH}(m) = m$ のとき PureEdDSA と呼ぶ．一方，あるハッシュ関数 $H'$ により $\mathsf{PH}(m) = H'(m)$ のとき HashEdDSA と呼ぶ．推奨パラメータ Ed25519 と Ed448 のオプション Ed25519ctx, Ed25519ph, Ed448ph では context と呼ばれる署名者と検証者で予め合意しておく情報が必要になるが今回の安全性評価には関係しないので以後無視する．

RFC8032 では PureEdDSA は衝突耐性 (collision resistance) があると述べられているが，これはハッシュ関数 $H$ に衝突が見つかったとしても PureEdDSA の偽造署名（定義 10 参照）が作れるわけでは無いことからそのように呼んでいる

Neven ら [53] は Schnorr 署名が偽造署名が作られないための（衝突耐性より弱い）ハッシュ関数 $H$ の必要条件としてランダムプレフィクス原像計算困難性 (random-prefix preimage resistant (RPP) ) とランダムプレフィクス第 2 原像計算困難性 (random-prefix second-preimage resistant (RPSP)) という性質を考えた．さらに彼らは generic group model でハッシュ関数が RPP 安全かつ RPSP 安全であれば Schnorr 署名は EUF-CMA 安全であることを証明している．PureEdDSA においても $r = H(s', m)$ が $s'$ を鍵とする疑似ランダム関数の出力であるか $H$ がランダム関数であるか仮定すれば同様の結果が成り立つ．

HashEdDSA は $\mathsf{PH}(m) = \mathsf{PH}(m')$ なる $m, m'$（$m \neq m'$）の衝突を見つければ存在的偽造が可能なため衝突耐性は無い．

**注釈 6**　署名者が PureEdDSA と HashEdDSA を同じ鍵で利用する場合，HashEdDSA の署名は PureEdDSA の平文 $\mathsf{PH}(m)$ の署名になるため注意が必要である．

## 3.6 Key-Prefixing と関連鍵攻撃

通常の Schnorr 署名と異なり EdDSA 署名では平文の直前に署名者の署名検証鍵 $A$ を連結させ $(A, m)$（HashEdDSA の場合は $(A, H'(m))$）をハッシュ関数に入力させる．これを key-prefixing と呼ぶ．これは**関連鍵攻撃 (related-key attack)** を防ぐ役割を果たしている．

key-prefixing が無いと仮定する．今 $C \in E_{(p,a,d)}$ を位数 $2^c$ の元とする．署名検証鍵 $A$ を持つ署名者が平文 $m$ の署名 $(R, z)$ を生成したとする．そのとき署名 $(R, z)$ は新しい（攻撃者が選んだ）署名検証鍵 $A + C$ における平文 $m$ の署名になっている（なぜならば $(2^c e)(A + C) = (2^c e)A$）．もし $A, R \in E_{(p,a,d)}$ というチェックのかわりに $A, R \in \langle B \rangle$ というチェックをしていればこの攻撃は元々無効であるが実際 EdDSA では前者のチェックを行っている．key-prefixing を行わない場合，実はこのチェックを行っていても関連鍵攻撃をすることが可能である [50].

森田らの結果 [50] を援用すると，もとの署名検証鍵を $A$ としたとき，変更された署名検証鍵がある多項式 $f$ により $f(A)$ と変更される範囲においては key-prefixing をすることで攻撃を防御できることが期待できる．

**注釈 7** EdDSA では key-prefixing を行っていても co-factor 部分の遊びにより正規の署名者は本来の署名方式を逸脱して $A + C$ を署名検証鍵に登録したり署名時に $R = rB + C$ を使って署名をすることが可能である．これは攻撃と言えないかもしれないが注意しておくべきことと考える．

## 3.7 ノンス $r$ の生成

通常の Schnorr 署名と異なり EdDSA 署名ではノンス $r$ を秘密情報 $s'$ と平文のハッシュ値 $H(s', m)$ により生成する．この狙いは弱い疑似乱数生成によるノンス $r$ の衝突や推測を避けるためである．

もし $r$ の衝突が異なる二つの平文の署名に対して起これば Schnorr 署名および EdDSA 署名の署名鍵 $s$ が簡単に計算できてしまう[*5].

同様に $r$ を推測できても署名鍵 $s$ は計算できてしまう．EdDSA 署名に対するサイドチャネル攻撃には，偏った出力のノンス $r$ の署名を集めることでノンスを計算し署名鍵を復元するというものがある．

よってノンス $r$ の衝突や推測可能性は EdDSA 署名（と Schnorr 署名）に致命的であるため十分に気を付ける必要がある．

証明可能安全性の文脈で言うと $H(s', \cdot)$ が $s'$ を秘密シードとする疑似ランダム関数であるとみなせるのであれば，EdDSA 署名の安全性は真の乱数 $r$ を利用した EdDSA 署名の安全性に帰着で

[*5] 実際，過去に脆弱な疑似乱数生成のためノンスに衝突がおき Sony PlayStation3 の ECDSA 署名の署名鍵が抜き取り可能であることが示されたことがある．

きる．

推奨パラメータ Ed448 では $H$ に SHA-3 ハッシュ関数である SHAKE256 が使われており $s'$ を乱数とみなせるのであれば疑似ランダム関数の出力とみなして良いと思われる．一方推奨パラメータ Ed25519 の場合，$H$ は SHA512 である．SHA512 は Merkle-Damgård 構造 [49, 31] を使っているため length-extension 攻撃によりランダム関数と容易に識別可能なため疑似ランダム関数にはなり得ない．このため Ed25519-EdDSA 署名の安全性は真の乱数 $r$ を利用した EdDSA 署名の安全性に帰着することはできない．ただしこれは Ed25519-EdDSA 署名が安全でないということをただちに意味しているものではない．ノンス $r$ は公開ではなく $R = rB$ の離散対数問題を解読しなければわからない．

$s'$ は $b$ ビット長の乱数 $s_\epsilon$ のハッシュ関数 $H$ の出力値 $H(s_\epsilon)$ の右から $b$ ビットを選んだ値である（$H$ の出力長は $2b$ ビット）．$s_\epsilon$ をどう選ぶかのこれ以上詳しい言及は RFC8032 にはない．

## 3.8 ハッシュ関数 $H$ の出力長

ハッシュ関数 $H$ の出力長は $|\ell|(= b - c)$ ビットの 2 倍強の $2b$ ビットである．$r \bmod \ell$（または $e \bmod \ell$）が衝突が起きた場合安全性に問題が生じる．ただしこれはハッシュ関数 $H$ の出力を切り捨て (truncated output)$|\ell|$ ビット長にした場合とどちらが衝突が起きやすいかは判断出来ない．

例えば，Ed448 の場合 $H = \mathsf{SHAKE256}$ をランダム関数とみなしても良いと思われるがその時 $r \bmod \ell$ と $e \bmod \ell$ は共に $2^b$ 回の折り返しがあるので $\mathbb{Z}/\ell\mathbb{Z}$ 上の一様乱数との統計的識別距離は $2^{-b}$ で抑えられるため $\mathbb{Z}/\ell\mathbb{Z}$ に出力するランダム関数とみなし安全性を考察することができる．

## 3.9 タイミング攻撃と電力解析攻撃対策

ツイスト Edwards 曲線では任意の $P, Q \in E_{(p,a,d)}$ に対して式 (1) で加法 $P + Q$ が計算できる（完全性）ので加法と 2 倍算で共通の公式が使える．これは署名生成アルゴリズムが $R = rB$ を計算する際，加法と 2 倍算の切り替えによる違いをわかりにくくすることが出来，タイミング攻撃や電力解析攻撃でノンス $r$ を推定することを難しくする利点がある．一方，より高速に計算するために加法と 2 倍算を別の式で切り替えるやり方も RFC8032 には記載されている．こちらを使う場合タイミング攻撃や電力解析攻撃が可能でない環境で使用するよう注意するか，実装になんらかの防御手段を入れることが望ましい．

SUPRECOP [70] の実装では，（加法と 2 倍算を切り替える高速版を使っているが）スカラー倍算 $R = rB$ の加法と 2 倍算の呼び出しが（$|r|$ の）定数回になるよう対策が施されているため，タイミング攻撃や電力解析攻撃に強くなっている．

### 3.10 推奨パラメータのツイスト Edwards 曲線について

Ed25519 および Ed448 で規定されるツイスト Edwards 曲線は RFC7748 で規定される Montgomery 曲線 Curve25519 と Curve448 とそれぞれ（同一の有限体上の群として）同型写像が存在する曲線である．よって RFC8032 の推奨のツイスト Edwards 曲線上の離散対数問題は RFC7748 で推奨の曲線上の離散対数問題と同程度に難しい．

## 4 EdDSA の詳細な安全性評価

本章では EdDSA 署名の安全性の評価を考察する．比較対象になる Schnorr 署名の定義や安全性のクラスの定義などをまず行い，その後 Schnorr 署名と比較する形で EdDSA 署名の安全性を評価していく．

### 4.1 署名

$\mathsf{SIG} = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ をセキュリティパラメータ $\kappa \in \mathbb{N}$ に依存するアルゴリズムの組とし次のように定義する：

- システムパラメータ生成アルゴリズム $\mathsf{PGen}$: $1^\kappa$ を入力としてとり，システムパラメータ $\mathsf{para}$ を出力する多項式時間アルゴリズム．この試行を $\mathsf{para} \leftarrow \mathsf{PGen}(1^k)$ と書く．
- 鍵生成アルゴリズム $\mathsf{KGen}$: システムパラメータ $\mathsf{para}$ を入力としてとり，検証鍵と署名鍵 $(vk, sk)$ を出力する確率的多項式時間アルゴリズム．この試行を $(vk, sk) \leftarrow \mathsf{KGen}(\mathsf{para})$ と書く．
- 署名生成アルゴリズム $\mathsf{Sign}$: 署名鍵 $sk$，検証鍵 $vk$，平文 $m \in \{0,1\}^*$ を入力としてとり，署名 $\sigma$ を出力する確率的多項式時間アルゴリズム．この試行を $\sigma \leftarrow \mathsf{Sign}(sk, vk, m)$ と書く．
- 署名検証アルゴリズム $\mathsf{Vrfy}$: 検証鍵 $vk$，平文と署名の組 $(m, \sigma)$ を入力としてとり，承認の場合 1 を拒絶の場合 0 を出力する確定的多項式時間アルゴリズム．この試行を $b \leftarrow \mathsf{Vrfy}(vk, m, \sigma)$ と書く（$b \in \{0,1\}$）．

**定義 8** $\mathsf{SIG} = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ が，十分大きな全ての $\kappa \in \mathbb{N}$ に対して，$\mathsf{para} \in \mathsf{PGen}(1^\kappa)$，$(vk, sk) \in \mathsf{KGen}(\mathsf{para})$，$m \in \{0,1\}^*$，$\sigma \in \mathsf{Sign}(sk, vk, m)$ なら，常に $\mathsf{Vrfy}(vk, m, \sigma) = 1$ を満足するとき，$\mathsf{SIG}$ を**ディジタル署名**とよぶ．

## 4.2 離散対数 (DL) 問題

$\mathbb{G}$ を素数位数 $\ell$ の巡回群とする．元 $X, Y \in \mathbb{G}\backslash\{0\}$ が与えられたとき，$Y = zX$ なる $z \in \mathbb{Z}$ を求める問題を $\mathbb{G}$ 上の離散対数 (Discrete-log (DL)) 問題という．

$\mathsf{Gen}_{\mathrm{GROUP}}$ を次のような確率的アルゴリズムとする．

1. $\mathsf{Gen}_{\mathrm{GROUP}}$ はセキュリティパラメータ $1^\kappa$ を入力として受け取る．
2. $|\ell| = \kappa$ となる素数 $\ell$ と，$\ell$ を位数とする巡回群 $\mathbb{G}$ を選ぶ．
3. $(\mathbb{G}, \ell)$ を出力する．

このアルゴリズムの試行を

$$(\mathbb{G}, \ell) \leftarrow \mathsf{Gen}_{\mathrm{GROUP}}(1^\kappa)$$

と書く．

**定義 9（DL 仮定）** DL 問題を解くアルゴリズム $A$ の成功確率を

$$\mathsf{Adv}^{\mathsf{dl}}_{A,\mathsf{KGen}_{\mathrm{GROUP}}}(\kappa) := \Pr[(\mathbb{G}, \ell) \leftarrow \mathsf{Gen}_{\mathrm{GROUP}}(1^\kappa); (X, Y) \leftarrow (\mathbb{G}\backslash\{0\})^2; \alpha \leftarrow A(X, Y) : Y = \alpha \cdot X]$$

と定義する．ある巡回群生成アルゴリズム $\mathsf{Gen}_{\mathrm{GROUP}}$ が存在して，いかなる多項式時間アルゴリズムの敵 $A$ に対しても $\mathsf{Adv}^{\mathsf{dl}}_{A,\mathsf{Gen}_{\mathrm{GROUP}}}(\kappa) = \mathsf{negl}(\kappa)$ となるとする．このような $\mathsf{Gen}_{\mathrm{GROUP}}$ の存在を仮定することを DL 仮定と言う．

DL 問題の汎用的解法としては Shanks の baby-step-giant-step や Polland の $\rho$ 法や $\lambda$ 法 (kangaloo algorithm) などがある．汎用的解法を使う多項式時間アルゴリズム $A$ では

$$\mathsf{Adv}^{\mathsf{dl}}_{A,\mathsf{Gen}_{\mathrm{GROUP}}}(\kappa) \approx O\Big(\frac{1}{\sqrt{\ell}}\Big) \tag{4}$$

となる．$\ell \approx 2^\kappa$ より $\mathsf{Adv}^{\mathsf{dl}}_{A,\mathsf{Gen}_{\mathrm{GROUP}}}(\kappa) = \mathsf{negl}(\kappa)$ である．

現在のところ EdDSA で使われる推奨のツイスト Edwards 曲線ではこれらの汎用的解法以外知られていない．

## 4.3 Schnorr 署名

Schnorr 署名 [65, 66] は次のようなアルゴリズムの組として定義される．

$\mathbb{G}$ を $\mathsf{Gen}_{\mathrm{GROUP}}$ により生成された DL 仮定の成立する位数 $\ell$ の巡回群とする．システム生成アルゴリズム $\mathsf{PGen}$ を図 2 なるアルゴリズムとする．

そのとき Schnorr 署名 $\mathsf{Schnorr} = (\mathsf{PGen}, \mathsf{SchKgen}, \mathsf{SchSig}, \mathsf{SchVrf})$ は図 3 のようなアルゴリズムの組である．

$\mathsf{para} \leftarrow \mathsf{PGen}(1^\kappa)$.

input: $1^\kappa$

$(\mathbb{G}, \ell) \leftarrow \mathsf{Gen}_{\mathrm{GROUP}}(1^\kappa)$;

$B \leftarrow \mathbb{G}\backslash\{0\}$;

return $\mathsf{para} := (\mathbb{G}, \ell, B)$.

図 2　共通パラメータ生成 PGen

$(vk, sk) \leftarrow \mathsf{SchKgen}(\mathsf{para})$.

input: $\mathsf{para} = (\mathbb{G}, \ell, B)$

$s \leftarrow \mathbb{Z}/\ell\mathbb{Z}$;

$A := sB$;

$vk := A$;

$sk := s$;

return $(vk, sk)$.

$\sigma \leftarrow \mathsf{SchSig}_{sk}(m)$.

input: $(sk, m)$

$r \leftarrow \mathbb{Z}/\ell\mathbb{Z}$;

$R := rB$;

$e := H(R, m)$;

$z := r + es \bmod \ell$;

$\sigma := (R, z)$;

return $\sigma$.

$\tau \leftarrow \mathsf{SchVrf}(vk, m, \sigma)$.

input: $(vk, m, \sigma)$

parse $\sigma = (R, z)$;

if $A, R \notin \mathbb{G}$;

return $\tau := 0$

else

$e := H(R, A, m)$;

if $zB = R + eA$

return $\tau := 1$

else return $\tau := 0$.

図 3　Schnorr 署名

## 4.4　Schnorr 署名から EdDSA 署名への変形

EdDSA 署名は Schnorr 署名の変形版であるのでその対比を表 1 にまとめた．$c = 2$ or 3, $c < n < b$, $|\ell| = b - c$（$\ell$ は素数）．

Schnorr 署名でノンス $r$ を EdDSA のやり方で確定的にした署名を dSchnorr 署名と呼び，さらにハッシュ関数 $H$ の値域を $\mathbb{Z}/\ell\mathbb{Z}$ から $\{0,1\}^{2b}$ に変更したものを lhdSchnorr と呼ぶことにする．チャレンジ $e$ の生成において公開鍵 $A$ をハッシュ関数の入力に含めるのを key-prefixing と言う．上記（変形）Schnorr 署名が key-prefixing を追加したものを接頭に kp を追加する．EdDSA 署名において key-prefixing 無しの EdDSA 署名を nkp-EdDSA 署名とする．

Schnorr 署名から EdDSA 署名までの変形が図 4.4 のようになる．厳密には EdDSA 署名では署名鍵（の一部）$s$ の選び方が Schnorr 署名とわずかに違うが DL 仮定で吸収できるので省略した．

表 1　Schnorr 署名と EdDSA 署名対応表

| 方式 | Schnorr | EdDSA |
|---|---|---|
| 群 | $\mathbb{G}$ $(\#\mathbb{G} = \ell)$ | $E_{(p,a,d)}$ $(\#E_{(p,a,d)} = 2^c\ell)$ |
| ベースポイント | $\langle B\rangle = \mathbb{G}$ | $\langle B\rangle \subset E_{(p,a,d)}$ $(\#B = \ell)$ |
| ハッシュ関数 | $H : \{0,1\}^* \to \mathbb{Z}/\ell\mathbb{Z}$ | $H : \{0,1\}^* \to \{0,1\}^{2b}$ $(\lvert\ell\rvert = b - c)$ |
| $A = sB$ | $s \in \mathbb{Z}/\ell\mathbb{Z}$ | $s \in \{0,1\}^{n+1}$ $(n < b)$ |
| ノンス | $r \leftarrow \mathbb{Z}/\ell\mathbb{Z}$ (probabilistic) | $r = H(s', m)$ (deterministic) |
| チャレンジ | $e = H(R, m)$ | $e = H(R, A, \mathsf{PH}(m))$ (key-prefixing) |
| 群要素チェック | $A, R \in \langle B\rangle$ | $A, R \in E_{(p,a,d)}$ |
| 検証式 | $zB = R + eA$ | $(2^c z)B = 2^c R + (2^c e)A$ |

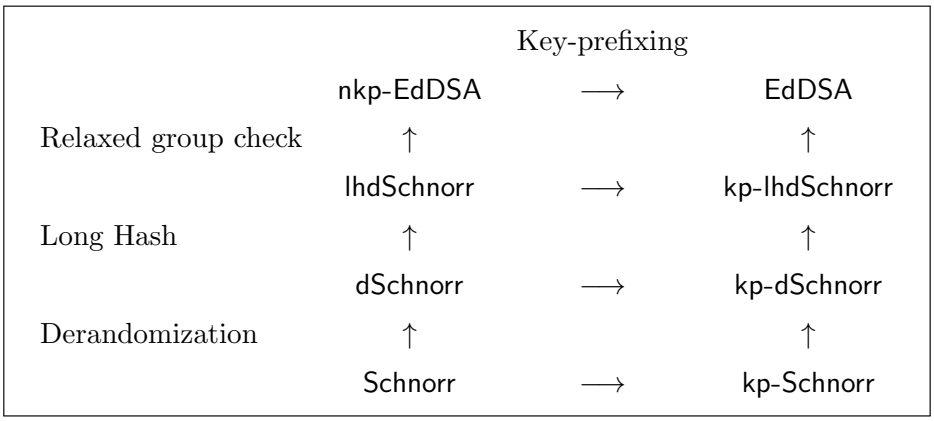


図 4　Schnorr 署名から EdDSA 署名への変換

## 4.5　存在的偽造不可能性 (EUF-CMA 安全性), 存在的強偽造不可能性 (sEUF-CMA 安全性)

ディジタル署名 $\mathsf{SIG} = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ に対する次のような攻撃者 $A$ を考える．$A$ は検証鍵 $vk$ を受け取り，署名オラクル $\mathsf{Sign}_{sk}$ に $q$ 回まで平文を送ることができる．署名オラクル $\mathsf{Sign}_{sk}$ は各質問された平文に正しい署名をそのつど返す．$A$ の署名オラクル $\mathsf{Sign}_{sk}$ へ質問した平文と対応する署名の集合を $L = \{(m_1, \sigma_1), \ldots, (m_q, \sigma_q)\}$ とし，平文の集合を特に $L_m = \{m_1, \ldots, m_q\}$ と書くことにする．$A$ が最終的に $L_m$ に含まれない平文 $m^*$ への正しい署名 $\sigma^*$ （正しい署名とは, $\mathsf{Vrfy}(vk, m^*, \sigma^*) = 1$ なる署名のこと）を出力して来た場合，$A$ の勝ちと定義する．

$A$ が，ディジタル署名 SIG に対する EUF-CMA ゲームに勝つ確率を，$A$ の SIG に対するアド

バンテージと呼び,

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{A,\mathsf{SIG}}(\kappa) := \Pr[\mathsf{Expt}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},A}(\kappa) = 1]$$

によって表す（$\mathsf{Expt}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},A}(\kappa)$ は図 5 と定義）．上記確率は, $\mathsf{KGen}, \mathsf{Sign}, A$（の内部コイン）に依存する．

sEUF-CMA ゲームは，EUF-CMA ゲームとほぼ同じだが，$A$ の勝利条件が，$L$ に含まれない平文と正しい署名の組 $(m^*, \sigma^*)$（すなわち，$(m^*, \sigma^*) \in L$）を出力して来た場合, $A$ の勝ちと定義する． これは EUF-CMA ゲームの勝利条件より厳しくなっている．$A$ が，ディジタル署名 SIG に対する sEUF-CMA ゲームに勝つ確率を，$A$ の SIG に対するアドバンテージと呼び,

$$\mathsf{Adv}^{\mathsf{seuf\text{-}cma}}_{A,\mathsf{SIG}}(\kappa) := \Pr[\mathsf{Expt}^{\mathsf{seuf\text{-}cma}}_{\mathsf{SIG},A}(\kappa) = 1]$$

によって評価される（$\mathsf{Expt}^{\mathsf{seuf\text{-}cma}}_{\mathsf{SIG},A}(\kappa)$ は図 6 と定義）．上記確率は, $\mathsf{KGen}, \mathsf{Sign}, A$（の内部コイン）に依存する．

$\mathsf{Expt}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},A}(\kappa)$:

para $\leftarrow \mathsf{PGen}(1^\kappa)$
$(vk, sk) \leftarrow \mathsf{KGen}(\mathsf{para})$
$(m^*, \sigma^*) \leftarrow A^{\mathsf{Sign}_{sk}(vk,\cdot)}(vk)$
If $m^* \notin L_m$,
  return $\mathsf{Vrfy}(vk, m^*, \sigma^*)$
else return 0.

図 5　EUF-CMA 試行

$\mathsf{Expt}^{\mathsf{seuf\text{-}cma}}_{\mathsf{SIG},A}(\kappa)$:

para $\leftarrow \mathsf{PGen}(1^\kappa)$
$(vk, sk) \leftarrow \mathsf{KGen}(\mathsf{para})$
$(m^*, \sigma^*) \leftarrow A^{\mathsf{Sign}_{sk}(vk,\cdot)}(vk)$
If $(m^*, \sigma^*) \notin L$,
  return $\mathsf{Vrfy}(vk, m^*, \sigma^*)$
else return 0.

図 6　sEUF-CMA 試行

**定義 10（EUF-CMA 安全性，sEUF-CMA 安全性）**　署名オラクルに $q$ 回までアクセスできる任意の $t$-時間アルゴリズム（族）の攻撃者 $A$ に対して, 十分大きな全ての $\kappa$ で, $\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{A,\mathsf{SIG}}(\kappa) \leq \epsilon$ が成立するとき，ディジタル署名 SIG は $(t, q, \epsilon)$-**EUF-CMA 安全**という．同様に十分大きな全ての $\kappa$ で, $\mathsf{Adv}^{\mathsf{seuf\text{-}cma}}_{A,\mathsf{SIG}}(\kappa) \leq \epsilon$ が成立するとき，ディジタル署名 SIG は, $(t, q, \epsilon)$-**sEUF-CMA 安全**という．特に, $t, q = O(\mathrm{poly}(\kappa))$ で，$\epsilon(\kappa) = \mathsf{negl}(\kappa)$ であるとき，SIG はそれぞれ EUF-CMA **安全**, **sEUF-CMA 安全**という．

## 4.6　複数署名者版の存在的（強）偽造不可能性（m(s)EUF-CMA 安全性）

Galbraith ら [38] や Menezes と Smart [48] によって複数の署名版での（強）存在的偽造不可能性が定義されているので紹介する．

攻撃者 $A$ は独立に作られた $n$ 個の検証鍵 $vk_1, \ldots, vk_n$ を受け取り, 各 $vk_i$ に対応する $n$ 個の署名オラクルに合計最大 $q$ 回まで質問できる．便宜上，$A$ の質問を $(vk, m)$ とすることで $n$ 個の署

名オラクルを一つの署名オラクル $\mathsf{Sign}(\cdot,\cdot)$ とする．署名オラクルは $vk$ が $A$ に与えられた $n$ 個の検証鍵のどれかに一致するのであれば対応する秘密鍵 $sk$ を用い署名 $\sigma \leftarrow \mathsf{Sign}_{sk}(vk, m)$ を $A$ に返す．$A$ の署名オラクル $\mathsf{Sign}$ との記録を $L = \{(vk_{t_1} m_1, \sigma_1), \ldots, (vk_{t_q}, m_q, \sigma_q)\}$ とし，特に $L_m = \{(vk_{t_1}, m_1), \ldots, (vk_{t_q}, m_q)\}$ と書くことにする．複数署名者版 EUF-CMA ゲームでは，$A$ が $(vk^*, m^*) \notin L_m$（ただし $vk^*$ は与えられた $n$ 個の検証鍵の一つ）に対する正しい署名 $\sigma^*$ を出力した場合 $A$ の勝利とし，複数署名者版 sEUF-CMA ゲームでは，$A$ が $(vk^*, m^*, \sigma^*) \notin L$（ただし $vk^*$ は与えられた $n$ 個の検証鍵の一つ）で $\sigma^*$ が正しい署名のとき $A$ の勝利とする．このゲームの $A$ の試行はそれぞれ図 7 と図 8 の通りである．

$\mathsf{Expt}^{\mathsf{meuf\text{-}cma}}_{\mathsf{SIG},A}(\kappa)$:
- $\mathsf{para} \leftarrow \mathsf{PGen}(1^\kappa)$
- $(vk_1, sk_1), \ldots, (vk_n, sk_n) \leftarrow \mathsf{KGen}(\mathsf{para})$
- $(vk^*, m^*, \sigma^*) \leftarrow A^{\mathsf{Sign}(\cdot,\cdot)}(vk_1, \ldots, vk_n)$
- If $\exists i$ s.t. $vk^* = vk_i$ and $(vk^*, m^*) \notin L_m$,
  - return $\mathsf{Vrfy}(vk, m^*, \sigma^*)$
- else return 0.

図 7　mEUF-CMA 試行

$\mathsf{Expt}^{\mathsf{mseuf\text{-}cma}}_{\mathsf{SIG},A}(\kappa)$:
- $\mathsf{para} \leftarrow \mathsf{PGen}(1^\kappa)$
- $(vk_1, sk_1), \ldots, (vk_n, sk_n) \leftarrow \mathsf{KGen}(\mathsf{para})$
- $(m^*, \sigma^*) \leftarrow A^{\mathsf{Sign}(\cdot,\cdot)}(vk_1, \ldots, vk_n)$
- If $\exists i$ s.t. $vk^* = vk_i$ and $(vk^*, m^*, \sigma^*) \notin L$,
  - return $\mathsf{Vrfy}(vk, m^*, \sigma^*)$
- else return 0.

図 8　msEUF-CMA 試行

攻撃者 $A$ のアドバンテージも同様に定義できる．

$$\mathsf{Adv}^{\mathsf{meuf\text{-}cma}}_{A,\mathsf{SIG}}(\kappa) := \Pr[\mathsf{Expt}^{\mathsf{meuf\text{-}cma}}_{\mathsf{SIG},A}(\kappa) = 1],$$
$$\mathsf{Adv}^{\mathsf{mseuf\text{-}cma}}_{A,\mathsf{SIG}}(\kappa) := \Pr[\mathsf{Expt}^{\mathsf{mseuf\text{-}cma}}_{\mathsf{SIG},A}(\kappa) = 1]$$

よって EUF-CMA 安全性や sEUF-CMA 安全性の時と同様に $(t, q, \epsilon)$-**mEUF-CMA 安全**や $(t, q, \epsilon)$-**msEUF-CMA 安全**, さらに **mEUF-CMA 安全**や **msEUF-CMA 安全**が定義できる．

**定理 11（[38]）**　任意の署名方式 $\mathsf{SIG}$ に対して，多項式時間の複数署名者版 (s)EUF-CMA 攻撃者 $A$ が存在したとすると下記の関係を満たす多項式時間の（単一署名者版）(s)EUF-CMA 攻撃者 $A'$ が存在し

$$\mathsf{Adv}^{\mathsf{m(s)euf\text{-}cma}}_{A,\mathsf{SIG}}(\kappa) \leq n \cdot \mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A',\mathsf{SIG}}(\kappa)$$

を満たす．

すなわちこれは $n = \mathrm{poly}(\kappa)$ である限り $\mathsf{SIG}$ が (s)EUF-CMA であれば必ず m(s)EUF-CMA であることを示している．一方で安全性の具体的値をみると署名者の数だけ安全性が劣化する（可能性がある）ことを示している．

Bernstein は，kp-Schnorr 署名の m(s)EUF-CMA 安全性が単一署名者版の Schnorr 署名の (s)EUF-CMA 安全性に帰着する以下の定理を示した [9].

**定理 12（[9]）** Schnorr 署名が $(t, q, \epsilon)$-(s)EUF-CMA 安全とすると kp-Schnorr 署名は $(t', q, \epsilon')$-m(s)EUF-CMA 安全であり，

$$t' \approx t \quad (\text{ただし } t' \leq t) \qquad \text{かつ} \qquad \epsilon' \leq \epsilon$$

である [*6].

上記の Bernstein の定理を使うと m(s)EUF-CMA ゲームの署名者数を特に $n = 1$ にすれば，kp-Schnorr 署名の (s)EUF-CMA 安全性は Schnorr 署名の (s)EUF-CMA 安全性に劣化せず帰着する．すなわち

$$\mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A,\mathsf{kp\text{-}Schnorr}}(\kappa) \leq \mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A',\mathsf{Schnorr}}(\kappa),$$
$$\mathsf{Adv}^{\mathsf{m(s)euf\text{-}cma}}_{A,\mathsf{kp\text{-}Schnorr}}(\kappa) \leq \mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A',\mathsf{Schnorr}}(\kappa)$$

が成り立つ．ただし $\mathsf{Adv}^{\mathsf{m(s)euf\text{-}cma}}_{A,\mathsf{kp\text{-}Schnorr}}(\kappa) \approx \mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A,\mathsf{kp\text{-}Schnorr}}(\kappa)$ であるかはわからない．例えば

$$\mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A,\mathsf{kp\text{-}Schnorr}}(\kappa) << \mathsf{Adv}^{\mathsf{m(s)euf\text{-}cma}}_{A',\mathsf{kp\text{-}Schnorr}}(\kappa)$$

のような劣化が複数署名者では起きているかもしれない．

同じことが nkp-EdDSA 署名と EdDSA 署名に関しても成り立ち，

**命題 13** nkp-EdDSA 署名が $(t, q, \epsilon)$-(s)EUF-CMA 安全とすると EdDSA 署名は $(t', q, \epsilon')$-m(s)EUF-CMA 安全であり，

$$t' \approx t \quad (\text{ただし } t' \leq t) \qquad \text{かつ} \qquad \epsilon' \leq \epsilon$$

である．

よって同様に

$$\mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A,\mathsf{EdDSA}}(\kappa) \leq \mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A',\mathsf{nkp\text{-}EdDSA}}(\kappa),$$
$$\mathsf{Adv}^{\mathsf{m(s)euf\text{-}cma}}_{A,\mathsf{EdDSA}}(\kappa) \leq \mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A',\mathsf{nkp\text{-}EdDSA}}(\kappa)$$

が成り立つ．ただし $\mathsf{Adv}^{\mathsf{m(s)euf\text{-}cma}}_{A,\mathsf{EdDSA}}(\kappa) \approx \mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A,\mathsf{EdDSA}}(\kappa)$ であるかはわからない．

4.8 章の考察により，ランダムオラクルモデルにおいては EdDSA 署名が EUF-CMA 安全性を満たせば sEUF-CMA 安全性も満たしその安全性は劣化しない．

ディジタル署名の標準的な安全性のクラスは (s)EUF-CMA 安全性であるが，例えば Chalkias ら [26] は（強) 束縛性 ((strongly) binding) という安全性クラスを定義し Ed25519-EdDSA 署名がこれらの性質を満たすか調査している．

[*6] [9] は EUF-CMA と mEUF-CMA の関係のみを述べていると読める．しかし sEUF-CMA と msEUF-CMA の関係にも拡張できると考えられるので一般化された定理を彼らのものとして記述した．

## 4.7 ハッシュ関数の必要条件

この章では EdDSA 署名や Schnorr 署名におけるハッシュ関数の満たすべき必要条件について考察する．

**定義 14（衝突困難性 (Collision-Resistance)）** ハッシュ関数 $H : \{0,1\}^* \to \{0,1\}^\kappa$ を攻撃する敵 $A$ が衝突を成功させる確率を，

$$\mathsf{Adv}^{\mathsf{cr}}_{A,H}(\kappa) := \Pr[(m, m') \leftarrow A(H) : (H(m) = H(m')) \wedge (m \neq m')]$$

と定義する．任意の $t$-時間の敵 $A$ に対しても，$\mathsf{Adv}^{\mathsf{cr}}_{A,H}(\kappa) \leq \epsilon$ を満足するのであれば，$\mathcal{H}$ は，$(t, \epsilon)$-**衝突困難 (CR)** であると言う [*7]．$t(\kappa) = \mathrm{poly}(\kappa)$, $\epsilon(\kappa) = \mathsf{negl}(\kappa)$ ならば，$H$ は**衝突困難 (CR)** であると言う．

衝突困難性 (collision-resitance) はハッシュ関数 $H : \{0,1\}^* \to \{0,1\}^\kappa$ の基本的な要求条件で，実際 HashEdDSA では PH で使われるハッシュ関数に衝突が見つかるようだと署名の偽造が可能である．次の命題は明らかである．

**命題 15** $A$ を $t$-時間でハッシュ関数 PH の衝突を探すアルゴリズムとすると，HashEdDSA を EUF-CMA ゲームで攻撃するある $t'$-時間 $(t \approx t)$ アルゴリズム $A'$ を構成できその偽造確率は

$$\mathsf{Adv}^{\mathsf{cr}}_{A,\mathsf{PH}}(\kappa) \leq \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{A',\mathsf{HashEdDSA}}(\kappa) \tag{5}$$

となる．

一方，PureEdDSA 署名でハッシュ関数 $H$ の衝突が見つかったとしても偽造が可能であるかは明らかでない．Bernstein はこの事実をもって PureEdDSA 署名がハッシュの衝突により安全性が毀損されないという意味で collision resistant と述べている．

衝突困難性 (CR) より弱い安全性をもつハッシュ関数の性質として，Neven ら [53] により導入されたランダムプレフィクス原像計算困難性 (random-prefix preimage resistance (RPP)) [*8] とランダムプレフィクス第 2 原像計算困難性 (random-prefix second preimage resistance (RPSP)) を定義する．

**定義 16（[53]）** ハッシュ関数 $H : \{0,1\}^* \to \{0,1\}^\kappa$ を攻撃する敵 $A$ の次のような確率を定義する．

$\mathsf{Adv}^{\mathsf{rpp}}_{A,H}(\kappa) := \Pr[(y, st) \leftarrow A_1(H); R \leftarrow D; m \leftarrow A_2(R, st) : H(R, m) = y]$
$\mathsf{Adv}^{\mathsf{rpsp}}_{A,H}(\kappa) := \Pr[(m, st) \leftarrow A_1(H); R \leftarrow D; m' \leftarrow A_2(R, st) : (H(R, m) = H(R, m')) \wedge (m \neq m')]$

[*7] 厳密にはハッシュ関数族で定義すべきだが省略している．

[*8] RPP 安全性は Kelsey と Kohno が [45] で導入した chosen-target forced prefix (CTFP) preimage resistance と同じである．

任意の $t$-時間の敵 $A$ に対しても，$\mathsf{Adv}^{\mathsf{rpp}}_{A,\mathcal{H}}(\kappa) \leq \epsilon$ を満足するのであれば，$H$ は（$D$ に関する）$(t,\epsilon)$-**ランダムプレフィクス原像計算困難 (RPP)** と呼ぶ．同様に，任意の $t$-時間の敵 $A$ に対しても，$\mathsf{Adv}^{\mathsf{rpsp}}_{A,\mathcal{H}}(\kappa) \leq \epsilon$ を満足するのであれば，$H$ は（$D$ に関する）$(t,\epsilon)$-**ランダムプレフィクス第２原像計算困難 (RPSP)** と呼ぶ．$t(\kappa) = \mathrm{poly}(\kappa)$, $\epsilon(\kappa) = \mathsf{negl}(\kappa)$ ならば，それぞれ $H$ は（$D$ に関する）**ランダムプレフィクス原像計算困難 (RPP)** と**ランダムプレフィクス第２原像計算困難 (RPSP)** と呼ぶ．

次のようなハッシュ関数を考える．$H_1 : D \to \{0,1\}^\kappa$ を $D$ の元を $\kappa$ ビットにエンコードする関数で逆変換可能とする．一方 $H_2 : \{0,1\}^* \to \{0,1\}^\kappa$ を CR-ハッシュ関数とする．このとき新しいハッシュ関数 $H$ を次のように定義する．

$$H(R,m) := H_1(R) \oplus H_2(m) \tag{6}$$

ここで $\oplus$ は bit-wise xor を表す．

このように作られた $H$ は RPP かつ RPSP である．しかし CR ではない．この関数は [21] で指摘されているものと本質的に同じである．通常のハッシュ関数はこのような性質を持っていないが RPP や RPSP が CR より真に弱い性質であることを示している．

**命題 17**　$H_{vk} := H(\cdot, vk, \cdot)$ とする．$A$ を $t$-時間でハッシュ関数 $H_{vk}$ の $D = \langle B \rangle$ に関する RPP 衝突，または RPSP 衝突を探すアルゴリズムとする．すると PureEdDSA を EUF-CMA ゲームで攻撃するある $t'$-時間 $(t \approx t)$ アルゴリズム $A'$ を構成できその偽造確率は

$$\mathsf{Adv}^{\mathsf{rpp}}_{A,H_{vk}}(\kappa) \leq \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{A',\mathsf{PureEdDSA}}(\kappa) \tag{7}$$

$$\mathsf{Adv}^{\mathsf{rpsp}}_{A,H_{vk}}(\kappa) \leq \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{A',\mathsf{PureEdDSA}}(\kappa) \tag{8}$$

となる．

Neven らは RPP かつ RPSP ハッシュ関数の衝突を見つけるには $O(2^\kappa)$ 回の試行が必要と仮定しハッシュ関数出力長を従来の CR ハッシュ関数の出力長の半分にすることを提案している [53]. しかし Brown が指摘したように式 (6) は RPP かつ RPSP であるが $H_2$ が CR ハッシュ関数であるため $O(2^{\kappa/2})$ 回の試行で $m$ を発見することが期待できる [21].

よって PRR，RPSP は CR より真に弱い安全性であるが，セキュリティパラメータを小さくできるわけではないことがわかる．

## 4.8　ランダムオラクルモデルでの安全性評価

この章から EdDSA 署名の安全性の十分条件について考察していく．

ランダムオラクルモデルとはハッシュ関数をランダム関数とみなし，その出力結果を得たいときはオラクルとしてその関数にアクセスするモデルである．Schnorr 署名は DL 仮定のもとランダ

ムオラクルモデルで EUF-CMA 安全性を満たすことが Pointcheval と Stern により証明されている [58]．しかしその証明では DL 問題の困難性への帰着効率が明白ではない．ここでは Bellare と Neven の generalized forking lemma [7] をもとにしたつぎの Neven らの結果 [53] を引用して使用する．

**定理 18（[53]）** $\mathsf{KGen}_{\mathrm{GROUP}}$ が $(t_{\mathsf{dl}}, \epsilon_{\mathsf{dl}})$-DL 安全とする．そのとき $H$ をランダムオラクルとみなしたランダムオラクルモデルで Schnorr 署名は $(t_{\mathsf{sch}}, q_H, q_s, \epsilon_{\mathsf{sch}})$-EUF-CMA 安全である．

$$t_{\mathsf{sch}} \leq \frac{1}{2} t_{\mathsf{dl}} - q_s T - O(q_H + q_s + 1)$$
$$\epsilon_{\mathsf{sch}} \leq \sqrt{(q_H + q_s + 1) \cdot \epsilon_{\mathsf{dl}}} + \frac{(q_s + 1)(q_H + q_s + 1)}{\ell}.$$

ここで $q_H$ は攻撃者がランダムオラクルへアクセスできる上限数であり，$q_s$ は攻撃者が署名オラクルへアクセスできる上限数である．さらに $T$ は $\mathbb{G}$ での $\ell$ 倍算の計算時間を表す．

EUF-CMA 安全性を sEUF-CMA 安全に置き換えることを考える．sEUF-CMA ゲームでは攻撃者の勝利条件が緩和され，今までの偽造条件に加えてすでに署名オラクルに質問した平文 $m$ であっても署名オラクルから受け取った署名 $\sigma = (R, z)$ 以外の正しい署名 $\sigma^* = (R^*, z^*)$ $((R^*, z^*) \neq (R, z))$ を作ることができれば偽造は成功したことになる．このとき Schnorr 署名の特徴により $R = R^*$ の場合 $z = z^*$ しかない．よって既に署名オラクルに質問した平文 $m$ に対してゲームに勝利する偽造署名を出すためには $R \neq R^*$ が必要になるが，Schnorr 署名におけるランダムオラクル $H$ への質問は $(R^*, m)$ となり $(R, m)$ とは異なる質問となるためランダムオラクルモデルにおける EUF-CMA ゲームでの folking lemma のイベントに既に捕らえられている．よって次が成り立つ．

**系 19** Schnorr 署名がランダムオラクルモデルで $(t_{\mathsf{sch}}, q_H, q_s, \epsilon_{\mathsf{sch}})$-EUF-CMA 安全であるとする．そのとき Schnorr 署名はランダムオラクルモデルで $(t_{\mathsf{sch}}, q_H, q_s, \epsilon_{\mathsf{sch}})$-sEUF-CMA 安全である．

Bernstein の結果（定理 12）により，Schnorr 署名を kp-Schnorr 署名に置き換えても安全性は劣化しない．次に kp-Schnorr 署名を kp-dSchnorr 署名に置き換えてもノンス $r$ をつくるハッシュ関数 $H$ がランダム関数である限り評価にほとんど変化は生じない．$H(s', \cdot)$ を疑似ランダム関数と見る場合は $\epsilon_{\mathsf{prf}}$ の因子が帰着の上限に加算される（$\epsilon_{\mathsf{prf}}$ は疑似ランダム関数に多項式回アクセスした場合のランダム関数との識別確率）．kp-dSchnorr 署名を kp-lhdSchnorr 署名に置き換えた場合，$r \bmod \ell$ と $e \bmod \ell$ は $\mathbb{Z}/\ell\mathbb{Z}$ 上の一様分布との統計的距離の差が $2^{-b}$ 以内におさえられるため帰着の式にほぼ影響を与えない．最後に群要素チェックを $A, R \in E_{(p,a,d)}$ に緩める場合だが，$A$ は正規の署名者の公開鍵のため $A \in \langle B \rangle$ である．よって攻撃者が新たにできるのは $R' = R + C$ $(2^c C = 0, R \in \langle B \rangle)$ のような $R'$ で偽造署名をつくることであるが，$R' \neq R$ であるため EUF-CMA 安全性を sEUF-CMA 安全性に置き換えたときと同様でランダムオラクルモデル

での証明に問題を生じさせない．よって以下の命題 20 が成り立つ．

**命題 20**　Schnorr 署名がランダムオラクルモデルで $(t_{\mathsf{sch}}, q_H, q_s, \epsilon_{\mathsf{sch}})$-EUF-CMA 安全であるとする．そのとき PureEdDSA 署名はランダムオラクルモデルで $(t, q_H, q_s, \epsilon)$-sEUF-CMA 安全である．

$$t \approx t_{\mathsf{sch}} \quad （ただし\ t \leq t_{\mathsf{sch}}） \qquad かつ \qquad \epsilon \leq \epsilon_{\mathsf{sch}}$$

4.6 章の結果からさらに

**系 21**　Schnorr 署名がランダムオラクルモデルで $(t_{\mathsf{sch}}, q_H, q_s, \epsilon_{\mathsf{sch}})$-EUF-CMA 安全であるとする．そのとき PureEdDSA 署名はランダムオラクルモデルで $(t, q_H, q_s, \epsilon)$-msEUF-CMA（複数署名者版 sEUF-CMA）安全である．

$$t \approx t_{\mathsf{sch}} \quad （ただし\ t \leq t_{\mathsf{sch}}） \qquad かつ \qquad \epsilon \leq \epsilon_{\mathsf{sch}}$$

HashEdDSA 署名の場合 PH 関数の衝突確率分安全性が劣化すると考えられる．

$$\mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A,\mathsf{HashEdDSA}}(\kappa) \leq \mathsf{Adv}^{\mathsf{(s)euf\text{-}cma}}_{A',\mathsf{PureEdDSA}}(\kappa) + O\Big(\frac{q_H}{\sqrt{\ell}}\Big)$$

#### 4.8.1　推奨パラメータについて

推奨パラメータ Ed25519 でハッシュ関数に SHA512 が使われているが，SHA512 は Merkle-Damgård 構造 [49, 31] を使っているため length-extension 攻撃によりランダム関数と容易に識別可能である．よって Ed25519 パラメータではランダムオラクルでの安全性評価ができない．

一方推奨パラメータ Ed448 ではハッシュ関数に SHAKE256 が使われており，少なくともランダム関数と識別可能な方法は知られていない．よってランダムオラクルモデルでの結果はある程度の安全性への保証を与える．ただしランダムオラクルモデルでは偽造確率を $O(\sqrt{\epsilon_{\mathsf{dl}}})$ でしか抑えられない．Ed448 では 224 ビット安全性を目指しているはずだが，定理 18 は高々 112 ビット程度の安全性を保証しているだけとなる．

### 4.9　関連鍵攻撃安全性

公開鍵 $A(= sB)$ として平文 $m$ の Schnorr 署名を $(R, z)$ とする．すると公開鍵 $A' = A + \delta B$ の平文 $m$ の Schnorr 署名が $(R', z') := (R, z + H(R, m) \cdot \delta)$ のように作れてしまう．このような攻撃を関連鍵攻撃と言う．

森田らは key-prefixing をすることでこれらの攻撃をある程度防ぐことができることを示した [50]．より具体的に述べると，kp-Schnorr 署名の公開鍵 $A$ から多項式変換 $A' = f(A)$（すなわち $f$ が多項式）された公開鍵 $f(A)$ に対して選択平文攻撃ができる場合，ターゲットの公開鍵 $A$ の署名を偽造することは，ランダムオラクルモデルで強離散対数 (strong DL) 問題仮定の解読困難性

のもと困難であり，その偽造確率は $O(\sqrt{\epsilon_{d\text{-sdl}}})$ でほぼ抑えられることを示した．ここで $d$-strong DL 問題とは $(B, xB, (x^2)B, \ldots, (x^d)B)$ が与えられたとき $x \in \mathbb{Z}/\ell\mathbb{Z}$ を求める問題である．$d$ は多項式 $f$ の次数に対応しており $f$ が一次式であれば安全性の仮定は DL 仮定が対応する．

森田らの攻撃モデルは厳密にはここでいう関連鍵攻撃と少し異なっている（$A$ と $f(A)$ の関係が反対になっている）．しかし，EdDSA 署名をランダムオラクルモデルで評価できるときは，同様の結果を期待して良いと考えられる．

### 4.10 Generic Group model での安全性評価

Schnorr 署名を generic group model [68] で評価した Neven らの結果を紹介する．このモデルでは，暗号方式や離散対数問題をその定義される（有限巡回) 群の一般的性質のみを使って解読を試みたときの評価を得ることができる．

**定理 22（[53]）** $H : \{0,1\}^* \to \mathbb{Z}/\ell\mathbb{Z}$ を RPP-安全かつ，RPSP-安全とする．すると Schnorr 署名は generic group model で EUF-CMA 安全であり Schnorr 署名を攻撃する多項式時間制限の敵 $A$ の攻撃成功確率は，ある多項式時間のハッシュ関数の敵 $B_1$, $B_2$ の攻撃成功確率により次のように抑えられる．

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{A,\mathsf{Schnorr}}(\kappa) \le \frac{q_G}{\delta} \cdot \mathsf{Adv}^{\mathsf{rpp}}_{B_1,H}(\kappa) + \frac{q_s+2}{\delta} \cdot \mathsf{Adv}^{\mathsf{rpsp}}_{B_2,H}(\kappa) + O\Big(\frac{(q_s+q_G)^2}{\ell}\Big)$$

EdDSA 署名のエンコーディングを使うと $\delta = \frac{\ell}{2^{b-c}} \approx 1$ となる．

定理 22 を EdDSA 署名に適用しようと考えた場合，もっとも問題なのはノンス $r$ の部分をハッシュ関数の出力に置き換えられるかという部分である．ハッシュ関数の性質を RPP-安全かつ RPSP-安全としただけでは同様の結果を得ることは難しいと考える．一方，ノンス $r$ を疑似ランダム関数の出力と見なすのであれば PureEdDSA 署名では同様の結果が得られると考えられる．ただし既に述べているように Ed25519 ではハッシュ関数に SHA512 が使われており疑似ランダム関数と見なすことはできない．

Ed448 の場合，ハッシュ関数は SHAKE256 であるのでノンスを疑似ランダム関数の出力とみなすのであれば定理 22 の結果を利用できるものと思われる．そのとき（疑似ランダム関数が十分性能がよければ）PureEdDSA のビット安全性は $\frac{\kappa}{2} = 224$ となる．

## 5 ECDSA 署名

EdDSA 署名と比較するために ECDSA 署名について簡単に解説する．ECDSA 署名については SEC1 の記述に基づくが以下のような抽象的な記述に留める．

$p$ を $p > 3$ なる素数．素体 $\mathbb{F}_p$ 上の楕円曲線の Weierstrass 方程式を

$$E : y^2 = x^3 + ax + b \pmod{p} \tag{9}$$

とする．ここで $a, b \in \mathbb{F}_p$ を $4a^3 + 27b^2 \neq 0$ である．式 (9) 上の点 $(x_1, y_1), (x_2, y_2) \in \mathbb{F}_p \times \mathbb{F}_p$ の加法 $(x_3, y_3) := (x_1, y_1) + (x_2, y_2)$ を無限遠点 $O$ という特別な点を一点加えて次のように定義する．$x_1 \neq x_2$ のときは

$$x_3 := \Big(\frac{y_2 - y_1}{x_2 - x_1}\Big)^2 - x_1 - x_2 \pmod p$$
$$y_3 := \frac{y_2 - y_1}{x_2 - x_1}(x_1 - x_3) - y_1 \pmod p,$$

$(x_1, y_1) = (x_2, y_2)$ のときは

$$x_3 := \Big(\frac{3x_1^2 + a}{2y_1}\Big)^2 - 2x_1 \pmod p$$
$$y_3 := \frac{3x_1^2 + a}{2y_1}(x_1 - x_3) - y_1 \pmod p$$

と定義し，さらに $x_1 = x_2$ かつ $y_1 = -y_2$（$y_1 = y_2 = 0$ も含むことに注意）のとき $(x_3, y_3) := O$. 最後に $O := O + O$ と定義すると式 (9) 上の点と無限遠点 $O$ の集合は加法群になる．この群の元 $B$ の位数が素数 $\ell$ であるとし $B$ の生成する群を $\mathbb{G} := \langle B \rangle$ で表す．この $\mathsf{para} = (\mathbb{G}, \ell, B)$ を生成するアルゴリズムを $\mathsf{PGen}$（図 2）とすると，ECDSA 署名 $\mathsf{ECDSA} = (\mathsf{PGen}, \mathsf{ECKgen}, \mathsf{ECSig}, \mathsf{ECVrf})$ は図 9 のような署名方式である．

$(vk, sk) \leftarrow \mathsf{ECKgen}(\mathsf{para})$.

input: $\mathsf{para} = (\mathbb{G}, \ell, B)$
$s \leftarrow \mathbb{Z}/\ell\mathbb{Z}$;
$A := sB$;
$vk := A$;
$sk := s$;

return $(vk, sk)$.

$\sigma \leftarrow \mathsf{ECSig}_{sk}(m)$.

input: $(sk, m)$
$r \leftarrow (\mathbb{Z}/\ell\mathbb{Z})^\times$ (*) ;
$R = (x_r, y_r) := rB$;
$t := x_r \bmod \ell$;
if $t = 0$;
go back to (*)
$e := H(m)$;
$z := r^{-1}(e + t \cdot s) \bmod \ell$;
$\sigma := (t, z)$;

return $\sigma$.

$\tau \leftarrow \mathsf{ECVrf}(vk, m, \sigma)$.

input: $(vk, m, \sigma)$
parse $\sigma = (t, z)$;
if $t, z \notin \mathbb{Z}/\ell\mathbb{Z}$
return $\tau := 0$
$e := H(m)$;
$u_1 = e \cdot z^{-1} \bmod \ell$;
$u_2 = t \cdot z^{-1} \bmod \ell$;
$R = (x_r, y_r)$
$:= u_1 B + u_2 A$;
$t' := x_r \bmod \ell$;
if $t' \neq t$
return $\tau := 0$
return $\tau := 1$

図 9　ECDSA 署名

### 5.1 ECDSA 署名の安全性

ECDSA 署名については Brown による generic group model での安全性評価が存在する [17, 18, 19, 20]．ただし Stern ら [69] が指摘したように，ECDSA 署名は汎用的な群表現以上の性質を使い署名生成や検証を行っているため Brown の結果と齟齬をきたしている部分がある．例えば ECDSA 署名では $(t, z)$ が平文 $m$ の署名であるならば,

$$\frac{e}{-z} \cdot B + \frac{t}{-z} \cdot A = -\Big(\frac{e}{z} \cdot B + \frac{t}{z} \cdot A\Big) = -R = (x_r, -y_r)$$

となるので，$(t, -z)$ も同じ平文 $m$ に対する正しい署名である．これは ECDSA 署名が generic group model でハッシュ関数が CR-安全であれば sEUF-CMA 安全であるという Brown の主張に明らかに反している．

Fersch ら [36] は GenDSA という DSA 署名および ECDSA 署名を抽象化して捉えた署名方式の安全性証明を示しているが ECDSA 署名そのものの安全性を述べているわけではない．

この他 Vaudenay による ECDSA 署名についての安全性評価 [73] や，CRYPTREC [30] の外部評価レポート [67, 56] にも安全性評価が存在する．

## 6 EdDSA 署名と ECDSA 署名の計算量比較

この章では EdDSA 署名と ECDSA 署名の計算量について比較する．

### 6.1 署名生成と検証の計算量比較

ツイスト Edwards 曲線 $E_{(p,a,d)}$ 上の点 $B$ の $|\ell|$ ビット長スカラー倍演算の計算量を $\mathbf{Exp}_\ell(E_{(p,a,d)})$ で表し，同様に Weierstrass 標準形楕円曲線 $E : y^2 = x^3 + ax + b$ 上の点 $B$ の $|\ell|$ ビット長スカラー倍演算の計算量を $\mathbf{Exp}_\ell(E)$ で表すことにする．また，$\mathbf{M}_\ell$ で有限体 $\mathbb{F}_\ell$ 上の元 $\alpha, \beta$ の掛け算 $\alpha\beta$ の計算量を表し．$\mathbf{I}_\ell$ で有限体 $\mathbb{F}_\ell$ 上の元 $\alpha$ から逆算 $\alpha^{-1}$ を計算する計算量を表すとする．さらに $T_H$ でハッシュ関数の計算量，$T_{\mathsf{prf}}$ で ECDSA においてノンス $r$ を生成する計算量とする．

このとき EdDSA 署名と ECDSA 署名の 1 回の署名生成と署名検証の計算量は以下のようになる．

表 2　単独署名生成と署名検証の計算量比較

| | EdDSA 署名 | ECDSA 署名 |
|---|---|---|
| 署名生成 | $1\mathbf{Exp}_\ell(E_{(p,a,d)}) + 1\mathbf{M}_\ell + 2T_H$ | $1\mathbf{Exp}_\ell(E) + 1\mathbf{I}_\ell + 1\mathbf{M}_\ell + T_{\mathsf{prf}} + T_H$ |
| 署名検証 | $2\mathbf{Exp}_\ell(E_{(p,a,d)}) + T_H$ | $2\mathbf{Exp}_\ell(E) + 1\mathbf{I}_\ell + 2\mathbf{M}_\ell + T_H$ |

## 6.2 バッチ署名検証

複数の署名を同時に検証することで独立に全ての署名を検証するより少ない計算量で署名を検証する技術をバッチ署名検証という [51, 71, 6, 42, 24, 28, 40, 43].

単独署名者の $n$ 個の平文と署名のペア $(m_i, (R_i, z_i))$ に対して，ある（小さい）集合 $S \subset \mathbb{Z}/\ell\mathbb{Z}$ から $\alpha_i$ をランダムに選び

$$\sum_{i=1}^{n} \alpha_i R_i = \Big(\sum_{i=1}^{n} (\alpha_i z_i \bmod \ell)\Big) B + \sum_{i=1}^{n} (-\alpha_i e_i \bmod \ell) A \tag{10}$$

が成立することを確認することで，$1/\#S$ の検証エラーを許すことで EdDSA 署名を独立に検証するより高速に署名検証を終えることができる [42]．さらに $\sum_{i=1}^{k} \gamma_i P_i$ を計算する場合，$2^k$ 個の点 $\sum_{i=1}^{k} b_i P_i$ $(b_i \in \{0, 1\})$ を予め計算しておくことで $|\ell| \geq \max_i\{\gamma_i\}$ として $|\ell|(\mathbf{M}_P + (1 - 1/2^k)\mathbf{S}_P)$ の計算量で $\sum_{i=1}^{k} \gamma_i P_i$ を計算できる（ここで，$\mathbf{M}_P$ は加法群 $\langle P \rangle$ 上の加法，$\mathbf{S}_P$ は 2 倍算の計算量）．よって右辺の場合，$\Big(\sum_i (\alpha_i z_i \bmod \ell)\Big) B$ と $\sum_i (-\alpha_i e_i \bmod \ell) A$ を独立に計算することなく，$1\mathbf{M}_P + |\ell|(\mathbf{M}_P + 3/4\mathbf{S}_P)$ の計算量で求めることができる．左辺は $k = n$ が大きいのでこのやり方はあまり効率的で無い．Berstein らは $k$ が大きい場合 Bos-Coster 法 [16, 33] を使って $\sum_{i=1}^{k} \gamma_i P_i$ を求めることを推奨しているがその効果は定量的に書かれておらず不明である [11]．$\sum_{i=1}^{k} \gamma_i P_i$ の計算量は vector addition chain [27] などの方法で多少改良されることは知られている．

複数署名者の場合

$$\sum_{i=1}^{n} \alpha_i R_i + \sum_{i=1}^{n} (\alpha_i e_i \bmod \ell) A_i = \Big(\sum_{i=1}^{n} (\alpha_i z_i \bmod \ell)\Big) B \tag{11}$$

を検証することになる．左辺の $\sum_{i=1}^{k} \gamma_i P_i$ 型の計算において $k$ が単独署名者の場合の $n$ から $2n$ に増えるためバッチ検証の効果は少なくなる．

ECDSA 署名の場合，方式を多少（例えば文献 [28] のように）変形させると EdDSA 署名と同様のバッチ検証が適用できる．オリジナルの EdDSA 署名については，Karati ら [44] 及び Karati と Das [43] がバッチ検証法を提案しているが EdDSA 署名のバッチ検証法より効率が良いとは思えない．

## 6.3 楕円曲線上の加法演算の計算量比較

EdDSA 署名と ECDSA 署名はいずれも楕円曲線上の点 $B$ の大きなスカラー倍の点 $rB$ を計算する．これを計算する典型的な方法はスカラー $r$ をバイナリー表現にし，楕円曲線上の加法と 2 倍算を繰り返し用いることで合計 $O(\log_2(r))$ 回の演算で $rB$ を求める．平文のサイズが大きく無い場合，EdDSA 署名と ECDSA 署名の効率はこのスカラー倍計算の効率で決まる．

RFC8032 ではスカラー倍演算はツイスト Edwards 曲線上の（拡張）射影座標系で行うことが推

奨されているが，SEC1 ではアフィン座標系での演算ルールまでしか記述されていない．このため ECDSA 署名の効率は楕円曲線上のスカラー倍計算をどう実装するかで大きく変わってくる．楕円曲線上の実際の演算はアフィン座標系で直接計算するのではなく，射影座標や Jacobi 座標などの別の座標系に変換することで有限体上の逆算を避けることで高速化を実現することが普通であり，また複数の座標系を切り替えて計算する方法，制限された曲線に適用できる方法，そして実装の報告など，数多くの文献 [29, 22, 8, 13, 39, 10, 37, 46, 35] が存在する．

楕円曲線上の高速演算の情報については Bernstein のサイト [*9]が詳しいので参照して欲しい．

二つの署名方式の効率を比較するためには，使われる楕円曲線の離散対数問題が同程度の困難性を持っていなければ比較が成り立たない．よって，今回の比較では Ed25519 曲線と Curve25519 曲線のスカラー倍演算の計算量の比較と Ed448 曲線と Curve448 曲線のスカラー倍演算の計算量の比較を行うことにすることとする．Weierstrass 標準形で与えられた Curve25519 曲線と Curve448 曲線を射影座標系，Jacobi 座標系での加法と 2 倍算の計算量を [13] の結果を引用しツイスト Edwards 曲線上の推奨加法，2 倍算の計算量と比較する．

有限体 $\mathbb{F}_p$ 上の掛け算の計算量を $\mathbf{M}_p$，有限体 $\mathbb{F}_p$ 上の 2 乗演算の計算量を $\mathbf{S}_p$，有限体 $\mathbb{F}_p$ 上の元と楕円曲線のパラメータ定数との掛け算の計算量を $\mathbf{D}_p$ とする．$\mathbf{M}_p \geq \mathbf{S}_p$ である．文献 [13] には $\mathbf{S}_p \approx \mathbf{M}_p$ と $\mathbf{S}_p \approx 0.8\mathbf{M}_p$ の両方で比較した計算量が掲載されている．

- 射影座標，Weierstrass 曲線: $y^2 = x^3 + ax + b$ 上の点 $(x, y)$ を $Y^2Z = X^3 + aXZ^2 + bZ^3$ を満たす射影座標 $(X; Y; Z)$ で表す．ここで $(x, y)$ と $(X : Y : Z)$ は $x = X/Z$ と $y = Y/Z$ という関係を満たし，射影座標では $\lambda \in \mathbb{F}_p^\times$ なる全ての $\lambda$ に対して $(X : Y : Z) \sim (\lambda X : \lambda Y : \lambda Z)$（同じ点と見なす）．
- Jacobi 座標，Weierstrass 曲線: $y^2 = x^3 + ax + b$ 上の点 $(x, y)$ を $x = X/Z^2$ と $y = Y/Z^3$ と表現し $Y^2 = X^3 + aXZ^4 + bZ^6$ を満たす Jacobi 座標 $(X; Y; Z)$ で表す．ここで $\lambda \in \mathbb{F}_p^\times$ なる全ての $\lambda$ に対して $(X : Y : Z) \sim (\lambda^2 X : \lambda^3 Y : \lambda Z)$（同じ点と見なす）．
- 射影座標，ツイスト Edwards 曲線: $ax^2 + y^2 = 1 + dx^2y^2$ の点 $(x, y)$ を $x = X/Z$ と $y = Y/Z$ と表現し $(aX^2 + Y^2)Z^2 = Z^4 + dX^2Y^2$ を満たす射影座標 $(X : Y : Z)$ で表す．ここで $(x, y)$ と $(X : Y : Z)$ は $x = X/Z$ と $y = Y/Z$ という関係を満たし，射影座標では $\lambda \in \mathbb{F}_p - \{0\}$ なる全ての $\lambda$ に対して $(X : Y : Z) \sim (\lambda X : \lambda Y : \lambda Z)$（同じ点と見なす）．
- 拡張射影座標，$a = -1$ のツイスト Edwards 曲線: $-x^2 + y^2 = 1 + dx^2y^2$ の点 $(x, y)$ を，$(-X^2 + Y^2)Z^2 = Z^4 + dX^2Y^2$ を満たす（拡張）射影座標 $(X : Y : Z : T)$ で表す．ここで $(x, y)$ と $(X : Y : Z : T)$ は $x = X/Z$，$y = Y/Z$，$xy = T/Z$ という関係を満たし，（拡張）射影座標では $\lambda \in \mathbb{F}_p^\times$ なる全ての $\lambda$ に対して $(X : Y : Z : T) \sim (\lambda X : \lambda Y : \lambda Z : \lambda T)$（同じ点と見なす）．

[*9] http://hyperelliptic.org/EFD/g1p/index.html

表 3　楕円曲線上の加法の方式による比較

| 方式 | 加法の演算量 | 2 倍算の演算量 |
|---|---|---|
| Jacobian, Weierstrass (Curve25519) | $11\mathbf{M}_p + 5\mathbf{S}_p$ | $1\mathbf{M}_p + 8\mathbf{S}_p$ |
| Projective, Weierstrass (Curve25519) | $12\mathbf{M}_p + 2\mathbf{S}_p$ | $5\mathbf{M}_p + 6\mathbf{S}_p$ |
| Jacobian, Weierstrass (Curve448) | $11\mathbf{M}_p + 5\mathbf{S}_p$ | $1\mathbf{M}_p + 8\mathbf{S}_p$ |
| Projective, Weierstrass (Curve448) | $12\mathbf{M}_p + 2\mathbf{S}_p$ | $5\mathbf{M}_p + 6\mathbf{S}_p$ |
| Projective, Weierstrass (general) | $12\mathbf{M}_p + 2\mathbf{S}_p$ | $5\mathbf{M}_p + 6\mathbf{S}_p + 1\mathbf{D}_p$ |
| Extended Projective, Edwards (Ed25519) | $9\mathbf{M}_p + 1\mathbf{D}_p$ | $4\mathbf{M}_p + 4\mathbf{S}_p$ |
| Projective, Edwards (Ed448) | $10\mathbf{M}_p + 1\mathbf{S}_p$ | $3\mathbf{M}_p + 4\mathbf{S}_p$ |
| Projective, Edwards (general) | $10\mathbf{M}_p + 1\mathbf{S}_p + 2\mathbf{D}_p$ | $3\mathbf{M}_p + 4\mathbf{S}_p$ |

$\mathbf{M}_p$: 有限体 $\mathbb{F}_p$ 上の 2 元の掛け算.

$\mathbf{S}_p$: 有限体 $\mathbb{F}_p$ 上の元の 2 乗演算

$\mathbf{D}_p$: 有限体 $\mathbb{F}_p$ 上の元と楕円曲線の定数との掛け算

Curve25519 は $a = 486662$ であり，Curve488 は $a = 156326$ といずれも小さい定数のため $\mathbf{D}_p \approx 0$ とみなした．一般の $y^2 = x^3 + ax + b$ で表現される曲線の場合は $\mathbf{D}_p$ が省略されていない．同様に Ed448 では $a = 1$，$d = -39081$ と小さい定数であるため $\mathbf{D}_p \approx 0$ とみなした．一般の場合は $\mathbf{D}_p$ が省略されていない．Ed25519 は $a = -1$ の場合に特化した計算アルゴリズムになっている．

## 6.4　計算量比較まとめ

この章のこれまでの考察から次のことが言える．

- ツイスト Edwards 曲線上の演算はおそらく Weierstrass 曲線上の演算よりやや高速である．
- 平文がそれほど長く無い場合は，EdDSA 署名の署名生成が ECDSA 署名の署名生成より高速である [*10]．しかし，平文が長くなると EdDSA 署名の方ではノンス生成に時間がかかるため ECDSA より低速になる．ハッシュ関数の計算量が全体の計算量のほとんどになる場合，EdDSA 署名の署名生成は ECDSA 署名の署名生成のほぼ 2 倍の時間がかかるようになる．
- EdDSA 署名の署名検証が ECDSA 署名より署名検証より高速である．しかし平文が長くなるとハッシュ関数の計算量 $T_H$ が署名検証時間の大半を占めるため大差なくなる．

[*10] EdDSA 署名の署名生成において加法と 2 倍算で共通計算式を使った場合，平文が長くなくても ECDSA 署名と速度はほとんど変わらないかもしれない．しかし，この場合 ECDSA 署名はサイドチャネル攻撃を考慮していないので公平な比較とは言えない．

- 同一署名者の複数の署名を検証する場合，EdDSA 署名はバッチ検証処理が使え署名検証を署名数回繰り返すよりはかなり高速にできる．ECDSA 署名の場合，方式を多少変形させると（例えば文献 [28] などにあるように）EdDSA 署名と同様のバッチ検証が適用できるが，オリジナルの ECDSA 署名についてのバッチ検証を扱った論文は存在するが [44, 43]，EdDSA 署名と同程度に効率の良いバッチ検証技術は知られていない．

# 7 サイドチャネル攻撃

この章では EdDSA 署名に関係のあるサイドチャネル攻撃について述べる．

これまでノンスの偏りや一部の漏洩を利用した (EC)DSA 署名とその変形版 DSA 型署名への攻撃が多数発表されてきた [41, 47, 54, 55, 52, 32, 2, 72, 3]．これらは一部の漏洩したノンスから作られた多くの署名から Hidden Number Problem (HNP) [15] を構成し最終的に秘密鍵を見つけてしまう攻撃である。実際 ECDSA 署名の場合ノンスが数ビット漏洩していると５０～１５０個程度の署名から秘密鍵の復元が可能である．このとき HNP は格子の最近ベクトル問題 (CVP) もしくは最短ベクトル問題 (SVP) に埋め込まれて解かれる．HNP を解くのに格子に埋め込む以外にフーリエ変換を用いて解く方法が Bleichenbacher によって提案されている [14]．Bleichenbacher の攻撃では，格子の問題として解読するには漏洩が少なすぎるような場合も十分多くの署名を集めることで解読が可能になる．実際，[3] はノンスの漏洩が 1 ビット未満であっても 163-bit 曲線であれば $2^{24}$ 個，192-bit 曲線であれば $2^{35}$ 個の ECDSA 署名から秘密鍵を解読することに成功している．より詳細な条件及び Ed25519 レベルの曲線についての詳細を知りたい読者は [3] を読むことをお勧めする．

これらは DSA 型署名への攻撃であるが，EdDSA 署名もノンスの偏りや漏洩から DSA 型署名とほぼ同様に HNP を構成できるのでノンスの漏洩は EdDSA 署名にとっても重大な問題と捉えるべきである．

ノンスの偏りや漏洩は，例えば弱い疑似乱数生成器によるもの，スカラー倍算のときタイミング攻撃や電力解析攻撃で $r$ を推測するもの，フォルト攻撃により $r$ の一部を強制的にゼロクリアにするものなど様々な方法がある．EdDSA は加法と 2 倍算で共通式を使うことができるので，共通式を使うとスカラー倍算からタイミング攻撃や電力解析攻撃で $r$ を推測することが難しくなるが，EdDSA 署名のコードを提供している SUPERCOP [70] ではさらに本格的な対策として，スカラー倍算 $R = rB$ を生成するとき加法と 2 倍算はノンス $r$ の値によらず（群位数 $\ell$ のサイズの）定数回呼び出されるようになっている．このコードでは加法と 2 倍算は高速技法が使われているが $R$ の計算時間は定数となるのでタイミング攻撃や電力解析攻撃で $r$ を推測することが非常に難しくなっている．SUPERCOP [70] のコードはほとんどの EdDSA 署名の実装に使われているため，現実に実装された EdDSA 署名をタイミング攻撃や電力解析攻撃で攻撃することは現時点でかなり困難になっている．

参考として，Ed25519-EdDSA 署名のノンス生成に用いられる SHA512 からノンスを予測し攻撃する論文も存在する [64].

近年，HNP を用いて秘密鍵を求める攻撃とは別に，EdDSA 署名のようにノンスが確定的な確定型署名に対する新たなフォルト攻撃が提案されている．Aranha らは，[4] の第 3 章でこれらの攻撃についてまとめており，このレポートでは彼らの分類を紹介する．

- Special Soundness 攻撃 [5, 1, 62, 57, 63, 23, 25]: EdDSA 署名は Schnorr 署名の特徴から同一の $R = rB$ に対して異なる $e, e'$ に署名を生成してしまった場合，すなわち $(R, e, z)$ と $(R, e', z')$ という二つの署名が存在すると

  $$s = \frac{z - z'}{e - e'} \pmod{\ell}$$

  という関係が導かれる．このため異なる平文に対して異なるノンス $r$ が生成されるよう EdDSA 署名ではノンスが平文 $m$ と秘密情報 $s'$ のハッシュ値で生成されるようになっている．一方で同じ平文に繰り返し署名させてもノンスは同一となるため フォルト攻撃により $e$ を本来と異なる $e' \neq e$ $(e = H(R, A, m))$ に操作し最終的に上記のように $s$ 求めてしまう攻撃が考えられる．実際署名として出力されるのは $(R, z')$ なので $e'$ は何らかの手段で予測しなければいけないが，$e' - e$ の差分が小さくなるようなフォルト攻撃ができるのであれば $e'$ の値を容易に推測できる．最近，Cao ら [25] は，$e'$ を直接求めなくても HNP に帰着して秘密鍵を取り出せることを示している．
- Large Randomness Bias 攻撃 [5, 1, 23]: $e$ を直接操作する代わりにノンスの $r$ を操作し $r + \Delta$ というノンスに署名を付けさせる．もし差分 $\Delta$ を求めることができるのであれば $R' = R + \Delta B$ を求めることができ $e'$ も計算可能であるため，

  $$s = \frac{z - z' - \Delta}{e - e'} \pmod{\ell}$$

  により $s$ を求めることができる．

上記の攻撃はいずれも同一平文に対するノンスは確定であるというノンス確定型署名の特徴を使って攻撃している．一方，ノンス確定型署名に限らない フォルト攻撃として，ノンスを強制的に同じ値にしたり [1]，署名の計算にノンスを強制的に含めなくする [59] ような攻撃も提案されている．上記のようなことをされてしまえば当然秘密鍵は導出できてしまう．

以下は EdDSA 署名にではなく署名方式（または暗号アルゴリズム）一般に対する見解である．攻撃者側にフォルト攻撃を自由に許すような環境においては，いかなる方式も安全性を確保することは容易ではない．よって組み込みデバイスで使わなければいけない場合はなんらかの対策がなされていることが望ましい．一方，攻撃者側が物理的にデバイスにアクセスすることが難しい場合は，現在のところタイミング攻撃に対する対策ができていれば十分に思われるが，リモートであっても電力解析やフォルト攻撃 (RowHammer attack) ができたとの報告もあるので，将来的にはサ

イドチャネル攻撃耐性というのはより重要になってくるものと思われる．

# 8 まとめ

結論をいうと EdDSA 署名は，ECDSA 署名と比べても安全な署名だと思われる．

証明可能安全性の立場では，推奨パラメータ Ed25519 を利用した場合，ノンスの出力が疑似ランダム関数の出力と見做せないため安全性証明がつかないというデメリットがある．仮にノンスを HMAC 等の安全な疑似ランダム関数をみなせるものから生成していれば証明可能安全性の立場からはより望ましかった．推奨パラメータ Ed448 ではハッシュ関数をランダム関数とみなしランダムオラクルモデルで安全性証明をつけることが可能である．ただしこの場合でもビット安全性的には望まれる安全性の半分の 112 ビット程度しか保証できない．群の一般的性質のみを使い攻撃する generic group model での安全性を考えると，Ed448 は望まれる 224 ビット安全性を得ることが可能に思える．一方，Ed25519 は generic group model でも安全性証明をつけるにはノンスの出力が疑似ランダム関数と見做せないため障害がある．

以上は証明可能安全性の立場から述べたが，EdDSA 署名が Schnorr 署名という非常に研究された成熟した方式をもとにしていること，署名の内部乱数を弱い疑似乱数生成器にまかすことの危険を排除し，その作り方を仕様で明示したこと，またハッシュ関数を用いることで異なる平文に対する署名生成時にノンスがぶつからないように配慮したことなど，現実には安全性への高い配慮が感じられる．

サイドチャネル攻撃に関して，EdDSA 署名はノンスが漏洩しづらくする工夫をしているためノンスの漏洩を利用する攻撃に対して ECDSA 署名より安全と考えても良い．

# 謝辞

本報告書を作成するにあたり，6.2 章は NTT セキュアプラットフォーム研究所の星野文学氏に，7 章は Aarhus 大学の高橋彰氏にご協力を頂いたことに深く感謝する．高橋氏には初期原稿 4.6 章における筆者の認識誤りについてもご指摘頂いたこと深く感謝する．

本報告書の誤りの責任は全て筆者に帰す．

# 参考文献


[1] Christopher Ambrose, Joppe W. Bos, Björn Fay, Marc Joye, Manfred Lochter, and Bruce Murray. Differential attacks on deterministic signatures. In Nigel P. Smart, editor, *Topics in Cryptology – CT-RSA 2018*, Vol. 10808 of *Lecture Notes in Computer Science*, pp. 339–353. Springer, Heidelberg, April 2018.

[2] Diego F. Aranha, Pierre-Alain Fouque, Benoît Gérard, Jean-Gabriel Kammerer, Mehdi Tibouchi, and Jean-Christophe Zapalowicz. GLV/GLS decomposition, power analysis, and attacks on ECDSA signatures with single-bit nonce bias. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part I*, Vol. 8873 of *Lecture Notes in Computer Science*, pp. 262–281. Springer, Heidelberg, December 2014.

[3] Diego F. Aranha, Felipe Rodrigues Novaes, Akira Takahashi, Mehdi Tibouchi, and Yuval Yarom. LadderLeak: Breaking ECDSA with less than one bit of nonce leakage. In *ACM CCS 20: 27th Conference on Computer and Communications Security*, pp. 225–242. ACM Press, 2020.

[4] Diego F. Aranha, Claudio Orlandi, Akira Takahashi, and Greg Zaverucha. Security of hedged Fiat-Shamir signatures under fault attacks. Cryptology ePrint Archive, Report 2019/956, 2019. `https://eprint.iacr.org/2019/956`.

[5] Alessandro Barenghi and Gerardo Pelosi. A note on fault attacks against deterministic signature schemes. In Kazuto Ogawa and Katsunari Yoshioka, editors, *IWSEC 16: 11th International Workshop on Security, Advances in Information and Computer Security*, Vol. 9836 of *Lecture Notes in Computer Science*, pp. 182–192. Springer, Heidelberg, September 2016.

[6] Mihir Bellare, Juan A. Garay, and Tal Rabin. Fast batch verification for modular exponentiation and digital signatures. In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, Vol. 1403 of *Lecture Notes in Computer Science*, pp. 236–250. Springer, Heidelberg, May / June 1998.

[7] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006: 13th Conference on Computer and Communications Security*, pp. 390–399. ACM Press, October / November 2006.

[8] Daniel J. Bernstein. Curve25519: New Diffie-Hellman speed records. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006: 9th International Conference on Theory and Practice of Public Key Cryptography*, Vol. 3958 of *Lecture*

*Notes in Computer Science*, pp. 207–228. Springer, Heidelberg, April 2006.

[9] Daniel J. Bernstein. Multi-user Schnorr security, revisited. Cryptology ePrint Archive, Report 2015/996, 2015. `http://eprint.iacr.org/2015/996`.

[10] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted Edwards curves. Cryptology ePrint Archive, Report 2008/013, 2008. `http://eprint.iacr.org/2008/013`.

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

[12] Daniel J. Bernstein, Simon Josefsson, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. EdDSA for more curves. Cryptology ePrint Archive, Report 2015/677, 2015. `http://eprint.iacr.org/2015/677`.

[13] Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. Cryptology ePrint Archive, Report 2007/286, 2007. `http://eprint.iacr.org/2007/286`.

[14] Daniel Bleichenbacher. On the generation of one-time keys in DL signature schemes. Presentation at IEEE P1363 working group meeting, 2000.

[15] Dan Boneh and Ramarathnam Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In Neal Koblitz, editor, *Advances in Cryptology – CRYPTO'96*, Vol. 1109 of *Lecture Notes in Computer Science*, pp. 129–142. Springer, Heidelberg, August 1996.

[16] Jurjen N. Bos and Matthijs J. Coster. Addition chain heuristics. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, Vol. 435 of *Lecture Notes in Computer Science*, pp. 400–407. Springer, Heidelberg, August 1990.

[17] Daniel R. L. Brown. The exact security of ECDSA. Contributions to IEEE P1363a, January 2001. `http://grouper.ieee.org/groups/1363/`.

[18] Daniel R. L. Brown. Generic groups, collision resistance, and ECDSA. Contributions to IEEE P1363a, February 2002. Updated version for "The Exact Security of ECDSA." Available from `http://grouper.ieee.org/groups/1363/`.

[19] Daniel R. L. Brown. Generic groups, collision resistance, and ECDSA. Cryptology ePrint Archive, Report 2002/026, 2002. `http://eprint.iacr.org/2002/026`.

[20] Daniel R. L. Brown. Generic groups, collision resistance, and ECDSA. *Des. Codes Cryptogr.*, Vol. 35, No. 1, pp. 119–152, 2005.

[21] Daniel R. L. Brown. Short Schnorr signatures require a hash function with more than just random-prefix resistance. Cryptology ePrint Archive, Report 2015/169, 2015. `http://eprint.iacr.org/2015/169`.

[22] Michael Brown, Darrel Hankerson, Julio Cesar López-Hernández, and Alfred Menezes. Software implementation of the NIST elliptic curves over prime fields. In David Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, Vol. 2020 of *Lecture Notes in Computer Science*, pp. 250–265. Springer, Heidelberg, April 2001.

[23] Leon Groot Bruinderink and Peter Pessl. Differential fault attacks on deterministic lattice signatures. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, Vol. 2018, No. 3, pp. 21–43, 2018. `https://tches.iacr.org/index.php/TCHES/article/view/7267`.

[24] Jan Camenisch, Susan Hohenberger, and Michael Østergaard Pedersen. Batch verification of short signatures. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, Vol. 4515 of *Lecture Notes in Computer Science*, pp. 246–263. Springer, Heidelberg, May 2007.

[25] Weiqiong Cao, Hongsong Shi, Hua Chen, Jiazhe Chen, Limin Fan, and Wenling Wu. Lattice-based fault attacks on deterministic signature schemes of ECDSA and EdDSA. Cryptology ePrint Archive, Report 2020/803, 2020. `https://eprint.iacr.org/2020/803`.

[26] Konstantinos Chalkias, François Garillot, and Valeria Nikolaenko. Taming the many EdDSAs. Security Standardisation Research Conference (SSR 2020), 2020.

[27] Yuh-Jiun Chen, Chin-Chen Chang, and Wei-Pang Yang. Some properties of vectorial addition chains. *International Journal of Computer Mathematics*, Vol. 54, No. 3-4, pp. 185–196, 1994.

[28] Jung Hee Cheon and Jeong Hyun Yi. Fast batch verification of multiple signatures. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007: 10th International Conference on Theory and Practice of Public Key Cryptography*, Vol. 4450 of *Lecture Notes in Computer Science*, pp. 442–457. Springer, Heidelberg, April 2007.

[29] Henri Cohen, Atsuko Miyaji, and Takatoshi Ono. Efficient elliptic curve exponentiation using mixed coordinates. In Kazuo Ohta and Dingyi Pei, editors, *Advances in Cryptology – ASIACRYPT'98*, Vol. 1514 of *Lecture Notes in Computer Science*, pp. 51–65. Springer, Heidelberg, October 1998.

[30] Cryptography Research and Evaluation Committees. `https://www.cryptrec.go.jp/`.

[31] Ivan Damgård. A design principle for hash functions. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, Vol. 435 of *Lecture Notes in Computer Science*, pp. 416–427. Springer, Heidelberg, August 1990.

[32] Elke De Mulder, Michael Hutter, Mark E. Marson, and Peter Pearson. Using Bleichenbacher's solution to the hidden number problem to attack nonce leaks in 384-bit ECDSA:

extended version. *Journal of Cryptographic Engineering*, Vol. 4, No. 1, pp. 33–45, April 2014.

[33] Peter de Rooij. Efficient exponentiation using procomputation and vector addition chains. In Alfredo De Santis, editor, *Advances in Cryptology – EUROCRYPT'94*, Vol. 950 of *Lecture Notes in Computer Science*, pp. 389–399. Springer, Heidelberg, May 1995.

[34] Harold M. Edwards. A normal form for elliptic curves. In *Bulletin of the American Mathematical Society*, pp. 393–422, 2007.

[35] Björn Fay. Double-and-add with relative Jacobian coordinates. Cryptology ePrint Archive, Report 2014/1014, 2014. `http://eprint.iacr.org/2014/1014`.

[36] Manuel Fersch, Eike Kiltz, and Bertram Poettering. On the provable security of (EC)DSA signatures. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pp. 1651–1662. ACM Press, October 2016.

[37] Steven D. Galbraith, Xibin Lin, and Michael Scott. Endomorphisms for faster elliptic curve cryptography on a large class of curves. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, Vol. 5479 of *Lecture Notes in Computer Science*, pp. 518–535. Springer, Heidelberg, April 2009.

[38] Steven D. Galbraith, John Malone-Lee, and Nigel P. Smart. Public key signatures in the multi-user setting. *Inf. Process. Lett.*, Vol. 83, No. 5, pp. 263–266, 2002.

[39] Pierrick Gaudry and Emmanuel Thome. The mpFq library and implementing curve-based key exchanges. Technical report, Institut National de Recherche en Informatique et en Automatique, 2007.

[40] Keisuke Hakuta, Yosuke Katoh, Hisayoshi Sato, and Tsuyoshi Takagi. Batch verification suitable for efficiently verifying a limited number of signatures. In Taekyoung Kwon, Mun-Kyu Lee, and Daesung Kwon, editors, *ICISC 12: 15th International Conference on Information Security and Cryptology*, Vol. 7839 of *Lecture Notes in Computer Science*, pp. 425–440. Springer, Heidelberg, November 2013.

[41] Nick Howgrave-Graham and Nigel Smart. Lattice attacks on digital signature schemes. *Designs, Codes and Cryptography*, Vol. 23, pp. 283–290, 2001.

[42] Jung Hee Cheon and Dong Hoon Lee. Use of sparse and/or complex exponents in batch verification of exponentiations. *IEEE Transactions on Computers*, Vol. 55, No. 12, pp. 1536–1542, 2006.

[43] Sabyasachi Karati and Abhijit Das. Faster batch verification of standard ECDSA signatures using summation polynomials. In Ioana Boureanu, Philippe Owesarski, and Serge Vaudenay, editors, *ACNS 14: 12th International Conference on Applied Cryptography*

*and Network Security*, Vol. 8479 of *Lecture Notes in Computer Science*, pp. 438–456. Springer, Heidelberg, June 2014.

[44] Sabyasachi Karati, Abhijit Das, Dipanwita Roy Chowdhury, Bhargav Bellur, Debojyoti Bhattacharya, and Aravind Iyer. Batch verification of ECDSA signatures. In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *AFRICACRYPT 12: 5th International Conference on Cryptology in Africa*, Vol. 7374 of *Lecture Notes in Computer Science*, pp. 1–18. Springer, Heidelberg, July 2012.

[45] John Kelsey and Tadayoshi Kohno. Herding hash functions and the Nostradamus attack. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, Vol. 4004 of *Lecture Notes in Computer Science*, pp. 183–200. Springer, Heidelberg, May / June 2006.

[46] Patrick Longa and Catherine H. Gebotys. Efficient techniques for high-speed elliptic curve cryptography. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems – CHES 2010*, Vol. 6225 of *Lecture Notes in Computer Science*, pp. 80–94. Springer, Heidelberg, August 2010.

[47] Edwin El Mahassni, Phong Q. Nguyen, and Igor E. Shparlinski. The insecurity of Nyberg-Rueppel and other DSA-like signature schemes with partially known nonces. In Joseph H. Silverman, editor, *Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001, Revised Papers*, Vol. 2146 of *Lecture Notes in Computer Science*, pp. 97–109. Springer, 2001.

[48] Alfred Menezes and Nigel P. Smart. Security of signature schemes in a multi-user setting. *Des. Codes Cryptogr.*, Vol. 33, No. 3, pp. 261–274, 2004.

[49] Ralph C. Merkle. A certified digital signature. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, Vol. 435 of *Lecture Notes in Computer Science*, pp. 218–238. Springer, Heidelberg, August 1990.

[50] Hiraku Morita, Jacob C. N. Schuldt, Takahiro Matsuda, Goichiro Hanaoka, and Tetsu Iwata. On the security of the Schnorr signature scheme and DSA against related-key attacks. In Soonhak Kwon and Aaram Yun, editors, *ICISC 15: 18th International Conference on Information Security and Cryptology*, Vol. 9558 of *Lecture Notes in Computer Science*, pp. 20–35. Springer, Heidelberg, November 2016.

[51] David Naccache, David M'Raïhi, Serge Vaudenay, and Dan Raphaeli. Can D.S.A. be improved? Complexity trade-offs with the digital signature standard. In Alfredo De Santis, editor, *Advances in Cryptology – EUROCRYPT'94*, Vol. 950 of *Lecture Notes in Computer Science*, pp. 77–85. Springer, Heidelberg, May 1995.

[52] David Naccache, Phong Q. Nguyen, Michael Tunstall, and Claire Whelan. Experimenting with faults, lattices and the DSA. In Serge Vaudenay, editor, *PKC 2005: 8th International*

*Workshop on Theory and Practice in Public Key Cryptography*, Vol. 3386 of *Lecture Notes in Computer Science*, pp. 16–28. Springer, Heidelberg, January 2005.

[53] Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. Hash function requirements for Schnorr signatures. *J. Math. Cryptol.*, Vol. 3, No. 1, pp. 69–87, 2009.

[54] Phong Q. Nguyen and Igor E. Shparlinski. The insecurity of the digital signature algorithm with partially known nonces. *Journal of Cryptology*, Vol. 15, No. 3, pp. 151–176, June 2002.

[55] Phong Q. Nguyen and Igor E. Shparlinski. The insecurity of the elliptic curve digital signature algorithm with partially known nonces. *Des. Codes Cryptogr.*, Vol. 30, No. 2, pp. 201–217, 2003.

[56] 岡本龍明. ECDSA 評価報告書. 外部評価報告書 CRYPTREC EX-0004-2002, CRYPTREC, 2002.

[57] D. Poddebniak, J. Somorovsky, S. Schinzel, M. Lochter, and P. Rösler. Attacking deterministic signature schemes using fault attacks. In *2018 IEEE European Symposium on Security and Privacy (EuroSP)*, pp. 338–352, 2018.

[58] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, Vol. 13, No. 3, pp. 361–396, June 2000.

[59] Prasanna Ravi, Mahabir Prasad Jhanwar, James Howe, Anupam Chattopadhyay, and Shivam Bhasin. Exploiting determinism in lattice-based signatures: Practical fault attacks on pqm4 implementations of NIST candidates. In Steven D. Galbraith, Giovanni Russello, Willy Susilo, Dieter Gollmann, Engin Kirda, and Zhenkai Liang, editors, *ASIACCS 19: 14th ACM Symposium on Information, Computer and Communications Security*, pp. 427–440. ACM Press, July 2019.

[60] RFC7748: Request for Comments. `https://www.rfc-editor.org/info/rfc7748`.

[61] RFC8032: Request for Comments. `https://www.rfc-editor.org/info/rfc8032`.

[62] Y. Romailler and S. Pelissier. Practical fault attack against the Ed25519 and EdDSA signature schemes. In *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pp. 17–24, 2017.

[63] Niels Samwel and Lejla Batina. Practical fault injection on deterministic signatures: The case of EdDSA. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 18: 10th International Conference on Cryptology in Africa*, Vol. 10831 of *Lecture Notes in Computer Science*, pp. 306–321. Springer, Heidelberg, May 2018.

[64] Niels Samwel, Lejla Batina, Guido Bertoni, Joan Daemen, and Ruggero Susella. Breaking Ed25519 in WolfSSL. Cryptology ePrint Archive, Report 2017/985, 2017. `http://eprint.iacr.org/2017/985`.

[65] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, Vol. 435 of *Lecture Notes in Computer Science*, pp. 239–252. Springer, Heidelberg, August 1990.

[66] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, Vol. 4, No. 3, pp. 161–174, January 1991.

[67] 新保淳, 丹羽朗人, 岡田光司. ECDSA 評価報告書. 外部評価報告書 CRYPTREC EX-0003-2001, CRYPTREC, 2001.

[68] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, Vol. 1233 of *Lecture Notes in Computer Science*, pp. 256–266. Springer, Heidelberg, May 1997.

[69] Jacques Stern, David Pointcheval, John Malone-Lee, and Nigel P. Smart. Flaws in applying proof methodologies to signature schemes. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, Vol. 2442 of *Lecture Notes in Computer Science*, pp. 93–110. Springer, Heidelberg, August 2002.

[70] SUPERCOP: eBACS: ECRYPT Benchmarking of Cryptographic Systems. `https://bench.cr.yp.to/supercop.html`.

[71] S. Yen and C. Laih. Improved digital signature suitable for batch verification. *IEEE Transactions on Computers*, Vol. 44, No. 7, pp. 957–959, 1995.

[72] Akira Takahashi, Mehdi Tibouchi, and Masayuki Abe. New Bleichenbacher records: Fault attacks on qDSA signatures. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, Vol. 2018, No. 3, pp. 331–371, 2018. `https://tches.iacr.org/index.php/TCHES/article/view/7278`.

[73] Serge Vaudenay. The security of DSA and ECDSA. In Yvo Desmedt, editor, *PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography*, Vol. 2567 of *Lecture Notes in Computer Science*, pp. 309–323. Springer, Heidelberg, January 2003.