

# 鍵リカバリ型 ECDSA の BUFF 安全性について

江村 恵太<sup>1,2,a)</sup>

**概要：**通常の署名方式では、署名検証アルゴリズムは検証鍵、署名、メッセージを入力とし、1 または 0 を出力する。一方、Ethereum で利用されている ECDSA 方式の署名検証アルゴリズムでは、検証鍵は明示的にアルゴリズムの入力に含まれず、署名とメッセージから検証鍵を復元する。本論文では、この鍵リカバリ型 ECDSA 方式に対する BUFF (Beyond UnForgeability Features (Cremers et al., IEEE S&P 2021)) 安全性について調査、weak non-resignability (wNR) を除く BUFF 安全性を満たすことを示す。本成果により、復元された検証鍵のハッシュ値と Ethereum アドレスとを比較する手順が BUFF 安全性の達成に必要な不可欠であることを明らかにした。さらに wNR 安全性について、Ethereum におけるトランザクションへの署名用途においては必要ないと想定されることも述べる。さらに鍵リカバリ型 ECDSA のどの部分が BUFF 安全性の達成に起因しているのかを明確にするため、通常の ECDSA に対する BUFF 攻撃も提案する。その応用として、ECDSA に対する提案 BUFF 攻撃の 1 つが Aumayr らの ECDSA ベース Adaptor 署名 (ASIACRYPT 2021)、Qin らの ECDSA ベース Blind Adaptor 署名 (IEEE S&P 2023) に対しても有効であることも示す。なお本攻撃は Aumayr ら、Qin らの安全性モデル外の攻撃であることを強調しておく。

**キーワード：**ECDSA, 鍵リカバリ, BUFF 安全性

## On Beyond Unforgeability Features of ECDSA with Key Recovery

KEITA EMURA<sup>1,2,a)</sup>

**Abstract:** In the usual syntax of digital signatures, the verification algorithm takes a verification key in addition to a signature and a message, whereas in ECDSA with key recovery, which is used in Ethereum, no verification key is input to the verification algorithm. Instead, a verification key is recovered from a signature and a message. In this paper, we explore BUFF security of ECDSA with key recovery (KR-ECDSA), where BUFF stands for Beyond UnForgeability Features (Cremers et al., IEEE S&P 2021). As a result, we show that KR-ECDSA provides BUFF security, except weak non-resignability (wNR). We also show that the original ECDSA does not provide any BUFF security. As a by-product of the analysis, we show that one of our BUFF attacks also works against the Aumayr et al.'s ECDSA-based adaptor signature scheme (ASIACRYPT 2021) and the Qin et al.'s blind adaptor signature scheme (IEEE S&P 2023), which is based on the Aumayr et al.'s scheme. We emphasize that the attack is positioned outside of their security models.

**Keywords:** ECDSA with Key Recovery, BUFF Security

### 1. はじめに

楕円曲線を用いた署名方式とその安全性. ECDSA (Elliptic

Curve Digital Signature Algorithm) はブロックチェーンコミュニティにおいて広く認知されており、実際のシステムにてトランザクションの検証に用いられている。ECDSA の安全性については広く検証されており、例えば Vaudenay [36] による Domain Parameter Shifting 攻撃、Ferscher ら [17] による楕円曲線上の離散対数問題の困難性に基づく ECDSA の存在的偽造不可能性の証明、Groth と Shoup [21] による

<sup>1</sup> 金沢大学  
Kanazawa University

<sup>2</sup> 産業技術総合研究所  
National Institute of Advanced Industrial Science and Technology (AIST)

<sup>a)</sup> k-emura@se.kanazawa-u.ac.jp

| Transformation | Signature  | S-CEO | S-DEO | M-S-UEO | MBS | NR |
|----------------|--|-------|-------|---------|-----|----|
| PS-1           | $\text{Sign}(\text{sk}, m), H(m)$                          |       | ✓     |         | ✓   |    |
| PS-2           | $\text{Sign}(\text{sk}, m), H(\text{vk})$                  | ✓     | ✓     | ✓       |     |    |
| BUFF-lite      | $\text{Sign}(\text{sk}, m), H(m, \text{vk})$               | ✓     | ✓     | ✓       | ✓   |    |
| BUFF           | $\text{Sign}(\text{sk}, H(m, \text{vk})), H(m, \text{vk})$ | ✓     | ✓     | ✓       | ✓   | ✓  |

表 1 BUFF 変換 [7]

フォーマットチェック (後述) を付加した ECDSA の強存在的偽造不可能性の証明, Hartmann と Kiltz [22] による代数的帰着の存在性と programmability との関係性の評価などが挙げられる。

Brendel ら [4] は Ed25519 [3] (EdDSA を Curve25519 で実装した方式) の安全性を評価した。通常の (強) 存在的偽造不可能性に加え, 彼らは BUFF (Beyond UnForgeability Features) 安全性 [1, 7] (S-UEO (Strong Universal Exclusive Ownership), M-S-UEO (Malicious Strong Universal Exclusive Ownership), MBS (Message Bound Security)) についても評価している。

**一般的変換.** ここで Cremers ら [7] により纏められた, 通常の署名に BUFF 安全性を付加する一般的変換を紹介する (Table 1). Pornin と Stern [30] により与えられた変換を PS-1, PS-2, Cremers らにより与えられた変換を BUFF-lite<sup>\*1</sup>, BUFF と表記する。Sign を署名アルゴリズム, (vk, sk) を検証鍵と署名鍵のペアとする。なお PS-3: ( $\text{Sign}(\text{sk}, H(m, \text{vk}))$ ) はテーブルから除外した。これは追加で弱鍵の非存在性を仮定するためである。<sup>\*2</sup> Cremers ら [7] により S-CEO (Strong Conservative Exclusive Ownership) と S-DEO (Strong Destructive Exclusive Ownership) は S-UEO と等価であり, S-UEO は M-S-UEO に含まれることが示されている。Aulbach ら [1] は署名方式が S-CEO, S-DEO, MBS, wNR (weak Non-Resignability) を満たすとき, その署名方式は Full BUFF 安全であると定義している。

**鍵リカバリ型 ECDSA.** 通常の署名方式では, 署名検証アルゴリズムは検証鍵, 署名, メッセージを入力とし, 1 または 0 を出力する。一方, Ethereum で利用されている ECDSA 方式の署名検証アルゴリズムでは, 検証鍵は明示的にアルゴリズムの入力に含まれず, 署名とメッセージから検証鍵を復元する。本論文では, この ECDSA 方式を KR-ECDSA (ECDSA with key recovery) と呼ぶ。なお Ethereum Yellow Paper [37] では recoverable ECDSA と呼んでいる。本論文では何が recoverable なのかを明示するため, key recovery とした。KR-ECDSA の検証アルゴリズムでは, 署名の一部  $s$  が  $0 < s < q/2 + 1$  (ここで  $q$  は楕

円曲線の位数) を満たすことも確認する (以降, この確認をフォーマットチェックと呼ぶ)。また署名とメッセージに加え, Ethereum address `addr` も入力に取る。ここで `addr` は ECDSA 検証鍵のハッシュ値 (正確には Keccak-256 ハッシュ値の下位 160 ビット) であり, 署名とメッセージから復元された検証鍵のハッシュ値が `addr` と一致するかどうかを確認する。

**我々の動機.** 我々の知る限り, BUFF 安全性を含む KR-ECDSA の安全性についてはこれまで明示的に評価されていない。より正確には, Ethereum Yellow Paper [37] は Johnson, Menezes, Vanstone のドキュメント [24] を引用するだけで, 明示的に KR-ECDSA の安全性について評価していない。元々 BUFF 安全性は PQC (Post-Quantum Cryptography) 署名 [1, 7, 14, 25] に対して評価されてきたが, 現在実用化されている署名方式に対する BUFF 安全性の評価も重要であると考えられる。特に Ethereum ではトランザクションが KR-ECDSA により署名され, 巨額の暗号資産が取引されている現状を鑑みると, 潜在的な攻撃の可能性について明確することは非常に重要であると考えられる。この動機に基づき, 本論文では KR-ECDSA の BUFF 安全性を評価する。

**本論文の貢献.** 本論文<sup>\*3</sup>では, この鍵リカバリ型 ECDSA 方式に対する BUFF 安全性について調査, weak non-resignability (wNR) を除く BUFF 安全性を満たすことを示す (wNR に対しては具体的な攻撃を示す)。本成果により, 復元された検証鍵のハッシュ値と Ethereum アドレスとを比較する手順が BUFF 安全性の達成に必要不可欠であることを明らかにした。さらに wNR 安全性について, Ethereum におけるトランザクションへの署名用途においては必要ないと想定されることも述べる。さらに鍵リカバリ型 ECDSA のどの部分が BUFF 安全性の達成に起因しているのかを明確にするため, 通常の ECDSA に対する BUFF 攻撃を提案する。我々の成果を Table 2 に纏める。また ECDSA に対する提案 BUFF 攻撃の 1 つが Aumayr らの ECDSA ベース Adaptor 署名 [2], Qin らの ECDSA ベース Blind Adaptor 署名 [31] に対しても有効であることも示す。なお本攻撃は Aumayr ら, Qin らの安全性モデル外の攻撃であることを強調しておく。

**研究倫理.** 本論文では, KR-ECDSA に対する具体的な wNR

<sup>\*1</sup> 本論文では ePrint 版 (posted on October 2023 [6]) を明示的に引用する。これは Don ら [13] による指摘を受けて更新された版である。

<sup>\*2</sup> なお Düzlülü と Struck [15] により, MBS を満たせば弱鍵が存在しないこと, もし変換前の署名方式が MBS を満たせば PS-3 変換で BUFF 安全となることが示されている。

<sup>\*3</sup> 本論文の ePrint 版はこちら [16]。

|          | Strong EUF-CMA   | S-CEO | S-DEO | M-S-UEO | MBS | wNR |
|----------|------------------|-------|-------|---------|-----|-----|
| ECDSA    | ✓ (フォーマットチェック付き) |       |       |         |     |     |
| KR-ECDSA | ✓                | ✓     | ✓     | ✓       | ✓   |     |

Groth と Shoup [21] の評価に基づき、フォーマットチェック付き ECDSA を強存在的偽造不可能であると仮定する。  
Cremers ら [7] により ECDSA が MBS を満たさないことは既に言及されている。

表 2 本論文の貢献

攻撃を示す。これは潜在的に実際の Ethereum トランザクションに対して何等かの影響を与える可能性がある。しかし後ほど述べるように、(少なくとも Ethereum の文脈において) この攻撃は現実的な脅威ではないと主張する。さらに我々は (Blind) Adaptor 署名の応用における提案 BUFF 攻撃による脆弱性も現状見つけられていない。また著者は Ethereum アカウントを所持していない。

## 2. 定義

署名方式を  $\text{Sig} = (\text{KeyGen}, \text{Sign}, \text{Verify})$  とする。鍵生成アルゴリズムはセキュリティパラメータ  $\lambda$  を入力とし、検証鍵と署名鍵のペア  $(vk, sk)$  を出力する。署名アルゴリズム  $\text{Sign}$  は  $sk$  と署名するメッセージ  $M$  を入力とし、署名  $\sigma$  を出力する。検証アルゴリズム  $\text{Verify}$  は  $vk, M, \sigma$  を入力とし、0 または 1 を返す。任意の  $\lambda \in \mathbb{N}$ ,  $(vk, sk) \leftarrow \text{KeyGen}(1^\lambda)$ ,  $M$  に対し、 $\text{Verify}(vk, M, \text{Sign}(sk, M)) = 1$  がセキュリティパラメータ  $\lambda$  に関し圧倒的な確率で成り立つとき、署名方式は正当性 (Correctness) を満たすと定義する。

### 2.1 強存在的偽造不可能性

$\mathcal{A}$  を攻撃者、 $\mathcal{C}$  をチャレンジャーとする。 $\mathcal{C}$  は  $(vk, sk) \leftarrow \text{KeyGen}(1^\lambda)$  を実行し、 $vk$  を  $\mathcal{A}$  に与えるとともに  $\text{SigSet} = \emptyset$  と初期化する。 $\mathcal{A}$  の署名クエリ  $M$  に対し、 $\mathcal{C}$  は  $\sigma \leftarrow \text{Sign}(sk, M)$  を実行し、 $\sigma$  を  $\mathcal{A}$  に返すとともに  $(M, \sigma)$  を  $\text{SigSet}$  に保存する。最後に  $\mathcal{A}$  は偽造  $(M^*, \sigma^*)$  を出力する。

$$\text{Verify}(vk, M^*, \sigma^*) = 1 \wedge (M^*, \sigma^*) \notin \text{SigSet}$$

のとき、 $\mathcal{A}$  が勝利すると定義する。 $\mathcal{A}$  の利得を  $\text{Adv}_{\mathcal{A}, \text{Sig}}^{\text{strong}}(\lambda) = \Pr[\mathcal{A} \text{ wins}]$  と定義する。もし  $\text{Adv}_{\mathcal{A}, \text{Sig}}^{\text{strong}}(\lambda)$  が全ての確率的多項式時間攻撃者  $\mathcal{A}$  に対して無視できるとき、署名方式  $\text{Sig}$  は選択文書攻撃に対して強存在的偽造不可能, strong EUF-CMA (strongly existentially unforgeability against chosen message attack) 安全であると定義する。 $\mathcal{A}$  が  $M^*$  を署名クエリとして送付することを禁止する場合、 $\text{Sig}$  は EUF-CMA 安全であると定義する。

### 2.2 S-CEO, S-DEO, M-S-UEO

S-CEO は、元々の検証鍵で検証に通る署名に対し、検証に通る別の検証鍵を生成できないことを保証する安全性である。つまり  $\text{Verify}(vk, M^*, \sigma^*) = 1$  を満たす  $(M^*, \sigma^*)$  に対し、 $vk^* \neq vk$  かつ  $\text{Verify}(vk^*, M^*, \sigma^*) = 1$  が成り立つと

き、攻撃者の勝利とする。形式的な定義は以下。 $\mathcal{A}$  を攻撃者、 $\mathcal{C}$  をチャレンジャーとする。 $\mathcal{C}$  は  $(vk, sk) \leftarrow \text{KeyGen}(1^\lambda)$  を実行し、 $vk$  を  $\mathcal{A}$  に与えるとともに  $\text{SigSet} = \emptyset$  と初期化する。 $\mathcal{A}$  の署名クエリ  $M$  に対し、 $\mathcal{C}$  は  $\sigma \leftarrow \text{Sign}(sk, M)$  を実行し、 $\sigma$  を  $\mathcal{A}$  に返すとともに  $(M, \sigma)$  を  $\text{SigSet}$  に保存する。最後に  $\mathcal{A}$  は  $(vk^*, M^*, \sigma^*)$  を出力する。

$$\text{Verify}(vk^*, M^*, \sigma^*) = 1 \wedge (M^*, \sigma^*) \in \text{SigSet} \wedge vk^* \neq vk$$

のとき、 $\mathcal{A}$  が勝利すると定義する。 $\mathcal{A}$  の利得を  $\text{Adv}_{\mathcal{A}, \text{Sig}}^{\text{S-CEO}}(\lambda) = \Pr[\mathcal{A} \text{ wins}]$  と定義する。もし  $\text{Adv}_{\mathcal{A}, \text{Sig}}^{\text{S-CEO}}(\lambda)$  が全ての確率的多項式時間攻撃者  $\mathcal{A}$  に対して無視できるとき、署名方式  $\text{Sig}$  は S-CEO 安全であると定義する。

S-DEO 安全性も S-CEO 安全性と同様に定義されるが、攻撃者が元とは別のメッセージを出力するという違いがある。すなわち  $\text{Verify}(vk, M', \sigma^*) = 1$  を満たす  $(M', \sigma^*)$  に対し、 $vk^* \neq vk$ ,  $M^* \neq M'$ ,  $\text{Verify}(vk^*, M^*, \sigma^*) = 1$  のとき、 $\mathcal{A}$  が勝利すると定義する。 $\mathcal{A}$  の利得を  $\text{Adv}_{\mathcal{A}, \text{Sig}}^{\text{S-DEO}}(\lambda) = \Pr[\mathcal{A} \text{ wins}]$  と定義する。もし  $\text{Adv}_{\mathcal{A}, \text{Sig}}^{\text{S-DEO}}(\lambda)$  が全ての確率的多項式時間攻撃者  $\mathcal{A}$  に対して無視できるとき、署名方式  $\text{Sig}$  は S-DEO 安全であると定義する。

最後に M-S-UEO 安全性を定義する。これは Exclusive Ownership に関する安全性の中で最も強い安全性である。攻撃者  $\mathcal{A}$  (暗にセキュリティパラメータ  $\lambda$  と他のパラメータ  $((\text{KR})\text{-ECDSA}$  であれば  $(E, G, p, q)$  を入力とする) の出力を  $(M_1, M_2, \sigma, vk_1, vk_2)$  とする。

$$\text{Verify}(vk_1, M_1, \sigma) = 1 \wedge \text{Verify}(vk_2, M_2, \sigma) = 1 \wedge vk_1 \neq vk_2$$

のとき、 $\mathcal{A}$  が勝利すると定義する。 $\mathcal{A}$  の利得を  $\text{Adv}_{\mathcal{A}, \text{Sig}}^{\text{M-S-UEO}}(\lambda) = \Pr[\mathcal{A} \text{ wins}]$  と定義する。もし  $\text{Adv}_{\mathcal{A}, \text{Sig}}^{\text{M-S-UEO}}(\lambda)$  が全ての確率的多項式時間攻撃者  $\mathcal{A}$  に対して無視できるとき、署名方式  $\text{Sig}$  は M-S-UEO 安全であると定義する。

PS-2 変換では署名に  $H(vk)$  が含まれる。対象の署名  $\sigma^*$  が署名クエリにて生成されることを鑑みると、ハッシュ関数  $H$  の衝突困難性により別の署名鍵  $vk^*$  を攻撃者が生成できない。同様に  $H(vk_1) = H(vk_2)$  を満たす異なる検証鍵  $vk_1, vk_2$  を生成できない。これが PS-2 変換により M-S-UEO 安全となる背景である。

### 2.3 MBS

MBS は、ある署名に対し、検証に通る異なるメッセージ

と検証鍵のペアを作成できないことを保証する安全性である。もし MBS 安全ではない場合、攻撃者は署名が生成された後にメッセージと検証鍵を別のものに置き換えることが可能となる。ここで攻撃者  $\mathcal{A}$  は検証鍵  $vk$  を生成することが許されることに注意されたい。攻撃者  $\mathcal{A}$  (暗にセキュリティパラメータ  $\lambda$  と他のパラメータ ((KR-)ECDSA であれば  $(E, G, p, q)$  を入力とする) の出力を  $(M_1, M_2, \sigma, vk)$  とする。

$$\text{Verify}(vk, M_1, \sigma) = 1 \wedge \text{Verify}(vk, M_2, \sigma) = 1 \wedge M_1 \neq M_2$$

のとき、 $\mathcal{A}$  が勝利すると定義する。 $\mathcal{A}$  の利得を  $\text{Adv}_{\mathcal{A}, \text{Sig}}^{\text{MBS}}(\lambda) = \Pr[\mathcal{A} \text{ wins}]$  と定義する。もし  $\text{Adv}_{\mathcal{A}, \text{Sig}}^{\text{MBS}}(\lambda)$  が全ての確率的多項式時間攻撃者  $\mathcal{A}$  に対して無視できるとき、署名方式 Sig は MBS 安全であると定義する。

## 2.4 wNR

NR は、攻撃者が対象のメッセージは与えられない状況で、検証に通る署名と検証鍵のペアを生成できないことを保証する安全性である。例えば (KR-)ECDSA では  $h = H(M)$  を計算し、 $h$  を署名生成に使用する。このとき攻撃者は (ハッシュ関数の一方向性の意味で)  $M$  を  $h$  から得られない場合が想定される。Aulbach ら [1] により、追加情報を考慮しない弱い NR (weak NR, wNR) が導入された。 $\mathcal{A}_0$  と  $\mathcal{A}_1$  を攻撃者、 $\mathcal{C}$  をチャレンジャーとする。 $\mathcal{C}$  は  $(vk, sk) \leftarrow \text{KeyGen}(1^\lambda)$  を実行し、 $vk$  を  $\mathcal{A}_0$  に与える。 $\mathcal{A}_0$  は  $M^*$  を宣言、 $\mathcal{C}$  は  $\sigma \leftarrow \text{Sign}(sk, M^*)$  を計算し、 $(vk, \sigma)$  を  $\mathcal{A}_1$  に与える。<sup>\*4</sup> ここで  $\mathcal{A}_0$  と  $\mathcal{A}_1$  の間で情報をやり取りするステート情報がないため、 $\mathcal{A}_1$  は明示的に  $M^*$  を知ることはないことに注意されたい。 $\mathcal{A}_1$  は  $(vk^*, \sigma^*)$  を出力する。

$$\text{Verify}(vk^*, M^*, \sigma^*) = 1 \wedge vk \neq vk^*$$

のとき、 $(\mathcal{A}_0, \mathcal{A}_1)$  が勝利すると定義する。 $\mathcal{A}$  の利得を  $\text{Adv}_{(\mathcal{A}_0, \mathcal{A}_1), \text{Sig}}^{\text{wNR}}(\lambda) = \Pr[(\mathcal{A}_0, \mathcal{A}_1) \text{ wins}]$  と定義する。もし  $\text{Adv}_{(\mathcal{A}_0, \mathcal{A}_1), \text{Sig}}^{\text{wNR}}(\lambda)$  が全ての確率的多項式時間攻撃者  $\mathcal{A}$  に対して無視できるとき、署名方式 Sig は MBS 安全であると定義する。

## 3. KR-ECDSA

Ethereum Yellow Paper [37] に従い、 $B_{96..255}$  を入力のハッシュ値の下位 160 ビットとし、 $\text{addr} = B_{96..255}(H(vk))$  をアドレスとする。 $p$  と  $q$  を素数、 $H: \{0, 1\}^* \rightarrow \mathbb{Z}_q$  をハッシュ関数、 $E/\mathbb{F}_p: y^2 = x^3 + ax + b$  を  $\mathbb{F}_p$  上定義された位数  $q$  を持つ楕円曲線、 $G \in E(\mathbb{F}_p)$  をベースポイントする。以下、全てのアルゴリズムは暗に  $(E, G, p, q)$  を入力にとり仮定する。楕円曲線の点  $R \in E(\mathbb{F}_p)$  の座標を  $R = (R_x, R_y)$

で表記する。ここである  $R_x$  に対し  $R_y^2 = R_x^3 + aR_x + b$  を満たす  $y$  座標は 2 つ存在する。 $R_x$  から一意的に  $R$  を復元するため、KR-ECDSA ではフラグ  $v$  を導入する ( $R_y$  が  $q/2$  より大きいかな否かを指定)。すなわち、 $(R_x, v)$  から  $R = (R_x, R_y)$  が一意的に決定する。

**KeyGen( $1^\lambda$ ):**  $d \xleftarrow{\$} \mathbb{Z}_q$  を選び、 $P = dG$  を計算、 $vk = P$ ,  $sk = (d, P)$ ,  $\text{addr} = B_{96..255}(H(P))$  を出力する。

**Sign( $sk, M$ ):**  $sk = (d, P)$  とする。 $r \xleftarrow{\$} \mathbb{Z}_q$  を選び、 $h = H(M)$  と  $R = rG$  を計算する。ここで  $R = (R_x, R_y)$  とし、 $v$  をそのフラグとする。 $s = \frac{h+dR_x}{r} \bmod q$  を計算する。もし  $0 < s < q/2 + 1$  の場合、署名  $\sigma = (\text{addr}, s, R_x, v)$  を出力する (ここで  $\text{addr} = B_{96..255}(H(P))$ )。そうでない場合、 $s' = -s \bmod q$  とし、 $\bar{v}$  を  $v$  とは逆のフラグとして署名  $\sigma = (\text{addr}, s', R_x, \bar{v})$  を出力する。

**Verify( $\sigma, M$ ):**  $\sigma = (\text{addr}, s, R_x, v)$  とする。もし  $0 < s < q/2 + 1$  の場合、0 を出力する。 $(R_x, v)$  より  $R = (R_x, R_y)$  を計算、検証鍵  $P = \frac{s}{R_x}(R - \frac{h}{s}G)$  を復元する。 $\text{addr} = B_{96..255}(H(P))$  の場合 1 を、そうでない場合 0 を出力する。

オリジナルの ECDSA における検証アルゴリズムでは、以下の関係式により  $\frac{h}{s}G + \frac{R_x}{s}P$  の  $x$  座標が  $R_x$  の場合 1 を出力していた。

$$\frac{h}{s}G + \frac{R_x}{s}P = \frac{h}{s}G + \frac{R_x d}{s}G = \frac{h + R_x d}{s}G = rG = R$$

この関係式より

$$P = \frac{s}{R_x}(R - \frac{h}{s}G)$$

が成り立つ。

よく知られた事実として、 $\frac{h}{s}G + \frac{R_x}{s}P = R$  の場合、 $\frac{h}{-s}G + \frac{R_x}{-s}P = -R$  が成り立つ。ここで  $R$  と  $-R$  の  $x$  座標が同じであるため、 $s$  の範囲を確認しない場合は  $(\text{addr}, -s, R_x, \bar{v})$  も正当な署名として受理してしまう。KR-ECDSA では  $s$  の範囲を明示的に確認している。

## 4. KR-ECDSA の安全性評価

本章では、KR-ECDSA の安全性を評価する。まずフォーマットチェック付き ECDSA が強存在的偽造不可能であれば、KR-ECDSA も強存在的偽造不可であることは容易に示すことができる (詳細は ePrint 版 [16] を参照されたい)。

次に Exclusive Ownership 安全性について評価する。KR-ECDSA の検証アルゴリズムでは、署名とメッセージから復元された検証鍵のハッシュ値とアドレスとを比較するフェーズがある。このフェーズは PS-2 変換における署名に  $H(vk)$  を付加する処理と同様の効果があり、ハッシュ関数の衝突困難性により別の署名鍵を用いる余地がない (正確には Keccak-256 ハッシュ値の下位 160 ビットに対する衝突困難性を仮定する必要がある)。そのため、KR-ECDSA が

<sup>\*4</sup> Don ら [12] は NR の亜種として  $sk$  を攻撃者に与える安全性を導入している。本論文で提案する攻撃には  $sk$  は使用しないため、この定義の導入は見送った。

M-S-UEO 安全 (すなわち S-CEO かつ S-DEO 安全) であることも容易にわかる。

MBS についても同様にハッシュ関数の衝突困難性に帰着される。以下、MBS 攻撃者  $\mathcal{A}$  の出力を  $(M_1, M_2, (\text{addr}, s, R_x, v))$  とする。  $(R_x, v)$  より  $R = (R_x, R_y)$  が一意的に決定する。  $h_1 = H(M_1)$ ,  $h_2 = H(M_2)$ ,  $P_1 = \frac{s}{R_x}(R - \frac{h_1}{s}G)$ ,  $P_2 = \frac{s}{R_x}(R - \frac{h_2}{s}G)$  と置く。もし  $P_1 \neq P_2$  の場合,  $\text{addr} = \mathcal{B}_{96..255}(H(P_1)) = \mathcal{B}_{96..255}(H(P_2))$  により衝突困難性に反する。よって  $P_1 = P_2$  である。このとき  $h_1 = h_2$  が成り立つ。これは  $M_1 \neq M_2$  であるため、衝突困難性に反する。後述の ECDSA に対する MBS 攻撃では  $-R$  を用いているため、直感的にはフォーマットチェックの効果でこの攻撃を防いでいるように思える。しかし実際はフォーマットチェックなしでも KR-ECDSA は MBS を満たす。これは  $(R_x, v)$  より  $R = (R_x, R_y)$  が一意的に決定することから、暗に  $-R = (R_x, -R_y)$  の使用を除外しているためである。

最後に、KR-ECDSA は wNR 安全ではないことを示す。wNR 攻撃者  $(\mathcal{A}_0, \mathcal{A}_1)$  を以下で構成する。  $\mathcal{C}$  は  $\text{vk} = P$  を計算し、  $P$  を  $\mathcal{A}_0$  に与える。  $\mathcal{A}_0$  はメッセージ  $M$  を宣言する。ここで我々はメッセージに特別な構造を要求しない (ランダムなメッセージで十分である)。  $\mathcal{C}$  は  $M$  の署名  $\sigma = (\text{addr}, s, R_x, v)$  を計算し、  $\sigma$  を  $\mathcal{A}_1$  に送る。  $\mathcal{A}_1$  は  $0 < s^* < q/2 + 1$  を満たす  $s^* \in \mathbb{Z}_q$  を選び、  $r^* = 1/s^*$  と設定、  $R^* = r^*G$  を計算する。  $\mathcal{A}_1$  は暗に

$$d^* = \frac{1-h}{R_x^*}$$

と定義する (ここで  $R^* = (R_x^*, R_y^*)$ )。なお  $\mathcal{A}_1$  は  $h$  を直接知るわけではないものの、  $\frac{h}{s}G + \frac{R_x}{s}P = R$  かつ  $R$  は  $(R_x, v)$  から一意的に決定することから、  $hG = sR - R_xP$  は計算可能であることに注意されたい。  $\mathcal{A}_1$  は以下で  $P^*$  を計算する。

$$\begin{aligned} P^* &= \frac{1}{R_x^*}G - \frac{1}{R_x^*}(sR - R_xP) \\ &= \frac{1}{R_x^*}G - \frac{h}{R_x^*}G \\ &= \frac{1-h}{R_x^*}G \\ &= d^*G \end{aligned}$$

$\mathcal{A}_1$  は  $(\text{addr}^*, s^*, R_x^*, v^*)$  を出力する。ここで  $\text{addr}^* = \mathcal{B}_{96..255}(H(P^*))$  であり、フラグ  $v^*$  は  $R^* = (R_x^*, R_y^*)$  から一意的に決定する。  $0 < s^* < q/2 + 1$  であり、  $R^* = (R_x^*, R_y^*)$  が  $(R_x^*, v^*)$  から一意的に決定され、  $P^*$  が以下の計算により復元される。

$$\frac{s^*}{R_x^*}(R^* - \frac{h}{s^*}G) = \frac{1}{R_x^*} \frac{1}{r^*}(r^*G - r^*hG) = \frac{1-h}{R_x^*}G = P^*$$

よって  $\text{addr}^* = \mathcal{B}_{96..255}(\frac{s^*}{R_x^*}(R^* - \frac{h}{s^*}G))$  が成り立つことから、検証アルゴリズムは 1 を出力する。ここで  $M$  も  $d$  もラ

ンダムに選ばれていることから、  $H(M) = 1 - dR_x^*$  が成り立つ確率は無視できる。よって圧倒的な確率で  $P \neq P^*$  が成り立つ。

## 5. Ethereum における wNR 安全性の意義

Ethereum における実際の用途では、トランザクション  $M$  に対し KR-ECDSA 署名  $\sigma$  が作成される。ここで攻撃者が  $M$  を知らずに  $\sigma$  のみを知るとことは考えづらい。より正確には、あるアドレス  $\text{addr}$  に対し、  $\text{sk} = (d, P)$  を持つアドレスの所持者がトランザクション  $M$  に対し署名  $\sigma$  を生成する。  $\sigma$  を検証する際、検証者は  $\text{addr}$  と  $M$  の両方を知ることとなる。このとき検証者は  $\sigma$  と  $M$  から  $P$  を復元し、  $\text{addr} = \mathcal{B}_{96..255}(H(P))$  を確認する。この使用形態において検証者は常にトランザクション  $M$  を知ることとなる。なお  $M$  が既知であれば、任意の署名鍵を生成し  $M$  に署名することで容易に新たな署名を生成することが可能である。さらに我々の wNR 攻撃においては攻撃者がアドレス  $\text{addr}$  を生成しているが、  $\text{addr}$  はブロックチェーンに保存されていることを鑑みるとこの設定が有効とは言えない。以上により、Ethereum におけるトランザクション検証の用途では wNR 安全性の効果は期待されないと、本論文では主張する。なお他の用途 [12, 23] における NR 安全性の意義について疑問を呈したわけではないことを強調する。

## 6. ECDSA の BUFF 安全性評価

KR-ECDSA の検証アルゴリズムにおける  $\text{addr} = \mathcal{B}_{96..255}(H(P))$  の確認が BUFF 安全性を達成する本質であることを確認するため、オリジナルの ECDSA に対する BUFF 攻撃を記述する。なおベースポイント  $G$  を楕円曲線上の別の点に置き換える domain parameter shifting 攻撃 [36] とは異なり、  $G$  は変更しないことに注意されたい。  $\sigma = (s, R_x)$  をメッセージ  $M$  に対する ECDSA 署名、  $P = dG$  を検証鍵とする。すなわち  $R = (R_x, R_y)$  に対し  $\frac{h}{s}G + \frac{R_x}{s}P = R$  が成り立つ。以下、S-CEO 攻撃を提案する。  $P^* = \frac{-2h}{R_x}G - P$  を計算する。このとき

$$\begin{aligned} \frac{h}{s}G + \frac{R_x}{s}P^* &= \frac{h}{s}G + \frac{R_x}{s}(\frac{-2h}{R_x}G - P) \\ &= \frac{h}{s}G - \frac{2h}{s}G - \frac{R_x}{s}P \\ &= -(\frac{h}{s}G + \frac{R_x}{s}P) \\ &= -R \end{aligned}$$

が成り立つ。よって  $\sigma = (s, R_x)$  は  $M$  に対する  $P^*$  の基での正当な署名である。ここで  $d \neq \frac{h}{R_x}$  が圧倒的な確率で成り立つことから、  $P \neq P^*$  である。

次に S-DEO 攻撃を提案する。S-DEO 攻撃者は任意の  $M^* \neq M$  を選び、  $h^* = H(M^*)$  と  $\frac{h}{s}G + \frac{R_x}{s}P = R$  を計算、  $P^* = \frac{s}{R_x}(R - \frac{h^*}{s}G)$  と設定する。このとき

$$\begin{aligned}\frac{h^*}{s}G + \frac{R_x}{s}P^* &= \frac{h^*}{s}G + \frac{R_x}{s}\left(\frac{s}{R_x}(R - \frac{h^*}{s}G)\right) \\ &= R\end{aligned}$$

が成り立つ。よって  $\sigma = (s, R_x)$  は  $M^*$  に対する  $P^*$  の基での正当な署名である。なお  $P = P^*$  と  $h = h^*$  は等価であり、 $M \neq M^*$  からハッシュ関数の衝突困難性により  $h \neq h^*$  が圧倒的な確率で成り立つ。

MBS について。Cremers ら [7] が言及した通り、本質的には Stern ら (Section 4. Duplicates in ECDSA [32]) と Vaudenay (Section 2.1. Signature Manipulation in ECDSA [35]) による攻撃と同じである。異なる  $M_1$  と  $M_2$  を選ぶ。  $h_1 = H(M_1)$ ,  $h_2 = H(M_2)$  とする。  $r \xleftarrow{\$} \mathbb{Z}_q$  を選び、  $R = rG$  を計算する。  $R = (R_x, R_y)$  と記述する。  $d = -\frac{h_1+h_2}{2R_x}$  と定義し、  $P = dG$  を計算する。  $s = \frac{h_1-h_2}{2r}$  を計算する。  $(M_1, M_2, (s, R_x), P)$  を出力する。攻撃者の出力  $(M_1, M_2, (s, R_x), P)$  に対し、  $\frac{h_1}{s}G + \frac{R_x}{s}P = \frac{h_1+dR_x}{s}G = rG = (R_x, R_y)$  かつ  $\frac{h_2}{s}G + \frac{R_x}{s}P = -rG = (R_x, -R_y)$  が成り立つ。

## 7. Aumayr らの ECDSA ベース Adaptor 署名に対する S-DEO 攻撃

Aumayr ら [2] はフォーマットチェック付き ECDSA (彼らは Positive ECDSA と呼んでいる) を基にした Adaptor 署名方式を提案している。このフォーマットチェックにより、検証アルゴリズムで  $-R$  が得られる場合を排除することができ、前章における S-CEO 攻撃や MBS 攻撃は回避できる (S-CEO 安全, MBS 安全であるという主張ではないことに注意されたい)。一方、S-DEO 攻撃では検証アルゴリズムにて  $R$  を計算するため、フォーマットチェックの効果が無い。そのため、事前署名を入手した攻撃者が検証に通る別のメッセージ、別の検証鍵を生成できる余地がある。なお現在 Adaptor 署名の応用 (atomic swap, payment channels [29], private coin mixing [28, 31], oracle-based payments [28]) における S-DEO 攻撃の影響は見つかっていないため、理論的な結果と言える。しかし多くの暗号資産の取引が行われる以上、潜在的な脆弱性を明らかにすることには一定の意義があると考えられる。(後述する Blind Adaptor 署名を含め) Adaptor 署名に対する S-DEO 攻撃 (その他の BUFF 安全性) のインパクトの精査は今後の課題とする。

Rel を hard relation とし、ステートメント  $Y$  と証拠 (witness)  $y$  に対し  $(Y, y) \in \text{Rel}$  と表記する。署名方式  $\text{Sig} = (\text{KeyGen}, \text{Sign}, \text{Verify})$  と hard relation Rel に関する Adaptor 署名  $\text{AS} = (\text{pSign}, \text{pVerify}, \text{Adapt}, \text{Ext})$  は以下で定義される。  $(\text{vk}, \text{sk}) \leftarrow \text{KeyGen}(1^\lambda)$ ,  $\sigma \leftarrow \text{Sign}(\text{sk}, M)$  とする。 Dai ら [9] により定義された Unlinkability では、事前署名を Adapt して得られる署名と Sign にて生成した署名とが識別不可能であることを保証する。現状最も強い安全性

定義を与えている Gerhart ら [20] も Unlinkability を採用している。 Liu ら [27] は Witness Hiding 安全性を導入しているが、これは Unlinkability に含まれる。 Ciampi ら [5] は Liu らの安全性定義を採用している。この状況を鑑みると、Unlinkability は自然な定義であると考えられる。そのため Adapt アルゴリズムの出力である Adaptor 署名として、Sign で生成された署名と同じ記号  $\sigma$  を用いることとする。

### Adaptor 署名

$\text{pSign}(\text{sk}, M, Y)$ : 事前署名生成アルゴリズムは署名鍵  $\text{sk}$ , メッセージ  $M$ , ステートメント  $Y$  を入力とし、事前署名  $\tilde{\sigma}$  を出力する。

$\text{pVerify}(\text{vk}, M, \tilde{\sigma}, Y)$ : 事前署名検証アルゴリズムは検証鍵  $\text{vk}$ ,  $M$ ,  $\tilde{\sigma}$ ,  $Y$  を入力とし、0 または 1 を出力する。

$\text{Adapt}(\text{vk}, M, \tilde{\sigma}, y)$ : アダプトアルゴリズムは  $\text{vk}$ ,  $M$ ,  $\tilde{\sigma}$ , 証拠  $y$  を入力とし、Adaptor 署名  $\sigma$  を出力する。

$\text{Ext}(\text{vk}, M, Y, \tilde{\sigma}, \sigma)$ : 抽出アルゴリズムは  $\text{vk}$ ,  $M$ ,  $Y$ ,  $\tilde{\sigma}$ ,  $\sigma$  を入力とし、 $y$  または  $\perp$  を出力する。

**Adaptor 署名における S-DEO 安全性** 次に Adaptor 署名における S-DEO 安全性を定義する。なおこの定義は (攻撃者が事前署名クエリしない、メッセージをチャレンジャーが選ぶという観点から) 非常に弱い安全性であることに注意されたい (なお我々の攻撃には十分である)。この安全性定義は、元々の Adaptor 署名の安全性定義には含まれていないことを強調する。チャレンジャー  $\mathcal{C}$  は  $(\text{vk}, \text{sk}) \leftarrow \text{KeyGen}(1^\lambda)$  を生成し、 $\text{vk}$  を攻撃者  $\mathcal{A}$  に与える。  $\mathcal{C}$  は  $M$  と  $(Y, y) \in \text{Rel}$  を選び、事前署名  $\tilde{\sigma} \leftarrow \text{pSign}(\text{sk}, M, Y)$  を計算、  $(M, Y, \tilde{\sigma})$  を  $\mathcal{A}$  に与える。  $\mathcal{A}$  は  $\text{vk}^* \neq \text{vk}$ ,  $M^* \neq M$ ,  $\text{pVerify}(\text{vk}^*, M^*, \tilde{\sigma}, Y) = 1$  を満たす  $(\text{vk}^*, M^*)$  を出力する。

Aumayr らは楕円曲線に関連した以下の 3 つの hard relation を導入している。  $\text{Rel}_G : \mathbb{G} \times \mathbb{Z}_q$  を

$$\text{Rel}_G := \{(Y, y) : Y = yG\}$$

$\text{Rel}'_G$  を

$$\text{Rel}'_G := \{((Y, \pi_Y), y) : (Y, y) \in \text{Rel}_G \wedge V_G(Y, \pi_Y) = 1\}$$

$\text{Rel}_{(Y, G)} : \mathbb{G}^2 \times \mathbb{Z}_q$  を

$$\text{Rel}_{(Y, G)} := \{((\tilde{R}, R), r) : \tilde{R} = rY \wedge R = rG\}$$

と定義する。ここで  $\pi_Y$  は  $Y = yG$  を満たす  $y$  に対する非対話ゼロ知識証明、  $(P_G, V_G)$  及び  $(P_{(P, G)}, V_{(P, G)})$  は証明システムである。 Aumayr らは関数  $f$  を導入しているが、本論文では  $f$  として楕円曲線の点に対しその  $x$  座標を返す処理を明示的に記述する。なお証拠  $y$  と楕円曲線上の点  $R$  の  $y$  座標  $R_y$  とが混在するが、文脈により判断されたい。 Aumayr らの方式では、  $\text{pSign}$  アルゴリズムの入力にインスタンス  $Y$  そのものに加え、非対話ゼロ知識証明  $\pi_Y$  も入力し、最終的に抽出された証拠  $y$  が関係  $\text{Rel}'_G$  に含まれるかを

確認しているという特徴がある。

#### Aumayr らの ECDSA ベース Adaptor 署名方式.

$\text{pSign}(\text{sk}, M, (Y, \pi_Y))$ :  $\text{sk} = d$  とする.  $r \xleftarrow{\$} \mathbb{Z}_q$  を選び,  $h = H(M)$ ,  $R = rG$ ,  $\tilde{R} = rY$  を計算する. ここで  $\tilde{R} = (\tilde{R}_x, \tilde{R}_y)$  と表記する.  $\tilde{s} = \frac{h+d\tilde{R}_x}{r} \bmod q$  を計算する. もし  $\tilde{s} > q/2+1$  であれば,  $\tilde{s} := -\tilde{s} \bmod q$  と設定する.  $\pi \leftarrow \text{P}_{(Y,G)}((\tilde{R}, R), r)$  を計算し,  $\tilde{\sigma} = (\tilde{s}, \tilde{R}_x, \tilde{R}, \pi)$  を出力する.

$\text{pVerify}(\text{vk}, M, \tilde{\sigma}, (Y, \pi_Y))$ :  $\text{vk} = P$  とする. もし  $\text{V}_G(Y, \pi_Y) = 0$  であれば 0 を出力する. もし  $\tilde{s} \leq q/2+1$  が成り立たない場合, 0 を出力する.  $h = H(M)$  と  $R = \frac{h}{\tilde{s}}G + \frac{\tilde{R}_x}{\tilde{s}}P$  を計算する. もし  $\text{V}_{(Y,G)}((\tilde{R}, R), \pi) = 0$  の場合, 0 を出力する.  $(\tilde{R})_x = \tilde{R}_x$  の場合 1 を, そうでない場合 0 を出力する.

$\text{Adapt}(\text{vk}, M, \tilde{\sigma}, y)$ :  $\tilde{\sigma} = (\tilde{s}, \tilde{R}_x, \tilde{R}, \pi)$  とする.  $s = \tilde{s}y^{-1} \bmod q$  を計算し,  $\sigma = (s, \tilde{R}_x)$  を出力する.

$\text{Ext}(\text{vk}, M, (Y, \pi_Y), \tilde{\sigma}, \sigma)$ :  $\tilde{\sigma} = (\tilde{s}, \tilde{R}_x, \tilde{R}, \pi)$ ,  $\sigma = (s, \tilde{R}_x)$  とする.  $y = s^{-1}\tilde{s} \bmod q$  を計算する. もし  $((Y, \pi_Y), y) \in \text{Rel}'_G$  の場合  $y$  を出力, そうでない場合  $\perp$  を出力する.

**S-DEO 攻撃:** 我々の S-DEO 攻撃を以下に示す.  $C$  は  $d \in \mathbb{Z}_q$  を選び,  $P = dG$  と  $(Y, \pi_Y)$  を計算する.  $C$  は  $M$  を選び,  $M$  に対する事前署名  $\tilde{\sigma} = (\tilde{s}, \tilde{R}_x, \tilde{R}, \pi)$  を計算,  $(M, (Y, \pi_Y), \tilde{\sigma})$  を  $\mathcal{A}$  に送る.  $\mathcal{A}$  は  $M^* \neq M$  を選び  $h^* = H(M^*)$ ,  $R = \frac{h^*}{\tilde{s}}G + \frac{\tilde{R}_x}{\tilde{s}}P$ ,

$$P^* := \frac{\tilde{s}}{\tilde{R}_x}(R - \frac{h^*}{\tilde{s}}G)$$

を計算する. ここで  $\text{V}_G(Y, \pi_Y) = 1$ ,  $0 < \tilde{s} < q/2 + 1$ ,  $\text{V}_{(Y,G)}((\tilde{R}, R), \pi) = 1$ ,

$$\frac{h^*}{\tilde{s}}G + \frac{\tilde{R}_x}{\tilde{s}}P^* = \frac{h^*}{\tilde{s}}G + \frac{\tilde{R}_x}{\tilde{s}}(\frac{\tilde{s}}{\tilde{R}_x}(R - \frac{h^*}{\tilde{s}}G)) = R$$

より,  $\text{pVrfy}(P^*, M^*, \tilde{\sigma}, (Y, \pi_Y)) = 1$  が成り立つ.  $\mathcal{A}$  は  $(P^*, M^*)$  を出力, S-DEO 安全性を破る.

Aumayr らの ECDSA ベース Adaptor 署名方式の拡張として, Tu ら [34] は 2 つの ECDSA ベース Adaptor 署名方式を提案している. 効率化を目的としているが, 他の検証鍵を用いる余地を排除している構成となっている. そのため, 上記 S-DEO 攻撃に対して耐性を持つ (S-DEO 安全であるという主張ではない).

**Blind Adaptor 署名:** BlindHub の構成要素として, Qin ら [31] は Aumayr らの ECDSA ベース Adaptor 署名方式を基に Blind Adaptor 署名を提案した. ユーザと署名者間の対話型署名プロトコルを除き元の Adaptor 署名と同じである. そのため, S-DEO 攻撃がそのまま適用できる (攻撃者がメッセージ  $M$  を選ぶ点を除く). 彼らの安全性モデル外の攻撃であることを再度強調しておく.

#### Qin らの Blind Adaptor 署名方式 (対話型署名部分)

**ユーザ.**  $h = H(M)$  を計算する. さらにハッシュ関数の入力に対する非対話ゼロ知識証明  $\pi_h$  を計算,  $(h, \pi_h)$  を (検証鍵  $P$  を管理する) 署名者に送る.

**署名者.** もし  $\pi_h$  が正しい証明ではない場合, 停止する. 以後, 事前署名アルゴリズムを実行する:  $r \xleftarrow{\$} \mathbb{Z}_q$  を選び,  $R = rG$ ,  $\tilde{R} = rY$  を計算する. ここで  $\tilde{R} = (\tilde{R}_x, \tilde{R}_y)$  と記述する.  $\tilde{s} = \frac{h+d\tilde{R}_x}{r} \bmod q$  を計算する. もし  $\tilde{s} > q/2 + 1$  の場合,  $\tilde{s} := -\tilde{s} \bmod q$  と設定する.  $\pi \leftarrow \text{P}_{(Y,G)}((\tilde{R}, R), r)$  を計算し,  $\tilde{\sigma} = (\tilde{s}, \tilde{R}_x, \tilde{R}, \pi)$  をユーザに送付する.

**ユーザ.**  $\tilde{s} = 0$  または  $\text{pVerify}(P, M, \tilde{\sigma}, (Y, \pi_Y)) = 0$  の場合, 停止する.

$\tilde{\sigma} = (\tilde{s}, \tilde{R}_x, \tilde{R}, \pi)$  は元の Adaptor 署名と同じであるため, ユーザは  $\text{pVerify}(P^*, M^*, \tilde{\sigma}, (Y, \pi_Y)) = 1$ ,  $P^* \neq P$ ,  $M^* \neq M$  を満たす  $(P^*, M^*)$  を出力することができる. なお Adaptor 署名に対する攻撃とは異なり, (Blind 署名のシンタックスに起因して) 攻撃者 (ユーザ) がメッセージ  $M$  を選ぶ設定となっていることに注意されたい.

## 8. 結論と今後の課題

本論文では, KR-ECDSA の BUFF 安全性について調査した. さらに KR-ECDSA のどの部分が BUFF 安全性の達成に起因しているのかを明確にするため, 通常の ECDSA に対する BUFF 攻撃を提案した. その応用として, Aumayr らの ECDSA ベース Adaptor 署名, Qin らの ECDSA ベース Blind Adaptor 署名に対する S-DEO 攻撃を与えた.

しきい値 ECDSA [8, 10, 11, 19, 26, 38] に関して. Fischlin ら [18] により, しきい値署名に対し BUFF 安全性を付加する変換手法が提案されている. この変換を通せば BUFF 安全なしきい値 ECDSA が得られる. Fischlin らによる評価, 今回の (Blind) Adaptor 署名に関する評価のように高機能署名に対する BUFF 安全性評価が成され始めている現状を受け, 他の高機能署名方式に対する BUFF 安全性の定義や意義の精査, さらに Adaptor 署名に対して BUFF 安全性を付加する変換手法の提案を今後の課題とする. また Struck と Weishäupl [33] により BUFF 安全性が拡張されている. これら安全性の KR-ECDSA における精査も今後の課題とする.

**謝辞.** 本研究は JSPS 科研費 JP21K11897, JP25H01106 の助成を受けたものです.

#### 参考文献

- [1] Thomas Aulbach, Samed Düzli, Michael Meyer, Patrick Struck, and Maximiliane Weishäupl. Hash your keys before signing: BUFF security of the additional NIST PQC signatures. In *Post-Quantum Cryptography*, pages 301–335, 2024.
- [2] Lukas Aumayr, Oguzhan Ersoy, Andreas Erwig, Sebastian Faust, Kristina Hostáková, Matteo Maffei, Pedro

- Moreno-Sanchez, and Siavash Riahi. Generalized channels from limited blockchain scripts and adaptor signatures. In *ASIACRYPT*, pages 635–664, 2021.
- [3] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. In *CHES*, pages 124–142, 2011.
  - [4] Jacqueline Brendel, Cas Cremers, Dennis Jackson, and Mang Zhao. The provable security of Ed25519: Theory and practice. In *IEEE S&P*, pages 1659–1676, 2021.
  - [5] Michele Ciampi, Xiangyu Liu, Ioannis Tzannetos, and Vassilis Zikas. Universal adaptor signatures from black-box multi-party computation. In *CT-RSA*, pages 375–398, 2025.
  - [6] Cas Cremers, Samed Düzlül, Rune Fiedler, Marc Fischlin, and Christian Janson. BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures. *IACR Cryptology ePrint Archive*, page 1525, 2020. Version 1.4.1, October 2023.
  - [7] Cas Cremers, Samed Düzlül, Rune Fiedler, Marc Fischlin, and Christian Janson. BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures. In *IEEE S&P*, pages 1696–1714, 2021.
  - [8] Handong Cui, Kwan Yin Chan, Tsz Hon Yuen, Xin Kang, and Cheng-Kang Chu. Bandwidth-efficient zero-knowledge proofs for threshold ECDSA. *The Computer Journal*, 67(4):1265–1278, 2024.
  - [9] Wei Dai, Tatsuki Okamoto, and Go Yamamoto. Stronger security and generic constructions for adaptor signatures. In *INDOCRYPT*, pages 52–77, 2022.
  - [10] Jack Doerner, Yashvanth Kondi, Eysa Lee, and Abhi Shelat. Threshold ECDSA from ECDSA assumptions: The multiparty case. In *IEEE S&P*, pages 1051–1066, 2019.
  - [11] Jack Doerner, Yashvanth Kondi, Eysa Lee, and Abhi Shelat. Threshold ECDSA in three rounds. In *IEEE S&P*, pages 3053–3071, 2024.
  - [12] Jelle Don, Serge Fehr, Yu-Hsuan Huang, Jyun-Jie Liao, and Patrick Struck. Hide-and-seek and the non-resignability of the BUFF transform. In *TCC*, pages 347–370, 2024.
  - [13] Jelle Don, Serge Fehr, Yu-Hsuan Huang, and Patrick Struck. On the (in)security of the BUFF transform. In *CRYPTO*, pages 246–275, 2024.
  - [14] Samed Düzlül, Rune Fiedler, and Marc Fischlin. Buffing FALCON without increasing the signature size. In *Selected Areas in Cryptography*, pages 131–150, 2024.
  - [15] Samed Düzlül and Patrick Struck. The role of message-bound signatures for the beyond unforgeability features and weak keys. In *ISC*, pages 61–80, 2024.
  - [16] Keita Emura. On the BUFF security of ECDSA with key recovery. *IACR Cryptol. ePrint Arch.*, page 2018, 2024. <https://eprint.iacr.org/2024/2018>.
  - [17] Manuel Ferschi, Eike Kiltz, and Bertram Poettering. On the provable security of (EC)DSA signatures. In *ACM CCS*, 2016.
  - [18] Marc Fischlin, Aikaterini Mitrokotsa, and Jenit Tomy. BUFFing threshold signature schemes. In *Public-Key Cryptography*, pages 137–168, 2025.
  - [19] Rosario Gennaro, Steven Goldfeder, and Arvind Narayanan. Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security. In *ACNS*, pages 156–174, 2016.
  - [20] Paul Gerhart, Dominique Schröder, Pratik Soni, and Sri Aravinda Krishnan Thyagarajan. Foundations of adaptor signatures. In *EUROCRYPT*, pages 161–189, 2024.
  - [21] Jens Groth and Victor Shoup. On the security of ECDSA with additive key derivation and presignatures. In *EUROCRYPT*, pages 365–396, 2022.
  - [22] Dominik Hartmann and Eike Kiltz. Limits in the provable security of ECDSA signatures. In *TCC*, pages 279–309, 2023.
  - [23] Dennis Jackson, Cas Cremers, Katriel Cohn-Gordon, and Ralf Sasse. Seems legit: Automated analysis of subtle attacks on protocols that use signatures. In *ACM CCS*, pages 2165–2180, 2019.
  - [24] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ECDSA). <https://web.archive.org/web/20170921160141/http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa-cert.pdf>.
  - [25] Mukul Kulkarni and Keita Xagawa. Strong existential unforgeability and more of MPC-in-the-head signatures. *IACR Cryptol. ePrint Arch.*, page 1069, 2024. <https://eprint.iacr.org/2024/1069>.
  - [26] Yehuda Lindell. Fast secure two-party ECDSA signing. In *CRYPTO*, pages 613–644, 2017.
  - [27] Xiangyu Liu, Ioannis Tzannetos, and Vassilis Zikas. Adaptor signatures: New security definition and a generic construction for NP relations. In *ASIACRYPT*, pages 168–193, 2024.
  - [28] Varun Madathil, Sri Aravinda Krishnan Thyagarajan, Dimitrios Vasilopoulos, Lloyd Fournier, Giulio Malavolta, and Pedro Moreno-Sanchez. Cryptographic oracle-based conditional payments. In *NDSS*. The Internet Society, 2023.
  - [29] Andrew Miller, Iddo Bentov, Surya Bakshi, Ranjit Kumareshan, and Patrick McCorry. Sprites and state channels: Payment networks that go faster than lightning. In *Financial Cryptography and Data Security*, pages 508–526, 2019.
  - [30] Thomas Pornin and Julien P. Stern. Digital signatures do not guarantee exclusive ownership. In *ACNS*, pages 138–150, 2005.
  - [31] Xianrui Qin, Shimin Pan, Arash Mirzaei, Zhimei Sui, Oguzhan Ersoy, Amin Sakzad, Muhammed F. Esgin, Joseph K. Liu, Jiangshan Yu, and Tsz Hon Yuen. Blind-Hub: Bitcoin-compatible privacy-preserving payment channel hubs supporting variable amounts. In *IEEE S&P*, pages 2462–2480, 2023.
  - [32] Jacques Stern, David Pointcheval, John Malone-Lee, and Nigel P. Smart. Flaws in applying proof methodologies to signature schemes. In *CRYPTO*, pages 93–110, 2002.
  - [33] Patrick Struck and Maximiliane Weishäupl. A framework for advanced signature notions. *IACR Cryptol. ePrint Arch.*, page 960, 2025. <https://eprint.iacr.org/2025/960>.
  - [34] Binbin Tu, Min Zhang, and Chen Yu. Efficient ECDSA-based adaptor signature for batched atomic swaps. In *ISC*, pages 175–193, 2022.
  - [35] Serge Vaudenay. The security of DSA and ECDSA. In *Public Key Cryptography*, pages 309–323, 2003.
  - [36] Serge Vaudenay. Digital signature schemes with domain parameters: Yet another parameter issue in ECDSA. In *ACISP*, pages 188–199, 2004.
  - [37] Gavin Wood. Ethereum yellow paper (Shanghai version 9fde3f4-2024-09-02), 2024. <https://ethereum.github.io/yellowpaper/paper.pdf>.
  - [38] Guy Zyskind, Avishay Yanai, and Alex Pentland. Unstoppable wallets: Chain-assisted threshold ECDSA and its applications. In *ASIACCS*. ACM, 2024.