BUFFing Threshold Signature Schemes

Marc Fischlin[1], Aikaterini Mitrokotsa[2], and Jenit Tomy[2]

[1]Cryptoplexity, Technische Universität Darmstadt, Germany,
marc.fischlin@tu-darmstadt.de
[2]University of St. Gallen, Switzerland,
{katerina.mitrokotsa,jenit.tomy}@unisg.ch

March 4, 2025

Abstract

We explore advanced security notions for threshold signature schemes, focusing on Beyond UnForgeability Features (BUFF), introduced by Cremers et al. (S&P'21) in the non-threshold setting. The BUFF properties protect against attacks based on maliciously chosen keys, e.g., expropriating a message-signature pair under a new public key (called exclusive ownership). We first formalize these notions in the threshold setting and examine their relationships. Notably, unlike regular signature schemes, the hierarchy of variants of exclusive ownership notions only holds for threshold schemes if they are also robust. We then present a generic compiler that transforms any threshold signature scheme to satisfy exclusive ownership, and message-bound signature properties with minimal overhead. Furthermore, we modify the threshold BLS signature scheme to achieve these additional properties without increasing the signature size. Lastly, we identify specific structures in threshold signature schemes where BUFF properties can be naturally extended from the underlying standard signature scheme, and we analyze and prove the security properties in some of the existing threshold schemes.

Keywords Threshold signatures, Beyond UnForgeability Features, exclude ownership, message-bound signatures

1 Introduction

Digital signatures are an important cryptographic primitive to guarantee message integrity and authenticity of origin. They allow a signer to sign a message m using her secret key sk and generate a signature σ, while everyone having the corresponding public key pk can verify the validity of the signature. The main security requirement for a digital signature is the resistance to existential unforgeability under chosen-message attacks ($\mathsf{UF\text{-}CMA}$) [24], which implies that an adversary who gets access to several message-signature pairs cannot generate a new valid signature-message pair.

Although the $\mathsf{UF\text{-}CMA}$ security notion was until recently used as an omnipotent security guarantee for digital signatures, it has been shown that it only covers digital security guarantees under the assumption that the signer's secret key is honestly generated, ignoring attacks in which an adversary can choose related keys maliciously. Early on, such maliciously chosen keys have been identified as a threat to key exchange protocols [5, 29] and certification schemes [31]. Recently, Jackson et al. [25] pointed out the importance of stronger security guarantees of digital signature schemes in other settings as well.

1.1 Beyond-Unforgeability Features of Signature Schemes

To address malicious-key attacks against digital signature schemes new security properties were introduced under the term *b*eyond-*unf*orgeability *f*eatures (BUFF) [31, 25, 10, 11]. This includes *exclusive ownership*, *message-bound signatures* and *non-resignability*. Testifying their significance, these properties have also been stated as an extra desirable security goal in NIST's recent call for additional post-quantum signature schemes [30].

In more detail, *exclusive ownership* considers an adversary who has access to a message and a signature pair (m, σ) associated with a public key pk and is asked to find a new public key pk^* which also verifies the pair (m, σ). This security notion has two variations: the *conservative exclusive ownership* (CEO), which is the one described above, and the *destructive exclusive ownership* (DEO), which requires the adversary to find a different message i.e., (m^*, σ) where $m^* \neq m$ such that the pair verifies under pk^*. *Message bound signatures* (MBS) consider an adversary that should find two messages m and m' with $m \neq m'$ and a signature σ s.t. both (m, σ) and (m', σ) are valid under pk. The main difference between MBS and $\mathsf{UF\text{-}CMA}$ is that pk can be generated arbitrarily and not necessarily be the output of the key generation algorithm. *Non-resignability* (NR) considers that an adversary with a signature σ corresponding to an unknown message m and a public key pk, is asked to provide a different public key pk^* and signature σ^* such that σ^* verifies correctly for the unknown message m under the key pk^*.

Cremers et al. [11] studied the BUFF security properties of all six round-3 candidate signature schemes of the NIST PQC standardization process. This includes the selected candidates Dilithium, Falcon, and SPHINCS. Besides studying BUFF security of individual schemes, Cremers et al. [11] discuss general transformations for signature schemes to achieve these extra security features. One is the general BUFF transformation, guaranteeing all properties by signing a hash $\mathsf{H}(\mathsf{pk}, m)$ of the public key pk and the message m (instead of the message) and appending this hash value to the signature. The BUFF-lite transformation only appends this hash value and achieves all properties except for non-resignability. Pornin and Stern [31] also proposed some transformations to achieve (weak) versions of exclusive ownership.

Aulbach et al. [1] studied the BUFF security of the additional signatures candidates for the NIST standardization process, based on codes, isogenies, lattices, multi-variate equations. They consider a weaker variation of the exclusive ownership notion, namely, S-UEO, that implies exclusive ownership for honestly generated key pairs and signatures. Furthermore, they introduced a weaker notion of non-resignability called wNR. They also argue that using one of the transformations in [31], usually denoted PS-3, already provides BUFF security for some schemes. A similar result has been established for the FALCON signature scheme by Düzlü et al. [22].

1.2 BUFFing Threshold Signature Schemes

In threshold signature schemes, the private key is distributed among a set of participants, and it is required that a quorum must cooperate to issue a signature. Signatures can be verified by a single public key. While the idea of threshold cryptography has been around for a while now [17, 18], the advances in cryptocurrencies and smart contracts encourage the design especially of threshold signature schemes, since they can be used by a set of signers to authenticate transactions or to sign a network consensus.

Although constructions of threshold signatures have received significant attention in the literature recently, e.g., [27, 3, 15, 34, 12, 13, 19, 2, 23, 26, 9], to our knowledge no results have been provided about whether the BUFF properties are satisfied by threshold signature schemes. This is a natural question, considering that threshold signature schemes are meant to substitute regular signature schemes in settings where the trust should be distributed, but are in principle also susceptible to such attacks. This is more true in decentralized finance, where identities often correspond to public keys.

In this paper, we therefore study the following question:

> *Do threshold signature schemes satisfy the BUFF properties, and, if not, is there a general way to transform threshold schemes to achieve BUFF properties?*

1.3 Our Contributions

We answer the above question in the affirmative in the following sense: We can show that threshold schemes that are based on BUFFed signature schemes inherit their security for the exclusive ownership and message-binding properties. If this is not the case, we can apply a general transform to BUFF threshold schemes. For specific cases like the threshold BLS scheme we can give more direct constructions.

We note that, analogously to the BUFF-lite transformation [11], we only investigate here ownership notions and message binding, neglecting the non-resignability property. Indeed, this property is concerned with signatures for unknown messages, which, in particular, means that in the threshold setting, no malicious party could participate in the signature generation process. As such the non-resignability property would relapse to the question regarding honestly generated final signatures. In this context, we remark that Don et al. [21] noted that the original non-resignability notion in [11] cannot be achieved assuming that auxiliary data about the unknown message is available. Düzlü et al. [22] and Don et al. [20] subsequently provided different definitions for non-resignability.

In more detail, our main contributions are as follows:

BUFF properties in the threshold setting We formalize beyond-unforgeability security notions for threshold signature schemes and provide the relationships between the properties. We focus here on the case of non-interactive schemes, as for some properties like message binding, the round complexity of the signature generation step is irrelevant; for other properties, the approach could be generalized to interactive settings. Remarkably, while for regular signature schemes, the strongest version of malicious-strong universal exclusive ownership $\mathsf{M\text{-}S\text{-}UEO}$ [10] implies other variants like $\mathsf{S\text{-}UEO}$, $\mathsf{S\text{-}CEO}$, and $\mathsf{S\text{-}DEO}$, this does not hold in general in the threshold setting. In this scenario, we can resurrect the implication if the threshold signature scheme is also *robust* [6]. Robustness is a common property of threshold signature schemes and says that an adversary cannot make a joint signature generation fail if sufficiently many valid partial signatures are provided. We also discuss that robustness is necessary for the implication from $\mathsf{M\text{-}S\text{-}UEO}$ to $\mathsf{S\text{-}UEO}$ to hold.

Generic Compiler We provide a generic compiler that takes as input any threshold signature scheme satisfying unforgeability and robustness and outputs a threshold signature scheme that satisfies unforgeability, robustness, exclusive ownership, and message-bound signature properties. The compiler is similar to the BUFF-lite transform for regular signature schemes, thus adding only a hash value $\mathsf{H}(\mathsf{pk}, m)$ of the scheme's public key and the message to the signature. It preserves the round complexity of the original scheme.

Direct Construction We show that the threshold BLS signature scheme can be modified without increasing the signature size to achieve exclusive ownership and the message-bound signature property in the random oracle model. The solution is to include the public key pk in the hashing step when computing the signature. This shows that for specific schemes, improved solutions exist.

Section	Description	S-CEO	S-DEO	M-S-UEO	MBS
4	Compiler: $\mathsf{TS}' \to \mathsf{TS}$	✓	✓	✓	✓
5	Construction: T-BLS$_{\mathsf{pk}}$	✓	✓	✓	✓
6.2	Analysis of TRaccoon [15]			✓	✓
6.3	Analysis of FROST [27]			✓	✓
6.4	Analysis of Thresholdizer [7]			✓	✓

Table 1: The sections and the security properties achieved by the resulting threshold signature schemes.

Inheriting BUFF properties We identify that if the threshold signature schemes are constructed with a specific structural form, namely, having an underlying standard signature scheme at the core, and if the underlying standard signature scheme achieves exclusive ownership and the message-bound signature property, then the properties transfer to the threshold signature scheme. We analyze some of the existing threshold signature schemes and provide proofs for beyond-unforgeability properties. This includes TRaccoon [15], FROST [27], and the universal thresholdizer of Boneh et al. [7].

Summary We summarize our constructions and transformations in Table 1. In Section 4, we provide the compiler that takes a threshold signature scheme TS' and transforms it into TS by adding a hash value. In Section 5, we show the modified BLS achieving the additional security properties. Lastly, in Section 6, we easily derive results for several threshold signatures [15, 27, 7] via our general inheritance result of the signature-to-threshold paradigm. The properties satisfied by the resultant threshold signature scheme are also provided in the table.

2 Preliminaries

Notations We denote by $\lambda \in \mathbb{N}$ the security parameter, usually employed in unary form as 1^λ that is implicitly given as input to all algorithms. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ is called negligible, if for every constant $c \geq 0$, there exists $\lambda_c \geq c$ s.t. $\forall \lambda \geq \lambda_c$ we have $\mathsf{negl}(\lambda) \leq \lambda^{-c}$. Given a polynomial $p(\cdot)$, an efficient randomized algorithm, A, is called probabilistic polynomial time, PPT in short, if its running time is bounded by a polynomial $p(|x|)$ for every input x. The set $\{1, \ldots, n\}$ is denoted as $[n]$ for a positive integer $n \in \mathbb{N}$. For the equality check of two elements, we use "=". The assignment operator is denoted with "$\leftarrow$", whereas the randomized assignment is denoted by $a \leftarrow\!\!\$\ A$, where A is a randomized algorithm.

Threshold Signature Schemes Below we recall the definition of threshold signature schemes as well as the corresponding security definitions for *unforgeability under chosen message attack* and *robustness under chosen message attack*. We focus here on non-interactive schemes.

Definition 2.1 (Threshold Signature). *Let $t, n \in \mathbb{N}$ such that $t \leq n$. A (n, t)-threshold signature scheme* TS *for a finite message space $\mathcal{M}$ having n signers with threshold t consists of a tuple of polynomial algorithms defined as follows:*

1. $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$*: The setup algorithm takes as input the security parameter 1^λ and outputs public parameters* pp.

2. $\mathsf{KeyGen}(\mathsf{pp}, n, t) \to (\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{sk}_i\}_{i\in[n]})$. *The distributed key generation protocol takes as input the number of signers n, the threshold t and outputs the verification key* pk*, the partial verification keys* $\{\mathsf{pk}_i\}_{i\in[n]}$ *and the signing keys* $\{\mathsf{sk}_i\}_{i\in[n]}$*, where secret key* sk_i *is held by the i-th signer.*

3. $\mathsf{PartSign}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_i, m) \to \sigma_{m,i}$. *The partial signing algorithm takes as input the verification key* pk*, the signing key* sk_i*, and the message $m \in \mathcal{M}$ and outputs the partial signature $\sigma_{m,i}$.*

4. $\mathsf{PartVer}(\mathsf{pp}, \mathsf{pk}, m, \mathsf{pk}_i, \sigma_{m,i}) \to 0/1$. *The partial signature verification algorithm takes as input the partial verification key* pk_i*, the message m, and the partial signature $\sigma_{m,i}$ and outputs* 0 *(reject) or* 1 *(accept).*

5. $\mathsf{PartComb}(\mathsf{pp}, \mathsf{pk}, \mathsf{S}, m, \{\mathsf{pk}_i, \sigma_{m,i}\}_{i\in\mathsf{S}}) \to \Sigma_m$. *The combining algorithm takes as inputs the verification key* pk*, the set of signers* S*, the message m, a set of tuples* $\{\mathsf{pk}_i, \sigma_{m,i}\}_{i\in[n]}$*, consisting of partial verification keys and partial signatures corresponding to the set* S *and outputs either a signature Σ_m or $\perp$.*

6. $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma_m) \to 0/1$. *The verification algorithm takes as input the verification key* pk*, the message m, the signature Σ_m and outputs* 0 *(reject) or* 1 *(accept). We assume that the algorithm never accepts the signature $\Sigma_m = \perp$.*

The standard digital signatures $\mathsf{Sig} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Ver})$ can be considered as a $(1,1)$-threshold signature scheme, where Sign would be $\mathsf{PartSign}$ and have no partial verification algorithm $\mathsf{PartVer}$ and combining algorithm $\mathsf{PartComb}$.

We could define the correctness of a threshold signature scheme here, stating that genuine partial signatures always validate under $\mathsf{PartVer}$ and that $\mathsf{PartComb}$ is able to construct a valid combined signature if it receives at least t valid partial signatures. We omit such a definition here as it can be considered as a special case of the robustness definition below.

Unforgeability and Robustness We require a (n,t)-threshold signature scheme to satisfy unforgeability and robustness. We provide descriptions of the oracles used in the security definitions in Fig 1. Besides the apparent oracles to corrupt parties and learn their secret keys (oracle $\mathcal{O}_{\mathsf{Corrupt}}$) and to create partial signatures for honest parties (oracle $\mathcal{O}_{\mathsf{PartSign}}$), we also have an oracle $\mathcal{O}_{\mathsf{augmSign}}$ that augments given partial signatures for (honest or corrupt) parties in a set $\mathcal{J}$ by freshly generated partial signatures for the remaining parties in a set $\mathcal{I}$ and derives the combined signatures. To the advantage of the adversary, it returns the derived signatures and the created partial signatures.

Definition 2.2 (Unforgeability under Chosen Message Attack)**.** *Let* TS *be a (n,t)-threshold signature scheme. We say that* TS *is* $\mathsf{UF\text{-}CMA}$ *secure if the following holds for all* PPT*adversaries $\mathcal{A}$, where* $\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}$ *is defined in Figure 2.*

$$\mathsf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t) = 1] \leq \mathsf{negl}(\lambda)$$

We say TS *is* $\mathsf{UF\text{-}CMA}$ *against static corruptions if all the* $\mathcal{O}_{\mathsf{Corrupt}}$ *queries are queried before any* $\mathcal{O}_{\mathsf{PartSign}}$ *oracle queries.*

We follow the definition of robustness from [13] but slightly adapt it to match our notions. The difference is that the approach in [13] separates violations of expected behavior according to (a single) invalid, honestly generated partial signatures, and according to the combined signatures from valid partial signatures. We combine this into a single attempt for partial signatures generated for users in a set $\mathcal{I}$ and for malicious contributions for users from a set $\mathcal{J}$. Both approaches are equivalent.

```
$\mathcal{O}_{\mathsf{Corrupt}}(i)$:
if $i \notin \mathcal{C}$ and $|\mathcal{C}| \geq t-1$:
    return $\perp$.
else
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$.
    return $\mathsf{sk}_i$.

$\mathcal{O}_{\mathsf{PartSign}}(m,i)$ :
return $\perp$ if $i \in \mathcal{C}$.
$\sigma_{m,i} \leftarrow\$ \mathsf{PartSign}(\mathsf{pp},\mathsf{pk},\mathsf{sk}_i,m)$.
$\mathcal{Q}_{\mathsf{psig}} \leftarrow \mathcal{Q}_{\mathsf{psig}} \cup \{(m,i)\}$.
return $\sigma_{m,i}$.

$\mathcal{O}_{\mathsf{augmSign}}(m,\mathcal{I},\mathcal{J},\{\sigma_{m,j}\}_{j\in\mathcal{J}})$:
return $\perp$ if $|\mathcal{I} \cup \mathcal{J}| < t$.
for $j \in \mathcal{J}$ do:
    if $\mathsf{PartVer}(\mathsf{pp},\mathsf{pk},m,\mathsf{pk}_j,\sigma_{m,j}) = 0$
    then return $\perp$.
for $i \in \mathcal{I}$ do:
    $\sigma_{m,i} \leftarrow\$ \mathsf{PartSign}(\mathsf{pp},\mathsf{pk},\mathsf{sk}_i,m)$.
$\Sigma_m \leftarrow\$ \mathsf{PartComb}(\mathsf{pp},\mathsf{pk},\mathcal{I}\cup\mathcal{J},$
        $m,\{\mathsf{pk}_k,\sigma_{m,k}\}_{k\in\mathcal{I}\cup\mathcal{J}})$.
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m,\Sigma_m,\mathcal{I})\}$.
return $(\Sigma_m,\{\sigma_{m,i}\}_{i\in\mathcal{I}})$.
```

Figure 1: The description of the oracles used in $\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}$, $\mathsf{Exp}^{\mathsf{RB\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}$, $\mathsf{Exp}^{\mathsf{S\text{-}CEO}}_{\mathsf{TS},\mathcal{A}}$, $\mathsf{Exp}^{\mathsf{S\text{-}DEO}}_{\mathsf{TS},\mathcal{A}}$, $\mathsf{Exp}^{\mathsf{S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}$, $\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}$, $\mathsf{Exp}^{\mathsf{MBS}}_{\mathsf{TS},\mathcal{A}}$.

```
$\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}(1^\lambda,n,t)$ :
$\mathsf{pp} \leftarrow\$ \mathsf{Setup}(1^\lambda)$.
$(\mathsf{pk},\{\mathsf{pk}_i\}_{i\in[n]},\{\mathsf{sk}_i\}_{i\in[n]}) \leftarrow\$ \mathsf{KeyGen}(\mathsf{pp},n,t)$.
$\mathcal{Q}_{\mathsf{psig}} \leftarrow \emptyset, \mathcal{C} \leftarrow \emptyset, \mathcal{Q} \leftarrow \emptyset$.
$(m^*,\Sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{PartSign}}(\cdot,\cdot),\mathcal{O}_{\mathsf{Corrupt}}(\cdot)}(\mathsf{pk},\{\mathsf{pk}_i\}_{i\in[n]})$.
$\mathcal{Q}[m^*] = \{i : (m^*,i) \in \mathcal{Q}_{\mathsf{psig}}\}$.
$d \leftarrow \mathsf{Ver}(\mathsf{pp},\mathsf{pk},m^*,\Sigma^*)$.
return $[d = 1 \wedge |\mathcal{Q}[m^*] \cup \mathcal{C}| < t]$.
```

Figure 2: The unforgeability game $\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}$ for a (n,t)-threshold signature scheme where the adversary is allowed to make adaptive corruptions.

Definition 2.3 (Robustness under Chosen Message Attack)**.** *Let* TS *be a* (n,t)*-threshold signature scheme. We say that* TS *is* $\mathsf{RB\text{-}CMA}$ *secure if the following holds for all* PPT*adversaries* $\mathcal{A}$*, where* $\mathsf{Exp}^{\mathsf{RB\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}$ *is defined in Figure 3.*

$$\mathsf{Adv}^{\mathsf{RB\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{RB\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}(1^\lambda,n,t) = 1] \leq \mathsf{negl}(\lambda)$$

3 Beyond-Unforgeability Definitions

We use here the definitions of the beyond-unforgeability security notions for signature schemes [31, 25, 10, 11] and transfer them to the threshold setting. At the end of this section, we discuss the relationship between the properties.

3.1 Definitions

The first definition, strong conservative exclusive ownership ($\mathsf{S\text{-}CEO}$), states that the adversary should not be able to transfer a threshold signature for a message to which at least one honest

```
$\underline{\mathsf{Exp}^{\mathsf{RB\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t)}$ :
$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$.
$(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{sk}_i\}_{i\in[n]}) \leftarrow\!\!\$\ \mathsf{KeyGen}(\mathsf{pp}, n, t)$.
$\mathcal{Q}_{\mathsf{psig}} \leftarrow \emptyset, \mathcal{C} \leftarrow \emptyset$.
$(m, \mathcal{I}, \mathcal{J}, \{\sigma_{m,j}\}_{j\in\mathcal{J}}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{PartSign}}(\cdot,\cdot),\mathcal{O}_{\mathsf{Corrupt}}(\cdot)}(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]})$.
// verification of honest partial signatures must be successful
for $i$ in $\mathcal{I}$ do
    $\sigma_{m,i} \leftarrow\!\!\$\ \mathsf{PartSign}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_i, m)$.
    if $\mathsf{PartVer}(\mathsf{pp}, \mathsf{pk}, m, \mathsf{pk}_i, \sigma_{m,i}) = 0$ : return 1.
// combining sufficiently many valid partial signatures must yield valid signature
if $|\mathcal{I} \cup \mathcal{J}| < t$: return 0.
for $j$ in $\mathcal{J}$ do:
    if $\mathsf{PartVer}(\mathsf{pp}, \mathsf{pk}, m, \mathsf{pk}_j, \sigma_{m,j}) = 0$ : return 0.
$\Sigma_m \leftarrow \mathsf{PartComb}(\mathsf{pp}, \mathsf{pk}, \mathcal{I} \cup \mathcal{J}, m, \{\mathsf{pk}_k, \sigma_k\}_{k\in\mathcal{I}\cup\mathcal{J}})$.
if $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma_m) = 0$ : return 1.
return 0.
```

Figure 3: The robustness security game $\mathsf{Exp}^{\mathsf{RB\text{-}CMA}}_{\mathsf{TS},\mathcal{A}}$ for a (n, t)-threshold signature scheme.

user contributed (via a query to the augmented signing oracle $\mathcal{O}_{\mathsf{augmSign}}$ as part of the set $\mathcal{I} \setminus \mathcal{C}$ of uncorrupted contributions) to a signature under a different public key:

Definition 3.1 (Strong Conservative Exclusive Ownership)**.** *Let* TS *be a (n, t)-threshold signature scheme. We say that* TS *provides strong conservative exclusive ownership(*S-CEO*) if, for every* PPT*adversary $\mathcal{A}$, the following inequality holds:*

$$\mathsf{Adv}^{\mathsf{S\text{-}CEO}}_{\mathsf{TS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{S\text{-}CEO}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t) = 1] \leq \mathsf{negl}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{S\text{-}CEO}}_{\mathsf{TS},\mathcal{A}}$ is defined in Figure 4.

```
$\underline{\mathsf{Exp}^{\mathsf{S\text{-}CEO}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t)}$:
$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$.
$(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{sk}_i\}_{i\in[n]}) \leftarrow\!\!\$\ \mathsf{KeyGen}(\mathsf{pp}, n, t)$.
$\mathcal{Q}_{\mathsf{psig}} \leftarrow \emptyset, \mathcal{Q} \leftarrow \emptyset, \mathcal{C} \leftarrow \emptyset$.
$(m^*, \mathsf{pk}^*, \Sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{augmSign}}(\cdots),\mathcal{O}_{\mathsf{PartSign}}(\cdot,\cdot),\mathcal{O}_{\mathsf{Corrupt}}(\cdot)}(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]})$.
$d \leftarrow \mathsf{Ver}(\mathsf{pp}, \mathsf{pk}^*, m^*, \Sigma^*)$.
// check validity and that at least one honest contribution:
return $[d = 1 \wedge \mathsf{pk}^* \neq \mathsf{pk} \wedge \exists (m^*, \Sigma^*, \mathcal{I}) \in \mathcal{Q} : \mathcal{I} \setminus \mathcal{C} \neq \emptyset]$.
```

Figure 4: The strong conservative exclusive ownership game $\mathsf{Exp}^{\mathsf{S\text{-}CEO}}_{\mathsf{TS},\mathcal{A}}$ for a (n, t)-threshold signature scheme.

The next definition, strong destructive exclusive ownership (S-DEO), is similar to S-CEO, but now the adversary has to modify the message as well:

Definition 3.2 (Strong Destructive Exclusive Ownership)**.** *Let* TS *be a (n, t)-threshold signature scheme. We say that* TS *provides strong destructive exclusive ownership (*S-DEO*) if, for every*

PPT *adversary* $\mathcal{A}$*, the following holds:*

$$\mathsf{Adv}^{\mathsf{S\text{-}DEO}}_{\mathsf{TS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{S\text{-}DEO}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t) = 1] \leq \mathsf{negl}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{S\text{-}DEO}}_{\mathsf{TS},\mathcal{A}}$ *is defined in Figure 5.*

```
$\mathsf{Exp}^{\mathsf{S\text{-}DEO}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t)$ :
  $\mathsf{pp} \leftarrow\$ \mathsf{Setup}(1^\lambda)$.
  $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{sk}_i\}_{i\in[n]}) \leftarrow\$ \mathsf{KeyGen}(\mathsf{pp}, n, t)$.
  $\mathcal{Q}_{\mathsf{psig}} \leftarrow \emptyset, \mathcal{Q} \leftarrow \emptyset, \mathcal{C} \leftarrow \emptyset$.
  $(m^*, \mathsf{pk}^*, \Sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{augmSign}}(\cdots), \mathcal{O}_{\mathsf{PartSign}}(\cdot,\cdot), \mathcal{O}_{\mathsf{Corrupt}}(\cdot)}(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]})$.
  $d \leftarrow \mathsf{Ver}(\mathsf{pp}, \mathsf{pk}^*, m^*, \Sigma^*)$.
  // check validity and that at least one honest contribution for different message:
  return $[d = 1 \wedge \mathsf{pk}^* \neq \mathsf{pk} \wedge \exists (m, \Sigma^*, \mathcal{I}) \in \mathcal{Q} : (\mathcal{I} \setminus \mathcal{C} \neq \emptyset \wedge m \neq m^*)]$
```

Figure 5: The strong destructive exclusive ownership game $\mathsf{Exp}^{\mathsf{S\text{-}DEO}}_{\mathsf{TS},\mathcal{A}}$ for a (n, t)-threshold signature scheme.

The notion of strong universal exclusive ownership ($\mathsf{S\text{-}UEO}$) combines the two previous definitions into one:

Definition 3.3 (Strong Universal Exclusive Ownership)**.** *Let* TS *be a* (n, t)*-threshold signature scheme. We say that* TS *provides strong universal exclusive ownership (*$\mathsf{S\text{-}UEO}$*) if, for every* PPT *adversary* $\mathcal{A}$*, the following holds:*

$$\mathsf{Adv}^{\mathsf{S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t) = 1] \leq \mathsf{negl}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}$ *is defined in Figure 6.*

```
$\mathsf{Exp}^{\mathsf{S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t)$ :
  $\mathsf{pp} \leftarrow\$ \mathsf{Setup}(1^\lambda)$.
  $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{sk}_i\}_{i\in[n]}) \leftarrow\$ \mathsf{KeyGen}(\mathsf{pp}, n, t)$.
  $\mathcal{Q}_{\mathsf{psig}} \leftarrow \emptyset, \mathcal{Q} \leftarrow \emptyset, \mathcal{C} \leftarrow \emptyset$.
  $(m^*, \mathsf{pk}^*, \Sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{augmSign}}(\cdots), \mathcal{O}_{\mathsf{PartSign}}(\cdot,\cdot), \mathcal{O}_{\mathsf{Corrupt}}(\cdot)}(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]})$.
  $d \leftarrow \mathsf{Ver}(\mathsf{pp}, \mathsf{pk}^*, m^*, \Sigma^*)$.
  // check validity and that at least one honest contribution:
  return $[d = 1 \wedge \mathsf{pk}^* \neq \mathsf{pk} \wedge \exists (m, \Sigma^*, \mathcal{I}) \in \mathcal{Q} : \mathcal{I} \setminus \mathcal{C} \neq \emptyset]$
```

Figure 6: The strong universal exclusive ownership game $\mathsf{Exp}^{\mathsf{S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}$ for a (n, t)-threshold signature scheme.

Brendel et al. [10] strengthened the notion of $\mathsf{S\text{-}UEO}$ to a malicious version, denoted $\mathsf{M\text{-}S\text{-}UEO}$, in which we even waive the requirement of transforming a partly honest signature. Now, the adversary can choose a signature which is valid for two maliciously chosen public keys and messages:

Definition 3.4 (Malicious-Strong Universal Exclusive Ownership)**.** *Let* TS *be a* (n, t)*-threshold signature scheme. We say that* TS *provides malicious-strong universal exclusive ownership (*$\mathsf{M\text{-}S\text{-}UEO}$*) if, for every* PPT *adversary* $\mathcal{A}$*, the following inequality holds:*

$$\mathsf{Adv}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t) = 1] \leq \mathsf{negl}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}$ *is defined on the left-hand side of Figure 7.*

```
$\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t)$:
$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$.
$(m, m^*, \mathsf{pk}, \mathsf{pk}^*, \Sigma) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pp})$.
$d \leftarrow \mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma)$.
$d^* \leftarrow \mathsf{Ver}(\mathsf{pp}, \mathsf{pk}^*, m^*, \Sigma)$.
return $[d = 1 \land d^* = 1 \land \mathsf{pk} \neq \mathsf{pk}^*]$.
```

```
$\mathsf{Exp}^{\mathsf{MBS}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t)$:
$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$.
$(m, m^*, \mathsf{pk}, \Sigma) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pp})$.
$d \leftarrow \mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma)$.
$d^* \leftarrow \mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m^*, \Sigma)$.
return $[d = 1 \land d^* = 1 \land m \neq m^*]$.
```

Figure 7: Definitions for malicious-strong universal exclusive ownership game $\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}$ and message-bound Signatures game $\mathsf{Exp}^{\mathsf{MBS}}_{\mathsf{TS},\mathcal{A}}$ for a (n, t)-threshold signature scheme.

The notion of message-bound signatures MBS informally says that a signature under a public key is binding to the message, i.e., the adversary cannot find two valid messages to one signature:

Definition 3.5 (Message-Bound Signatures). *Let* TS *be a (n, t)-threshold signature scheme. We say that* TS *provides message-bound signatures (*MBS*) if, for every* PPT*adversary $\mathcal{A}$, the following inequality holds:*

$$\mathsf{Adv}^{\mathsf{MBS}}_{\mathsf{TS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{MBS}}_{\mathsf{TS},\mathcal{A}}(1^\lambda, n, t) = 1] \leq \mathsf{negl}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{MBS}}_{\mathsf{TS},\mathcal{A}}$ is defined on the right-hand side of Figure 7.

Our notions S-CEO, S-DEO, and S-UEO of exclusive ownership require that at least one honest party contributes to the signature generation ($\mathcal{I} \backslash \mathcal{C} \neq \emptyset$) and demand that stealing message-signature pairs as above should be infeasible. This reflects the protection of the honest party declaring its will to sign that message. One could also define all notions more liberal and forgo the requirement $\mathcal{I} \backslash \mathcal{C} \neq \emptyset$ in all cases entirely. This would reflect that honest parties united under the public key pk are protected against takeovers, even if they have not contributed to an actual signature. We call these properties *extra-strong* and denote the security notions as ES-CEO, ES-DEO, and ES-UEO. It is obvious that each of the extra-strong notions implies the corresponding strong notion. In Appendix A.1 we show that the converse does not hold in general.

3.2 Relationship Between Notions

We note that if we view regular signature schemes as $(1, 1)$-threshold signature schemes in a straightforward way, then all separations in [11] of the properties immediately transfer to the (general) threshold setting. We discuss here in detail that the relationships of exclusive ownership notions in the threshold case partly agree with results for regular schemes but that there are also differences that require robustness as an additional property.

Proposition 3.6. *A threshold signature scheme* TS *is* S-CEO *and* S-DEO *if and only if it is* S-UEO*.*

It also holds that a scheme is ES-CEO and ES-DEO if and only if it is ES-UEO. We only discuss the case of strong notions here (with $\mathcal{I} \backslash \mathcal{C} \neq \emptyset$); the case of extra-strong notions follows analogously.

Proof. In order to win the S-UEO game, the adversary submits $(m^*, \mathsf{pk}^*, \Sigma^*)$. Then, $\mathsf{pk}^* \neq \mathsf{pk}$ and Σ^* must have been a valid signature of some query m, and there must be at least one honest contribution during the generation of Σ^*. If $m = m^*$, the S-CEO security is broken. Otherwise, $m \neq m^*$ and we break S-DEO security. To prove the other direction, any successful attack on S-CEO or S-DEO leads to a successful attack against S-UEO game. The reason is that any attack against the more restrictive versions S-CEO (where the adversary needs to re-use a queried message m^* in the forgery) or $\mathsf{S\text{-}DEO}$ (where the adversary needs to use a different message m^* in the forgery) also constitutes a valid attack against $\mathsf{S\text{-}UEO}$ where no restriction is put on m^*. □

We next show that the malicious version $\mathsf{M\text{-}S\text{-}UEO}$ implies the version $\mathsf{S\text{-}UEO}$. Unlike in the case of ordinary signatures, however, we also rely on the robustness property $\mathsf{RB\text{-}CMA}$ of the threshold scheme for this implication:

Proposition 3.7. *If a threshold signature scheme* TS *is* $\mathsf{M\text{-}S\text{-}UEO}$ *and* $\mathsf{RB\text{-}CMA}$*, it is also* $\mathsf{S\text{-}UEO}$*. More precisely, for any adversary $\mathcal{A}$ against* $\mathsf{S\text{-}UEO}$*, making at most q_a oracle queries to $\mathcal{O}_{\mathsf{augmSign}}$, there exists algorithms $\mathcal{A}'_{\mathsf{RB\text{-}CMA}}$ and $\mathcal{A}'_{\mathsf{M\text{-}S\text{-}UEO}}$ with roughly the same running time as $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\mathsf{S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}}(\lambda) \leq q_a \cdot \mathsf{Adv}^{\mathsf{RB\text{-}CMA}}_{\mathsf{TS},\mathcal{A}'_{\mathsf{RB\text{-}CMA}}}(\lambda) + \mathsf{Adv}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}'_{\mathsf{M\text{-}S\text{-}UEO}}}(\lambda).$$

We note that the bound in the theorem even holds for the stronger variant $\mathsf{ES\text{-}UEO}$, not only for $\mathsf{S\text{-}UEO}$. The reason is that all reductions in the proof would also work against a $\mathsf{ES\text{-}UEO}$ adversary.

Proof. If there exists an efficient PPT algorithm $\mathcal{A}$ that breaks $\mathsf{S\text{-}UEO}$ of TS, we can use $\mathcal{A}$ to build $\mathcal{A}'_{\mathsf{RB\text{-}CMA}}$ against the robustness of the scheme or to construct $\mathcal{A}'_{\mathsf{M\text{-}S\text{-}UEO}}$ that can break the $\mathsf{M\text{-}S\text{-}UEO}$ security game. We start with a game hop to exclude attacks against $\mathsf{S\text{-}UEO}$ for which there exists $(m, \Sigma^*, \mathcal{I}) \in \mathcal{Q}$ but where Σ^* does not constitute a valid signature for m under pk. The reduction $\mathcal{A}'_{\mathsf{RB\text{-}CMA}}$ against robustness works as follows:

1. The reduction receives $\mathsf{pp}, \mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}$ as input.
2. Let $\mathcal{A}$ make at most q_a oracle queries to oracle $\mathcal{O}_{\mathsf{augmSign}}$. Then $\mathcal{A}'_{\mathsf{RB\text{-}CMA}}$ initially picks an index $k \leftarrow\!\!\$\, [q_a]$.
3. The reduction runs $\mathcal{A}$ on $\mathsf{pp}, \mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}$.
4. It answers all queries of $\mathcal{A}$ to $\mathcal{O}_{\mathsf{Corrupt}}(i)$ with sk_i by querying its own corruption oracle. For any query $\mathcal{O}_{\mathsf{PartSign}}(m, i)$ of $\mathcal{A}$ to the partial signing oracle, the reduction calls its own partial signing oracle to compute the requested partial signature according to the experiment.
5. For the j-th query of $\mathcal{A}$ to oracle $\mathcal{O}_{\mathsf{augmSign}}$ for $j \neq k$, the reduction once more exploits knowledge of the secret keys or the partial signing oracle to compute the answer according to the experiment. For $j = k$ and query $(m, \mathcal{I}, \mathcal{J}, \{\sigma_{m,j}\}_{j\in\mathcal{J}})$, however, the reduction $\mathcal{A}'_{\mathsf{RB\text{-}CMA}}$ outputs this tuple and stops.

Note that, up to the point where the reduction stops, the simulation is identically distributed to an actual attack of $\mathcal{A}$ against $\mathsf{S\text{-}UEO}$. If there is some query $(m, \Sigma^*, \mathcal{I}) \in \mathcal{Q}$ in the attack (or simulation) of $\mathcal{A}$ where Σ^* does not constitute a valid signature, then this can only be because the combine algorithm $\mathsf{PartComb}$ outputs an invalid signature. Using that we only add $(m, \Sigma^*, \mathcal{I})$ to the set $\mathcal{Q}$ if all partial signatures for parties in $\mathcal{J}$ have been verified, a failure of algorithm $\mathsf{PartComb}$ can only happen if an honestly generated partial signature for a user in $\mathcal{I}$ is invalid or if the combination itself for all valid partial signatures from $\mathcal{I} \cup \mathcal{J}$ fails. Either of the two events constitutes a break of robustness. Hence, if there is such a query to oracle $\mathcal{O}_{\mathsf{augmSign}}$ resulting in $(m, \Sigma^*, \mathcal{I}) \in \mathcal{Q}$ with invalid Σ^*, then our reduction has a probability of $1/q_a$ to guess the first query of this type correctly and to violate robustness.

We may from now on assume that $\mathcal{A}$ against $\mathsf{S\text{-}UEO}$ outputs $m^*, \Sigma^*, \mathsf{pk}^* \neq \mathsf{pk}$ such that there is some $(m, \Sigma^*, \mathcal{I}) \in \mathcal{Q}$ where Σ^* is valid for m under pk. The description of $\mathcal{A}'_{\mathsf{M\text{-}S\text{-}UEO}}$ against $\mathsf{M\text{-}S\text{-}UEO}$ is now as follows:

1. On receiving pp from the challenger, sample $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{sk}_i\}_{i\in[n]}) \leftarrow\!\!\$\ \mathsf{KeyGen}(\mathsf{pp}, n, t)$ and send $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]})$ to $\mathcal{A}$.

2. For all queries from $\mathcal{A}$, respond correspondingly. This is possible as $\mathcal{A}'_{\mathsf{M\text{-}S\text{-}UEO}}$ has access to $\{\mathsf{sk}_i\}_{i\in[n]}$.

3. On receiving $(m^*, \mathsf{pk}^*, \Sigma^*)$, find m such that $(m, \Sigma^*, \mathcal{I}) \in \mathcal{Q}$. Finally send $(m, m^*, \mathsf{pk}, \mathsf{pk}^*, \Sigma^*)$ to the challenger.

In this reduction, $\mathcal{A}'_{\mathsf{M\text{-}S\text{-}UEO}}$ wins the game if $\mathcal{A}$ wins. The latter requires $\mathsf{pk} \neq \mathsf{pk}^*$, which is also necessary in the game of $\mathcal{A}'_{\mathsf{M\text{-}S\text{-}UEO}}$. Furthermore, the signature Σ^* must be valid for m^* and pk^* according to the game's requirement. By the previous game hop, we must also have that Σ^* is valid for m under pk, such that all stipulations for a successful attack of $\mathcal{A}'_{\mathsf{M\text{-}S\text{-}UEO}}$ against $\mathsf{M\text{-}S\text{-}UEO}$ are satisfied. □

We show in Appendix A.2 that robustness is in fact necessary for this implication to hold, by presenting a scheme that is $\mathsf{M\text{-}S\text{-}UEO}$ but not $\mathsf{RB\text{-}CMA}$ and neither $\mathsf{S\text{-}UEO}$.

4 Construction of BUFF Compiler

In this section, we provide a generic compiler that takes as input any threshold signature scheme $\mathsf{TS}' = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{PartSign}', \mathsf{PartVer}', \mathsf{PartComb}', \mathsf{Ver}')$ and a collision-resistant hash function H, and outputs a threshold signature scheme $\mathsf{TS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PartSign}, \mathsf{PartVer}, \mathsf{PartComb}, \mathsf{Ver})$ which satisfies unforgeability, robustness, exclusive ownership, and message-bound signature properties.

The transformation is provided in Figure 8. It follows the BUFF-lite transformation [11], appending $\mathsf{H}(\mathsf{pk}, m)$ to the signature for some recoverable encoding of (pk, m) into bit strings. For example, if all keys are of fixed size, then one could simply concatenate pk and m.

The security of the transformation is based on the collision-resistance of the hash function H. Formally, we define collision-resistance of H by considering an algorithm $\mathcal{A}$ outputting distinct inputs $x \neq x'$ such that $\mathsf{H}(x) = \mathsf{H}(x')$. Let $\Pr[\mathsf{Exp}^{\mathsf{CR}}_{\mathsf{H},\mathcal{A}} = 1]$ denote the probability that $\mathcal{A}$ is successful. Note that we do not use here asymptotic security notions, since (non-uniform) adversaries always manage to output collisions. Rather, we rely on the constructive approach [32] where our reductions to collision resistance of H will give concrete algorithms creating collisions.

Theorem 4.1. *Assuming the underlying threshold signature scheme* TS' *is* $\mathsf{UF\text{-}CMA}$ *secure, then,* TS *is* $\mathsf{UF\text{-}CMA}$ *secure. Specifically, for any adversary* $\mathcal{A}$ *against* $\mathsf{UF\text{-}CMA}$ *we have*

$$\Pr[\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\mathsf{TS},\mathcal{A}} = 1] \leq \Pr[\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\mathsf{TS}',\mathcal{A}'} = 1]$$

for an algorithm $\mathcal{A}'$ *running in roughly the same time as* $\mathcal{A}$.

Proof. We can prove the above inequality through a reduction to the $\mathsf{UF\text{-}CMA}$ game of the underlying threshold signature scheme TS'. Consider an adversary $\mathcal{A}$ that can break the $\mathsf{UF\text{-}CMA}$ security of the TS scheme. We build an adversary $\mathcal{A}'$ that breaks the $\mathsf{UF\text{-}CMA}$ property of the underlying scheme TS'. We describe the adversary $\mathcal{A}'$ below.

1. On receiving pp' from the TS' challenger, sample H, and send $\mathsf{pp} = (\mathsf{pp}', \mathsf{H})$ to $\mathcal{A}$.

2. On receiving $(\mathsf{pk}', \{\mathsf{pk}'_i\}_{i\in[n]})$ from the challenger, forward it to $\mathcal{A}$.

```
$\underline{\mathsf{Setup}(1^\lambda)}$:
$\mathsf{pp}' \leftarrow\!\!\$\ \mathsf{Setup}'(1^\lambda)$.
Let $\mathsf{H} : \{0,1\}^* \rightarrow \{0,1\}^\ell$, be a collision-resistant hash function.
return $\mathsf{pp} = (\mathsf{pp}', \mathsf{H})$.

$\underline{\mathsf{KeyGen}(\mathsf{pp}, n, t)}$:
Parse $\mathsf{pp}$ as $(\mathsf{pp}', \mathsf{H})$.
$(\mathsf{pk}', \{\mathsf{pk}'_i, \mathsf{sk}'_i\}_{i\in[n]}) \leftarrow\!\!\$\ \mathsf{KeyGen}(\mathsf{pp}', n, t)$
for $i \in [n]$, $\mathsf{pk} \leftarrow \mathsf{pk}', \mathsf{pk}_i \leftarrow \mathsf{pk}'_i$, $\mathsf{sk}_i \leftarrow \mathsf{sk}'_i$.
return $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{sk}_j)$ to signer $j$ for all $j \in [n]$.

$\underline{\mathsf{PartSign}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_i, m)}$:
Parse $\mathsf{pp}$ as $(\mathsf{pp}', \mathsf{H})$.
$\sigma'_{m,i} \leftarrow \mathsf{PartSign}'(\mathsf{pp}', \mathsf{pk}, \mathsf{sk}_i, m)$.
return $\sigma_{m,i} = \sigma'_{m,i}$.

$\underline{\mathsf{PartVer}(\mathsf{pp}, \mathsf{pk}, m, \mathsf{pk}_i, \sigma_{m,i})}$:
Parse $\mathsf{pp}$ as $(\mathsf{pp}', \mathsf{H})$.
If $\mathsf{PartVer}'(\mathsf{pp}', \mathsf{pk}, m, \mathsf{pk}_i, \sigma_{m,i}) = 1$
    return 1.
Else
    return 0.

$\underline{\mathsf{PartComb}(\mathsf{pp}, \mathsf{pk}, \mathsf{S}, m, \{\mathsf{pk}_i, \sigma_{m,i}\}_{i\in\mathsf{S}})}$:
Parse $\mathsf{pp}$ as $(\mathsf{pp}', \mathsf{H})$.
assert $|\mathsf{S}| \geq t$.
For each $i \in \mathsf{S}$: assert $\mathsf{PartVer}(\mathsf{pp}', \mathsf{pk}, m, \mathsf{pk}_i, \sigma_{m,i}) = 1$.
$\Sigma'_m \leftarrow \mathsf{PartComb}'(\mathsf{pp}', \mathsf{pk}, \mathsf{S}, m, \{\mathsf{pk}_i, \sigma_{m,i}\}_{i\in\mathsf{S}})$.
$h_m \leftarrow \mathsf{H}(\mathsf{pk}, m)$.
return $\Sigma_m \leftarrow (\Sigma'_m, h_m)$.

$\underline{\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma_m)}$:
Parse $\mathsf{pp}$ as $(\mathsf{pp}', \mathsf{H})$.
Parse $\Sigma_m$ as $(\Sigma'_m, h_m)$.
If $\mathsf{Ver}'(\mathsf{pp}', \mathsf{pk}, m, \Sigma'_m) = 1$ and $\mathsf{H}(\mathsf{pk}, m) = h_m$:
    return 1.
Else
    return 0.
```

Figure 8: Generic compiler that takes as input any threshold signature scheme TS' and outputs a threshold signature scheme TS satisfying $\mathsf{S\text{-}UEO}$, $\mathsf{M\text{-}S\text{-}UEO}$ and MBS.

3. On receiving partial signing oracle queries (m, i) from $\mathcal{A}$, forward it to the challenger. After receiving $\sigma_{m,i}$ from the challenger, hand it to $\mathcal{A}$.

4. On receiving corruption queries i from $\mathcal{A}$, forward it to challenger and send the response sk_i to $\mathcal{A}$.

5. When $\mathcal{A}$ submits the challenge $(m^*, \Sigma^*_{m^*})$ for $\Sigma^*_{m^*} = (\Sigma'_{m^*}, h_{m^*})$ check if it satisfies the verifiability condition. If it is verified, forward (m^*, Σ'_{m^*}) to the challenger.

If the challenge satisfies the verifiability condition of TS, it will also satisfy the verifiability of TS' as the Ver' algorithm is run inside Ver algorithm. All the partial verification oracle queries are also forwarded, thus, $|\mathcal{Q}[m^*] \cup \mathcal{C}| < t$. Thus, any valid challenge against the $\mathsf{UF\text{-}CMA}$ game of TS is also a valid challenge against the $\mathsf{UF\text{-}CMA}$ game of TS'. □

Theorem 4.2. *Assuming* H *is a collision-resistant hash function, and the underlying threshold signature scheme* TS' *is* $\mathsf{RB\text{-}CMA}$ *secure, then* TS *is* $\mathsf{RB\text{-}CMA}$ *secure. In particular, for any adversary* $\mathcal{A}$ *against* $\mathsf{RB\text{-}CMA}$ *we have*

$$\Pr[\mathsf{Exp}^{\mathsf{RB\text{-}CMA}}_{\mathsf{TS},\mathcal{A}} = 1] \leq \Pr[\mathsf{Exp}^{\mathsf{RB\text{-}CMA}}_{\mathsf{TS}',\mathcal{A}'} = 1]$$

for an algorithm $\mathcal{A}'$ *running in roughly the same time as* $\mathcal{A}$.

Proof. The robustness property follows from the robustness property of the underlying scheme TS'. There are two winning cases for the adversary, (1) honestly generated partial signatures do not satisfy the verification check, and (2) partial signatures pass verification, but the combined signature does not satisfy verification.

We analyze the first winning case. The $\mathsf{PartVer}$ algorithm merely runs the $\mathsf{PartVer}'$ algorithm. Hence, any successful attack for partial signatures against TS immediately yields a valid attack against TS'. For the second winning case, the Ver algorithm runs the Ver' and checks the $h_m = \mathsf{H}(\mathsf{pk}, m)$. The hash value $\mathsf{H}(\mathsf{pk}, m)$ is computed correctly by the $\mathsf{PartComb}$ algorithm. Thus, the hash check in the verification will be satisfied. Any failure to verify the final signature must therefore be due to the verification of the signature of TS'.

In summary, the adversary wins the game by failing verification checks $\mathsf{PartVer}'$ or Ver'. From the robustness of the TS' scheme, the probability of the verification checks failing is negligible. □

Theorem 4.3. *Assuming* H *is a collision-resistant hash function our construction is* $\mathsf{S\text{-}CEO}$ *secure. Specifically, for any adversary* $\mathcal{A}$ *against* $\mathsf{S\text{-}CEO}$*, we have*

$$\Pr[\mathsf{Exp}^{\mathsf{S\text{-}CEO}}_{\mathsf{TS},\mathcal{A}} = 1] \leq \Pr[\mathsf{Exp}^{\mathsf{CR}}_{\mathsf{H},\mathcal{A}'} = 1]$$

for algorithm $\mathcal{A}'$ *running in roughly the same time as* $\mathcal{A}$.

The proof actually shows that we also achieve the stronger version $\mathsf{ES\text{-}CEO}$.

Proof. Assume there exists an adversary $\mathcal{A}$ that can generate a valid signature $(\mathsf{pk}^*, m^*, (\Sigma_{m^*}, h_{m^*}))$ with respect to the $\mathsf{S\text{-}CEO}$ game. Then, there exists an entry $(m^*, (\Sigma_{m^*}, h_{m^*}), \mathcal{I}_{m^*}) \in \mathcal{Q}$ with $\Sigma^*_{m^*} \neq \perp$ (or else the adversary $\mathcal{A}$ cannot win, because such signatures are never accepted by assumption). It follows that all checks by algorithm $\mathsf{PartComb}$ succeeded when creating the signature in a call to oracle $\mathcal{O}_{\mathsf{augmSign}}$. In particular, $\mathsf{PartComb}$ has appended a valid hash value $h_{m^*} = \mathsf{H}(\mathsf{pk}, m^*)$ to the signature. For $\mathcal{A}$ to win for the very same signature $\Sigma^*_{m^*}$ with hash value h_{m^*}, its output must pass verification $\mathsf{PartVer}$ according to the experiment for the same message m^* but a different public key $\mathsf{pk}^* \neq \mathsf{pk}$. Verification, however, also includes a check that the hash value is sound. This implies $\mathsf{H}(\mathsf{pk}, m^*) = \mathsf{H}(\mathsf{pk}^*, m^*)$, breaking the collision resistance property of the hash function H, because the distinct pairs (pk, m^*), (pk^*, m^*) must encode to different strings. □

Theorem 4.4. *Assuming* H *is a collision-resistant hash function our construction is* $\mathsf{S\text{-}DEO}$ *secure. Specifically, for any adversary* $\mathcal{A}$ *against* $\mathsf{S\text{-}DEO}$*, we have*

$$\Pr[\mathsf{Exp}^{\mathsf{S\text{-}DEO}}_{\mathsf{TS},\mathcal{A}} = 1] \leq \Pr[\mathsf{Exp}^{\mathsf{CR}}_{\mathsf{H},\mathcal{A}'} = 1]$$

for algorithm $\mathcal{A}'$ *running in roughly the same time as* $\mathcal{A}$.

Once more, we actually achieve the stronger property $\mathsf{ES\text{-}DEO}$.

Proof. The proof is almost identical to the one for $\mathsf{S\text{-}CEO}$. Only this time the adversary also needs to modify the message $m^* \neq m$, in addition to the public key, in one of the queries $(m, \Sigma^*_m, \mathcal{I}_m) \in \mathcal{Q}$. The argument above again shows that this implies

$$\mathsf{H}(\mathsf{pk}, m) = \mathsf{H}(\mathsf{pk}^*, m^*)$$

for different $(\mathsf{pk}, m) \neq (\mathsf{pk}^*, m^*)$, breaking the collision resistance property of the hash function H. □

As we have shown in Proposition 3.6, $\mathsf{S\text{-}CEO}$ and $\mathsf{S\text{-}DEO}$ are equivalent to $\mathsf{S\text{-}UEO}$, we omit the proof of $\mathsf{S\text{-}UEO}$ security. We now provide the proof of $\mathsf{M\text{-}S\text{-}UEO}$ security below.

Theorem 4.5. *Assuming* H *is a collision-resistant hash function, our construction is* $\mathsf{M\text{-}S\text{-}UEO}$ *secure. Specifically, for any adversary* $\mathcal{A}$ *against* $\mathsf{M\text{-}S\text{-}UEO}$*, we have*

$$\Pr[\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{TS},\mathcal{A}} = 1] \leq \Pr[\mathsf{Exp}^{\mathsf{CR}}_{\mathsf{H},\mathcal{A}'} = 1]$$

for algorithm $\mathcal{A}'$ *running in roughly the same time as* $\mathcal{A}$*.*

Proof. Assume there exists an adversary $\mathcal{A}$ that can generate a valid signature $(m, m^*, \mathsf{pk}, \mathsf{pk}^*, (\Sigma, h))$ with respect to the $\mathsf{M\text{-}S\text{-}UEO}$ game. Then we get $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma, h) = 1$ and $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}^*, m^*, \Sigma, h) = 1$, such that we must have

$$\mathsf{H}(\mathsf{pk}, m) = h \quad \text{and} \quad \mathsf{H}(\mathsf{pk}^*, m^*) = h.$$

This implies $\mathsf{H}(\mathsf{pk}, m) = \mathsf{H}(\mathsf{pk}^*, m^*)$, which breaks the collision resistance property of the hash function H. □

Theorem 4.6. *Assuming* H *is a collision-resistant hash function, our construction is* MBS *secure.*

Proof. Assume there exists an adversary $\mathcal{A}$ that can generate a valid output $(m, m^*, \mathsf{pk}, \mathsf{pk}^*, (\Sigma, h))$ with respect to the MBS game. Then we conclude that $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma, h) = 1$ and $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}^*, m^*, \Sigma, h) = 1$, such that we must have

$$\mathsf{H}(\mathsf{pk}, m) = h \quad \text{and} \quad \mathsf{H}(\mathsf{pk}^*, m^*) = h.$$

This implies $\mathsf{H}(\mathsf{pk}, m) = \mathsf{H}(\mathsf{pk}^*, m^*)$ which breaks the collision resistance property of the hash function H. □

5 Buffing the Threshold BLS Scheme with Key Prefixing

In this section, we show that the threshold BLS signature scheme can be buffed without increasing the signature size if we assume that the underlying hash function H behaves like a random oracle (which is already assumed for the security proof). The idea is to use key-prefixing and compute the hash value during signature generation as $\mathsf{H}(\mathsf{pk}, m)$ instead of $\mathsf{H}(m)$. Key-prefixing is a well-known measure to prevent key substitution attacks [29] and accomplish tight multi-user security [4]. In fact, using key-prefixing for multi-user security for the BLS scheme has been considered in [28]. Here, we show that it also improves the security in terms of beyond-unforgeability properties.

The BLS signature scheme [8] and its threshold version T-BLS [6] work over pairing-based groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ with generators g_1, g_2, g_T. We assume an efficiently computable, non-degenerate pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ such that

$$e(g_1^x, g_2^y) = e(g_1, g_2)^{xy} \quad \text{and} \quad e(g_1, g_2) \neq 1.$$

We denote by q_T the order of the groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ and assume it is prime throughout. We assume a hash function H mapping strings to group elements $\mathbb{G}_1$. When modeled as a random oracle, the hash function thus returns random group elements. Formally, we assume that there is a parameter generating algorithm $\mathsf{BGGen}(1^\lambda)$ that takes the security parameter λ and outputs a bilinear group $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, g_T, q, e)$ where q is a λ-bit prime.

The threshold variant (T-BLS) of the BLS signature scheme, and the key-prefixing variant T-BLS$_{\mathsf{pk}}$, are displayed in Figure 9. One can view them as the basic BLS scheme with Lagrange interpolation on the exponent.

```
$\underline{\mathsf{Setup}(1^\lambda)}$:
$(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, g_T, q_T) \leftarrow \mathsf{BGGen}(1^\lambda)$ be pairing groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ of prime order $q_T$ with generators $g_1, g_2, g_T$, over field $\mathbb{Z}_{q_T}$, and bilinear pairing operation $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.
Let $\mathsf{H} : \{0,1\}^* \rightarrow \mathbb{G}_1$ be hash function modeled as random oracle.
return $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, g_1, g_2, q_T, \mathsf{H})$.

$\underline{\mathsf{KeyGen}(\mathsf{pp}, n, t)}$:
Let $s(\cdot) \leftarrow\$ \mathbb{Z}_{q_T}[x]$ be a polynomial of degree $t-1$.
Set $\mathsf{pk} \leftarrow g_1^{s(0)}$.
For each $i \in [n]$:
    $\mathsf{sk}_i \leftarrow s(i)$, $\mathsf{pk}_i \leftarrow g_1^{s(i)}$.
return $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{sk}_j)$ to signer $j$ for all $j \in [n]$

$\underline{\mathsf{PartSign}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_i, m)}$:
$\sigma_{m,i} \leftarrow \mathsf{H}([\mathsf{pk},]m)^{\mathsf{sk}_i}$.
return $\sigma_{m,i}$.

$\underline{\mathsf{PartVer}(\mathsf{pp}, \mathsf{pk}_i, m, \sigma_{m,i})}$:
If $e(\mathsf{H}([\mathsf{pk},]m), \mathsf{pk}_i) = e(\sigma_{m,i}, g_2)$:
    return 1.
Else
    return 0.

$\underline{\mathsf{PartComb}(\mathsf{pp}, \mathsf{pk}, \mathsf{S}, m, \{\mathsf{pk}_i, \sigma_{m,i}\}_{i\in\mathsf{S}})}$:
assert $|\mathsf{S}| \geq t$.
For $i \in \mathsf{S}$: assert $\mathsf{PartVer}(\mathsf{pp}, \mathsf{pk}_i, m, \sigma_{m,i}) = 1$.
Let $L_{i,\mathsf{S}}$ be the $i$-th Lagrange coefficients for $\mathsf{S}$.
return $\Sigma_m \leftarrow \prod_{i\in\mathsf{S}} \sigma_{m,i}^{L_{i,\mathsf{S}}}$.

$\underline{\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma_m)}$:
If $e(\mathsf{H}([\mathsf{pk},]m), \mathsf{pk}) = e(\Sigma_m, g_2)$:
    return 1.
Else
    return 0.
```

Figure 9: Threshold BLS scheme; the key-prefixing variant includes the optional argument $[\mathsf{pk},]$ in the hash computation.

Theorem 5.1. *Assuming the* T-BLS *scheme over message space* $\mathcal{M}' = \mathbb{G}_1 \times \mathcal{M}$ *is* UF-CMA*-secure against static corruptions in the random oracle model, then the key-prefixed scheme* $\text{T-BLS}_{\mathsf{pk}}$ *over message space* $\mathcal{M}$ *is* UF-CMA *secure against static corruptions in the random oracle model. Specifically, for any adversary* $\mathcal{A}$ *against* UF-CMA*, we have*

$$\Pr[\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\text{T-BLS}_{\mathsf{pk}},\mathcal{A}} = 1] \leq \Pr[\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\text{T-BLS},\mathcal{A}'} = 1]$$

for algorithm $\mathcal{A}'$ *running in roughly the same time as* $\mathcal{A}$.

Proof. Assume there exists an adversary $\mathcal{A}$ that breaks the unforgeability property of the $\text{T-BLS}_{\mathsf{pk}}$ scheme, we can build $\mathcal{A}'$ that breaks the unforgeability property of the T-BLS scheme. We describe the $\mathcal{A}'$ below.

1. Forward the public parameters pp and verification keys $(\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]})$ from the challenger to $\mathcal{A}$.

2. On receiving corruption queries from $\mathcal{A}$, forward it to the challenger and send the response sk_i from the challenger back to $\mathcal{A}$.

3. On receiving partial signing queries (m, i) from $\mathcal{A}$, prepend pk to the message m and forward (m', i) for $m' = (\mathsf{pk}, m)$ to the challenger. The response from the challenger $\sigma_{m',i}$ is then forwarded to the adversary $\mathcal{A}$. The queries are updated in the $\mathcal{Q}$.

4. The adversary $\mathcal{A}$ submits (m^*, Σ^*). If $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m^*, \Sigma^*) = 1$, send $m'' = (\mathsf{pk}, m^*)$ together with Σ^* to the challenger.

If the challenge satisfies the verifiability condition of T-BLS$_{\mathsf{pk}}$, it will also satisfy the verifiability condition of T-BLS as it is the same check $e(\mathsf{pk}, \mathsf{H}(\mathsf{pk}, m^*)) = e(g_1, \Sigma^*)$. All the partial signing oracle queries are also forwarded by prepending the public key to the messages, thus $|\mathcal{Q}[m''] \cup \mathcal{C}| < t$. Thus, any valid challenge against the $\mathsf{UF\text{-}CMA}$ game of T-BLS$_{\mathsf{pk}}$ is also a valid challenge against the $\mathsf{UF\text{-}CMA}$ game of T-BLS. □

Theorem 5.2. *Assuming the* T-BLS *over* $\mathcal{M}' = \mathbb{G}_1 \times \mathcal{M}$ *scheme is* $\mathsf{RB\text{-}CMA}$ *secure, then the key-prefixed scheme* T-BLS$_{\mathsf{pk}}$ *over* $\mathcal{M}$ *is* $\mathsf{RB\text{-}CMA}$ *secure. Specifically, for any adversary* $\mathcal{A}$ *against* $\mathsf{RB\text{-}CMA}$*, we have*

$$\Pr[\mathsf{Exp}^{\mathsf{RB\text{-}CMA}}_{\text{T-BLS}_{\mathsf{pk}},\mathcal{A}} = 1] \leq \Pr[\mathsf{Exp}^{\mathsf{RB\text{-}CMA}}_{\text{T-BLS},\mathcal{A}'} = 1]$$

for algorithm $\mathcal{A}'$ *running in roughly the same time as* $\mathcal{A}$.

Proof. The robustness property follows from the robustness property of the underlying scheme T-BLS. There are two winning cases for the adversary, (1) honestly generated partial signatures that do not satisfy the verification check, and (2) partial signatures that pass verification, but the combined signature does not satisfy the verification check.

We analyze the first case. Consider an adversary that can provide (m, i) where the honestly generated partial signature does not verify, then (pk, m, i) would be a winning scenario for an adversary $\mathcal{A}'$ against T-BLS with larger message space $\mathcal{M}' = \mathbb{G}_1 \times \mathcal{M}$. In the second case, if an adversary can submit $\mathsf{S}, m, \{\sigma_i\}_{i \in \mathsf{S}}$, the partial signatures pass the verification, but $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma)$ is not satisfied, then $e(\mathsf{pk}, \mathsf{H}(\mathsf{pk}, m)) \neq e(g_1, \Sigma)$. Then, $\mathsf{S}, \mathsf{pk}, m, \{\sigma_i\}_{i \in \mathsf{S}}$ also does not pass the verification of the T-BLS scheme. Thus, whenever the T-BLS$_{\mathsf{pk}}$ adversary breaks the robustness property, we could break the robustness property of T-BLS. □

We provide below the $\mathsf{M\text{-}S\text{-}UEO}$ proof of the T-BLS$_{\mathsf{pk}}$ scheme. From the Proposition 3.6 and Proposition 3.7, we know that $\mathsf{M\text{-}S\text{-}UEO}$ and $\mathsf{RB\text{-}CMA}$ together imply $\mathsf{S\text{-}CEO}$, $\mathsf{S\text{-}DEO}$ and $\mathsf{S\text{-}UEO}$. Thus, we do not provide explicit proofs for $\mathsf{S\text{-}CEO}$, $\mathsf{S\text{-}DEO}$ and $\mathsf{S\text{-}UEO}$ security.

Theorem 5.3. *In the random oracle model,* T-BLS$_{\mathsf{pk}}$ *is* $\mathsf{M\text{-}S\text{-}UEO}$*. Specifically for any adversary* $\mathcal{A}$ *against* $\mathsf{M\text{-}S\text{-}UEO}$*, making at most* q_{H} *random oracle queries, we have*

$$\Pr\Big[\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\text{T-BLS}_{\mathsf{pk}},\mathcal{A}}(1^\lambda, n, t) = 1\Big] \leq \frac{(q_{\mathsf{H}} + 2)^2}{q_T},$$

where q_T *is the prime order of the group* $\mathbb{G}_T$.

Proof. Assume first that the adversary $\mathcal{A}$, in its attack against the $\mathsf{M\text{-}S\text{-}UEO}$ property, does not query the random oracle about $\mathsf{H}(\mathsf{pk}, m)$ or $\mathsf{H}(\mathsf{pk}', m')$ before outputting its attempt $(m, m', \mathsf{pk}, \mathsf{pk}', \Sigma)$. Then, we can patch this by letting $\mathcal{A}$ make this query right before creating the output, increasing the number of random oracle queries by at most 2. So from now on, we assume that $\mathcal{A}$ at some point makes these random oracle queries in some of its at most $q_{\mathsf{H}} + 2$ queries. We may also assume that $\mathcal{A}$ never queries the random oracle about the same value twice.

Assume now that $\mathcal{A}$ at some point makes a random oracle query about $\mathsf{H}(\mathsf{pk}', m')$ for some pk'. Note that we can recover pk' from the query (or identify a false encoding) by assumption. If this pair (pk', m') is to be used in $\mathcal{A}$'s final output, including a yet-to-be-determined signature Σ, it must satisfy

$$e(\Sigma, g_2) = e(\mathsf{H}(\mathsf{pk}', m'), \mathsf{pk}').$$

Any other of the at most $q_{\mathsf{H}} + 1$ previous hash queries pk, m could be used for the same signature value Σ, it must also satisfy

$$e(\Sigma, g_2) = e(\mathsf{H}(\mathsf{pk}, m), \mathsf{pk}).$$

But then the outputs must satisfy:

$$e(\mathsf{H}(\mathsf{pk}, m), \mathsf{pk}) = e(\mathsf{H}(\mathsf{pk}', m'), \mathsf{pk}').$$

When $\mathcal{A}$ makes the query about pk', m', it is the first time by assumption, and the hash value is mapped to a random group element $\mathsf{H}(\mathsf{pk}', m') = g_1^h$ for random h. This group element is independent of pk' and all other previous data. Hence, the probability that it matches any of the previous values

$$e(\mathsf{H}(\mathsf{pk}, m), \mathsf{pk}) = e(\mathsf{H}(\mathsf{pk}', m'), \mathsf{pk}') = e(g_1^h, \mathsf{pk}') = e(g_1, \mathsf{pk}')^h$$

is at most $(q_{\mathsf{H}} + 2)/q_T$. Summing over at most $q_{\mathsf{H}} + 2$ hash queries implies that the probability the adversary only finds a pair of suitable hash inputs is bounded from above by $(q_{\mathsf{H}} + 2)^2/q_T$. Since we assumed that $\mathcal{A}$ must query H about the output values, this also provides an upper bound on the probability that $\mathcal{A}$ wins the M-S-UEO experiment. □

Theorem 5.4. *In the random oracle model,* T-BLS$_{\mathsf{pk}}$ *is* MBS*. Specifically for any adversary $\mathcal{A}$ against* MBS*, making at most q_{H} random oracle queries, we have*

$$\Pr\left[\mathsf{Exp}^{\mathsf{MBS}}_{\text{T-BLS}_{\mathsf{pk}},\mathcal{A}}(1^\lambda, n, t) = 1\right] \leq \frac{(q_{\mathsf{H}} + 2)^2}{q_T},$$

where q_T is the prime order of the group $\mathbb{G}_T$.

Proof. The proof is almost identical to M-S-UEO security proof (Theorem 5.3). This time, the adversary tries to find hash inputs pk, m, pk, m' for the same public key pk but different messages $m \neq m'$ such that the pairing operation maps them to the same value

$$e(\mathsf{H}(\mathsf{pk}, m), \mathsf{pk}) = e(\Sigma, g_2) = e(\mathsf{H}(\mathsf{pk}, m'), \mathsf{pk})$$

in the verification. It follows as in the M-S-UEO setting that the probability of finding such pairs is at most $(q_{\mathsf{H}} + 2)^2/q_T$. □

We are unaware if the (unkeyed) version of the threshold BLS scheme can be shown to achieve the BUFF properties. It is clear that it already achieves S-CEO, because the adversary cannot find $\mathsf{pk}^* \neq \mathsf{pk}$ such that

$$e(\mathsf{H}(m), \mathsf{pk}) = e(\Sigma_m, g_2) \quad \text{and} \quad e(\mathsf{H}(m), \mathsf{pk}^*) = e(\Sigma_m, g_2)$$

for the same message and the same signature value Σ_m, as long as group membership of public keys can be checked efficiently. We are not aware of positive or negative results concerning the other properties.

6 BUFF Threshold Signatures from BUFF Signatures

Many threshold signature schemes are based upon an ordinary signature scheme and "thresholdize" the computation of such an ordinary signature. We discuss here that such threshold schemes immediately inherit some of the BUFF properties of the underlying signature scheme. We then continue to show that some known protocols like TRaccoon [15], FROST [27], and the universal thresholdizer of Boneh et al. [7] can be subsumed under our paradigm.

6.1 Inheriting BUFF Properties

Consider a threshold signature $\mathsf{TS}_{\mathsf{Sig}}$ scheme having the following structure with an underlying signature scheme $\mathsf{Sig} = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Sign}', \mathsf{Ver}')$ satisfying the BUFF properties. We assume that the key generation and verification of the threshold scheme are the same as those of the signature scheme. The additional steps, such as computations of additional public parameters, the signing shares, the partial signatures, the combination of partial signatures, as well as the verification of partial shares and extra verification steps, are carried out by some algorithms $\mathsf{Algo}_i, i \in [6]$. The partial verification algorithm $\mathsf{PartVer}$ as well as the algorithm Algo_4 is optional. These algorithms can be interactive, probabilistic polynomial time algorithms. The description of $\mathsf{TS}_{\mathsf{Sig}}$ is provided in Figure 10.

We show below that if the underlying signature scheme Sig satisfies the $\mathsf{M\text{-}S\text{-}UEO}, \mathsf{MBS}$ properties, then the threshold signature scheme $\mathsf{TS}_{\mathsf{Sig}}$ based on Sig also satisfies the BUFF properties $\mathsf{M\text{-}S\text{-}UEO}, \mathsf{MBS}$. The advantage of these properties over, say, $\mathsf{S\text{-}CEO}$ is that they are based on "public" data chosen by the adversary. The relevant algorithms Algo_1 and Algo_6 only require public information. We are not aware if the same is true for the properties $\mathsf{S\text{-}CEO}$, $\mathsf{S\text{-}DEO}$, and $\mathsf{S\text{-}UEO}$ because in these cases, the adversary gets oracle access to functions with access to secrets, involving algorithm Algo_3. Recall, however, that $\mathsf{M\text{-}S\text{-}UEO}$ implies the further properties if we assume that the threshold scheme is also robust.

Theorem 6.1. *Assuming the underlying signature scheme* Sig *achieves* $\mathsf{M\text{-}S\text{-}UEO}$ *security, then,* $\mathsf{TS}_{\mathsf{Sig}}$ *in Figure 10 also achieves* $\mathsf{M\text{-}S\text{-}UEO}$*. Specifically, for any adversary* $\mathcal{A}$ *against* $\mathsf{M\text{-}S\text{-}UEO}$*, we have*

$$\Pr[\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{TS}_{\mathsf{Sig}},\mathcal{A}} = 1] \leq \Pr[\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{Sig},\mathcal{A}'} = 1].$$

for algorithm $\mathcal{A}'$ *running in roughly the same time as* $\mathcal{A}$*.*

Proof. We prove the above theorem through a reduction to the $\mathsf{M\text{-}S\text{-}UEO}$ security game of the underlying signature scheme Sig. Assume that there exists an adversary $\mathcal{A}$ that can break the $\mathsf{M\text{-}S\text{-}UEO}$ security of the $\mathsf{TS}_{\mathsf{Sig}}$ threshold scheme. Then we can build $\mathcal{A}'$ that can break the security of the signature scheme Sig. We provide the description of $\mathcal{A}'$.

1. On receiving pp' from the Sig challenger, it runs $\mathsf{pp}'' \leftarrow \mathsf{Algo}_1(1^\lambda, \mathsf{pp}')$ and sends $\mathsf{pp} = (\mathsf{pp}', \mathsf{pp}'')$ to $\mathcal{A}$.
2. On receiving $(m, m^*, \mathsf{pk}, \mathsf{pk}^*, \Sigma)$ from $\mathcal{A}$, forward it to the challenger if $\mathsf{pk} \neq \mathsf{pk}^*$, $\mathsf{Ver}'(\mathsf{pp}', \mathsf{pk}, m, \Sigma) = 1$, $\mathsf{Algo}_6(\mathsf{pp}, \mathsf{pk}, m, \Sigma) = 1$, $\mathsf{Ver}'(\mathsf{pp}', \mathsf{pk}^*, m^*, \Sigma) = 1$, and $\mathsf{Algo}_6(\mathsf{pp}, \mathsf{pk}^*, m^*, \Sigma) = 1$. Note that all these checks are possible given the public data.

The reduction $\mathcal{A}'$ wins $\mathsf{M\text{-}S\text{-}UEO}$ game whenever $\mathcal{A}$ wins the $\mathsf{M\text{-}S\text{-}UEO}$ game. Thus,

$$\Pr[\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{TS}_{\mathsf{Sig}},\mathcal{A}} = 1] \leq \Pr[\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{Sig},\mathcal{A}'} = 1].$$

This proves the theorem. □

Theorem 6.2. *Assuming the underlying signature scheme* Sig *achieves* MBS *security, then, the* $\mathsf{TS}_{\mathsf{Sig}}$ *signature scheme in Figure 10 also achieves* MBS*. Specifically, for any adversary* $\mathcal{A}$ *against* MBS*, we have*

$$\Pr[\mathsf{Exp}^{\mathsf{MBS}}_{\mathsf{TS}_{\mathsf{Sig}},\mathcal{A}} = 1] \leq \Pr[\mathsf{Exp}^{\mathsf{MBS}}_{\mathsf{Sig},\mathcal{A}'} = 1].$$

for algorithm $\mathcal{A}'$ *running in roughly the same time as* $\mathcal{A}$*.*

```
$\underline{\mathsf{Setup}(1^\lambda):}$
$\mathsf{pp}' \leftarrow\$ \mathsf{Setup}'(1^\lambda)$.
$\mathsf{pp}'' \leftarrow\$ \mathsf{Algo}_1(1^\lambda, \mathsf{pp}')$.
**return** $\mathsf{pp} = (\mathsf{pp}', \mathsf{pp}'')$.

$\underline{\mathsf{KeyGen}(\mathsf{pp}, n, t):}$
$(\mathsf{pk}', \mathsf{sk}') \leftarrow\$ \mathsf{KeyGen}'(\mathsf{pp}')$.
$(\{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{sk}_i\}_{i\in[n]}) \leftarrow\$ \mathsf{Algo}_2(\mathsf{pp}, \mathsf{pk}', \mathsf{sk}', n, t)$.
**return** $(\mathsf{pk} = \mathsf{pk}', \{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{sk}_i\}_{i\in[n]})$.

$\underline{\mathsf{PartSign}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_i, m):}$
$\sigma_{m,i} \leftarrow\$ \mathsf{Algo}_3(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_i, m)$.
**return** $\sigma_{m,i}$.

$\underline{\mathsf{PartVer}(\mathsf{pp}, \mathsf{pk}, m, \mathsf{pk}_i, \sigma_{m,i}):}$
$b \leftarrow\$ \mathsf{Algo}_4(\mathsf{pp}, \mathsf{pk}, m, \mathsf{pk}_i, \sigma_{m,i})$.
**return** $b$.

$\underline{\mathsf{PartComb}(\mathsf{pp}, \mathsf{pk}, \mathsf{S}, \{\mathsf{pk}_i, \sigma_{m,i}\}_{i\in\mathsf{S}}):}$
$\Sigma_m \leftarrow\$ \mathsf{Algo}_5(\mathsf{pp}, \mathsf{pk}, \mathsf{S}, \{\mathsf{pk}_i, \sigma_{m,i}\}_{i\in\mathsf{S}})$.
**return** $\Sigma_m$.

$\underline{\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma_m):}$
if $\mathsf{Ver}'(\mathsf{pp}', \mathsf{pk}', m, \Sigma_m) = 1$ and $\mathsf{Algo}_6(\mathsf{pp}, \mathsf{pk}, m, \Sigma_m) = 1$, **return** $1$.
else, **return** $0$.
```

Figure 10: Description of a threshold signature scheme $\mathsf{TS}_{\mathsf{Sig}}$ based on signature scheme Sig.

Proof. This follows the same way as the one described in the previous proof. Assume there exists an adversary $\mathcal{A}$ that can break the MBS property of $\mathsf{TS}_{\mathsf{Sig}}$, then we can build $\mathcal{A}'$ that breaks the MBS property of Sig using $\mathcal{A}$. We provide the description of $\mathcal{A}'$ below.

1. On receiving pp' from the Sig challenger, runs $\mathsf{pp}'' \leftarrow \mathsf{Algo}_1(1^\lambda, \mathsf{pp}')$ and sends $\mathsf{pp} = (\mathsf{pp}', \mathsf{pp}'')$ to $\mathcal{A}$.

2. On receiving $(m, m^*, \mathsf{pk}, \Sigma)$ from $\mathcal{A}$, forward it to the challenger if $m \neq m^*$, $\mathsf{Ver}'(\mathsf{pp}', \mathsf{pk}, m, \Sigma) = 1$, $\mathsf{Algo}_6(\mathsf{pp}, \mathsf{pk}, m, \Sigma) = 1$, $\mathsf{Ver}'(\mathsf{pp}', \mathsf{pk}, m^*, \Sigma) = 1$, and $\mathsf{Algo}_6(\mathsf{pp}, \mathsf{pk}, m^*, \Sigma) = 1$.

The reduction $\mathcal{A}'$ wins whenever $\mathcal{A}$ wins. □

We next analyze the M-S-UEO and MBS properties of some of the recent threshold signature schemes.

6.2 Analysis of TRaccoon [15]

The threshold signature scheme TRaccoon is a practical three-round lattice-based threshold signature that can be viewed as a threshold version of the standard signature scheme Raccoon[16].

We analyze the M-S-UEO and MBS properties of the TRaccoon scheme and provide the following result.

Proposition 6.3. *The threshold signature scheme* TRaccoon *is* M-S-UEO *and* MBS *secure, assuming the hash functions are collision-resistant and non-malleable.*

Proof. The threshold signature scheme TRaccoon could be viewed as the thresholdized version of Raccoon [16] and follows the structure we provide in Figure 10. The NIST submission of Raccoon [14] analyzes and proves that Raccoon provides M-S-UEO and MBS assuming that the hash functions used are collision-resistant and non-malleable in Section 4.4 of the technical report. Considering the above facts, and Theorems 6.1 and 6.2, TRaccoon provides M-S-UEO and MBS security. □

6.3 Analysis of FROST [27]

The FROST (Flexible Round-Optimized Schnorr Threshold Signatures) scheme is a thresholdized version of the key-prefixed Schnorr [33] signature scheme. It has a message-independent preprocessing phase before the partial signing phase.

Proposition 6.4. *Assuming the key-prefixed version of* Schnorr *signature scheme satisfies* M-S-UEO *and* MBS*, then the threshold signature scheme* FROST *[27] provides* M-S-UEO *and* MBS *security in the random oracle model.*

Proof. The FROST signature scheme is a thresholdized version of the Schnorr signature scheme and follows the structure provided in Figure 10. Thus, from Theorems 6.1 and 6.2, and assuming Schnorr is M-S-UEO and MBS, FROST is M-S-UEO and MBS. □

To complete the picture, we describe the key-prefixed version of Schnorr in Figure 11 and then prove that it provides M-S-UEO and MBS security.

```
Setup($1^\lambda$):
  $(\mathbb{G}, \mathbb{Z}_q, q, g) \leftarrow \mathsf{GGen}(1^\lambda)$.
  Let $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_q$ be a
  hash function modeled as random oracle.
  return $\mathsf{pp} = (\mathbb{G}, \mathbb{Z}_q, q, g, \mathsf{H})$

Sign(pp, sk) :
  $k \leftarrow\!\!\$\ \mathbb{Z}_q, R \leftarrow g^k$.
  $c \leftarrow \mathsf{H}(\mathsf{pk}, m, R)$.
  $z \leftarrow k + \mathsf{sk} \cdot c$.
  return $\sigma = (R, z)$.

KeyGen(pp):
  Let $\mathsf{sk} \leftarrow\!\!\$\ \mathbb{Z}_q$
  Set $\mathsf{pk} \leftarrow g^{\mathsf{sk}}$
  return $(\mathsf{pk}, \mathsf{sk})$

Ver($\mathsf{pp}, \mathsf{pk}, m, \sigma_m,$):
  Parse $\sigma$ as $(R, z)$.
  If $R \cdot \mathsf{pk}^{\mathsf{H}(\mathsf{pk}, m, R)} = g^z$:
      return 1
  Else
      return 0
```

Figure 11: Construction of key-prefixed Schnorr signature scheme.

Theorem 6.5. *In the random oracle model,* Schnorr *is* M-S-UEO*. Specifically for any adversary $\mathcal{A}$ against* M-S-UEO*, making at most q_{H} random oracle queries, we have*

$$\Pr\Big[\mathsf{Exp}^{\mathsf{M\text{-}S\text{-}UEO}}_{\mathsf{Schnorr},\mathcal{A}}(1^\lambda) = 1\Big] \leq \frac{(q_{\mathsf{H}} + 2)^2}{q},$$

where q is the order of the group $\mathbb{G}$.

Proof. The proof follows the same way as the proof of Theorem 5.3. Assume that the adversary $\mathcal{A}$ in the $\mathsf{M\text{-}S\text{-}UEO}$ game does not query the random oracle on $\mathsf{H}(\mathsf{pk}, m, R)$ or $\mathsf{H}(\mathsf{pk}', m', R)$ before outputting its attempt $(m, m', \mathsf{pk}, \mathsf{pk}', (R, z))$. Then, we can patch this by letting $\mathcal{A}$ make this query right before creating the output, increasing the number of random oracle queries by at most 2. We assume that $\mathcal{A}$ makes these random oracle queries at some point where the total number of random oracle queries is at most $q_{\mathsf{H}} + 2$ queries. We may also assume that $\mathcal{A}$ never queries the random oracle about the same value twice.

Assume that $\mathcal{A}$ at some point makes a random oracle query on $\mathsf{H}(\mathsf{pk}', m', R)$ for some pk' and R. Note that we can recover pk' and R from the query by assumption. If this tuple (pk', m') is to be used in $\mathcal{A}$'s final output, along with signature (R, z), it must satisfy:

$$R \cdot \mathsf{pk}'^{\mathsf{H}(\mathsf{pk}', m', R)} = g^z.$$

Any other of the at most $q_{\mathsf{H}}+1$ previous hash queries (pk, m, R) could be used for the same signature value Σ, it must also satisfy

$$R \cdot \mathsf{pk}^{\mathsf{H}(\mathsf{pk}, m, R)} = g^z.$$

But then the outputs must satisfy

$$\mathsf{pk}^{\mathsf{H}(\mathsf{pk}, m, R)} = \mathsf{pk}'^{\mathsf{H}(\mathsf{pk}', m', R)}.$$

When $\mathcal{A}$ makes a random oracle query on (pk', m', R) the first time by assumption, and the hash value is mapped to a random group element $\mathsf{H}(\mathsf{pk}', m', R) = h$ for random $h \in \mathbb{Z}_q$. This field element is independent of pk' and all other previous data. Hence, the probability that it matches any of the previous values

$$\mathsf{pk}^{\mathsf{H}(\mathsf{pk}, m, R)} = \mathsf{pk}'^{\mathsf{H}(\mathsf{pk}', m', R)} = h$$

is at most $(q_{\mathsf{H}} + 2)/q$. Summing over at most $q_{\mathsf{H}} + 2$ hash queries implies that the probability the adversary only finds a pair of suitable hash inputs is bounded from above by $(q_{\mathsf{H}} + 2)^2/q$. Because we assumed that $\mathcal{A}$ must query H about the output values, this also provides an upper bound on the probability that $\mathcal{A}$ wins the $\mathsf{M\text{-}S\text{-}UEO}$ experiment. □

Theorem 6.6. *In the random oracle model,* Schnorr *is* MBS *secure. Specifically for any adversary* $\mathcal{A}$ *against* MBS*, making at most* q_{H} *random oracle queries, we have*

$$\Pr\left[\mathsf{Exp}^{\mathsf{MBS}}_{\mathsf{Schnorr},\mathcal{A}}(1^\lambda) = 1\right] \leq \frac{(q_{\mathsf{H}} + 2)^2}{q},$$

where q *is the order of the group* $\mathbb{G}$*.*

Proof. The proof follows the same way as the proof of $\mathsf{M\text{-}S\text{-}UEO}$. Here, the adversary tries to find hash inputs $(m, m', \mathsf{pk}, (R, z))$ such that

$$\mathsf{pk}^{\mathsf{H}(\mathsf{pk}, m, R)} = \mathsf{pk}^{\mathsf{H}(\mathsf{pk}, m', R)} = h$$

where $m \neq m'$. It follows as in the $\mathsf{M\text{-}S\text{-}UEO}$ setting where the probability of finding such pairs is at most $(q_{\mathsf{H}} + 2)^2/q$. □

6.4 Analysis of TS by Boneh et al. [7]

The work of Boneh et al. provides a general approach for adding a threshold functionality to a large class of (non-threshold) cryptographic schemes. In particular, they introduced the concept of *universal thresholdizer* which can be applied to a non-threshold lattice signature system Sig to obtain a single-round threshold signature scheme TS. We analyze the M-S-UEO and MBS properties of this signature scheme and provide the following results.

Proposition 6.7. *If the underlying signature scheme* Sig *used in the threshold signature construction of [7] satisfies* M-S-UEO *and* MBS *properties, then the* TS *scheme is also* M-S-UEO *and* MBS *secure.*

Proof. The universal thresholdizer is used to thresholdize Sig to construct TS. The signing key of the signature scheme Sig is given as an input to the universal thresholdizer in their construction. The verification of TS is the same as that of Sig. So, the TS follows the structure provided in Figure 6.1. Thus, from theorems 6.1 and 6.2, and assuming Sig is M-S-UEO and MBS, TS is M-S-UEO and MBS. □

7 Conclusion

We explored advanced security notions for threshold signature schemes, focusing on Beyond UnForgeability Features (BUFF). Our contributions include a generic compiler for transforming threshold signature schemes to achieve exclusive ownership and message-bound signature properties with minimal overhead, as well as a modified threshold BLS scheme achieving BUFF properties without increasing the signature size. Future work could focus on examining the S-CEO, S-DEO, and robustness properties in other relevant threshold signature schemes.

Concerning the BUFF security of existing schemes, we have already mentioned that the original ("prefixing-free") threshold BLS scheme already supports S-CEO directly, but that we are unaware if it satisfies further BUFF properties. Key-prefixing is, on the other hand, an easy patch to ensure all properties. It would also be interesting to see if the non-robust scheme FROST also achieves the S-UEO property. Our general result states that it is M-S-UEO based on the security of the underlying Schnorr signature scheme. The lack of robustness of FROST, however, does not allow us to draw the conclusion that this implies S-UEO security. For TRaccoon it is currently unclear if the scheme is also robust [15]. Hence, we neither derive the other properties instantaneously but would need to prove them from scratch——or show robustness of TRaccoon.

Acknowledgements

We thank the anonymous reviewers of PKC 2025 for their valuable comments. Marc Fischlin has carried out this work under grant Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1119 – 236615297. This work was partially supported by A. Mitrokotsa's Stepping Stone grant funded by the Univ. of St. Gallen.

References

[1] Thomas Aulbach, Samed Düzlü, Michael Meyer, Patrick Struck, and Maximiliane Weishäupl. Hash your keys before signing - BUFF security of the additional NIST PQC signatures. In Markku-Juhani O. Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography* -

15th International Workshop, PQCrypto 2024, Oxford, UK, June 12-14, 2024, Proceedings, Part II, volume 14772 of *Lecture Notes in Computer Science*, pages 301–335. Springer, 2024.

[2] Renas Bacho, Julian Loss, Stefano Tessaro, Benedikt Wagner, and Chenzhi Zhu. Twinkle: Threshold signatures from DDH with full adaptive security. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I*, volume 14651 of *Lecture Notes in Computer Science*, pages 429–459. Springer, 2024.

[3] Mihir Bellare, Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. Better than advertised security for non-interactive threshold signatures. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV*, volume 13510 of *Lecture Notes in Computer Science*, pages 517–550. Springer, 2022.

[4] Daniel J. Bernstein. Multi-user schnorr security, revisited. *IACR Cryptol. ePrint Arch.*, page 996, 2015.

[5] Simon Blake-Wilson and Alfred Menezes. Unknown key-share attacks on the station-to-station (STS) protocol. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC '99, Kamakura, Japan, March 1-3, 1999, Proceedings*, volume 1560 of *Lecture Notes in Computer Science*, pages 154–170. Springer, 1999.

[6] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In Yvo Desmedt, editor, *Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2003.

[7] Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 565–596, Cham, 2018. Springer International Publishing.

[8] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. *J. Cryptol.*, 17(4):297–319, 2004.

[9] Cecilia Boschini, Darya Kaviani, Russell W. F. Lai, Giulio Malavolta, Akira Takahashi, and Mehdi Tibouchi. Ringtail: Practical two-round threshold signatures from learning with errors. Cryptology ePrint Archive, Paper 2024/1113, 2024.

[10] Jacqueline Brendel, Cas Cremers, Dennis Jackson, and Mang Zhao. The provable security of ed25519: Theory and practice. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 1659–1676. IEEE, 2021.

[11] Cas Cremers, Samed Düzlü, Rune Fiedler, Marc Fischlin, and Christian Janson. Buffing signature schemes beyond unforgeability and the case of post-quantum signatures. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 1696–1714. IEEE, 2021.

[12] Sourav Das, Philippe Camacho, Zhuolun Xiang, Javier Nieto, Benedikt Bünz, and Ling Ren. Threshold signatures from inner product argument: Succinct, weighted, and multi-threshold. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, pages 356–370. ACM, 2023.

[13] Sourav Das and Ling Ren. Adaptively secure BLS threshold signatures from DDH and co-cdh. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VII*, volume 14926 of *Lecture Notes in Computer Science*, pages 251–284. Springer, 2024.

[14] Rafaël del Pino, Thomas Espitau, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, Mélissa Rossi, and Markku-Juhani Saarinen. Raccoon. Technical report, National Institute of Standards and Technology, 2023. Available at `https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures`.

[15] Rafael del Pino, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, and Markku-Juhani Saarinen. Threshold raccoon: Practical threshold signatures from standard lattice assumptions. In *Advances in Cryptology – EUROCRYPT 2024: 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26–30, 2024, Proceedings, Part II*, page 219–248, Berlin, Heidelberg, 2024. Springer-Verlag.

[16] Rafaël del Pino, Shuichi Katsumata, Thomas Prest, and Mélissa Rossi. Raccoon: A masking-friendly signature proven in the probing model. In *Advances in Cryptology – CRYPTO 2024: 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part I*, page 409–444, Berlin, Heidelberg, 2024. Springer-Verlag.

[17] Yvo Desmedt. Society and group oriented cryptography: A new concept. In Carl Pomerance, editor, *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, volume 293 of *Lecture Notes in Computer Science*, pages 120–127. Springer, 1987.

[18] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer, 1989.

[19] Jack Doerner, Yashvanth Kondi, Eysa Lee, and Abhi Shelat. Threshold ECDSA in three rounds. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*, pages 3053–3071. IEEE, 2024.

[20] Jelle Don, Serge Fehr, Yu-Hsuan Huang, Jyun-Jie Liao, and Patrick Struck. Hide-and-seek and the non-resignability of the BUFF transform. Cryptology ePrint Archive, Paper 2024/793, 2024.

[21] Jelle Don, Serge Fehr, Yu-Hsuan Huang, and Patrick Struck. On the (in)security of the BUFF transform. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part I*, volume 14920 of *Lecture Notes in Computer Science*, pages 246–275. Springer, 2024.

[22] Samed Düzlü, Rune Fiedler, and Marc Fischlin. Buffing FALCON without increasing the signature size. *IACR Cryptol. ePrint Arch.*, page 710, 2024.

[23] Thomas Espitau, Guilhem Niot, and Thomas Prest. Flood and submerse: Distributed key generation and robust threshold signature from lattices. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VII*, volume 14926 of *Lecture Notes in Computer Science*, pages 425–458. Springer, 2024.

[24] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[25] Dennis Jackson, Cas Cremers, Katriel Cohn-Gordon, and Ralf Sasse. Seems legit: Automated analysis of subtle attacks on protocols that use signatures. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 2165–2180. ACM, 2019.

[26] Shuichi Katsumata, Michael Reichle, and Kaoru Takemure. Adaptively secure 5 round threshold signatures from MLWE/MSIS and DL with rewinding. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VII*, volume 14926 of *Lecture Notes in Computer Science*, pages 459–491. Springer, 2024.

[27] Chelsea Komlo and Ian Goldberg. Frost: Flexible round-optimized schnorr threshold signatures. In *Selected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers*, page 34–65, Berlin, Heidelberg, 2020. Springer-Verlag.

[28] Marie-Sarah Lacharité. Security of BLS and BGLS signatures in a multi-user setting. *Cryptogr. Commun.*, 10(1):41–58, 2018.

[29] Alfred Menezes and Nigel P. Smart. Security of signature schemes in a multi-user setting. *Des. Codes Cryptogr.*, 33(3):261–274, 2004.

[30] National Institute of Standards and Technology. Call for additional digital signature schemes for the post-quantum cryptography standardization process. Technical report, National Institute of Standards and Technology, 2022. Available at `https://csrc.nist.gov/Projects/pqc-dig-sig/standardization/call-for-proposals`.

[31] Thomas Pornin and Julien P. Stern. Digital signatures do not guarantee exclusive ownership. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings*, volume 3531 of *Lecture Notes in Computer Science*, pages 138–150, 2005.

[32] Phillip Rogaway. Formalizing human ignorance. In Phong Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 2006*, pages 211–228, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[33] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '89, page 239–252, Berlin, Heidelberg, 1989. Springer-Verlag.

[34] Haiyang Xue, Man Ho Au, Mengling Liu, Kwan Yin Chan, Handong Cui, Xiang Xie, Tsz Hon Yuen, and Chengru Zhang. Efficient multiplicative-to-additive function from joye-libert cryptosystem and its application to threshold ECDSA. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, pages 2974–2988. ACM, 2023.

A Further Relationships between BUFF Notions

A.1 Separating Extra-Strong from Strong Exclusive Ownership Notions

As mentioned before, all extra-strong versions ES-CEO, ES-DEO, and ES-UEO versions imply their strong counterparts S-CEO, S-DEO, and S-UEO. We show here that the converse is not true in general. We start with a separating example of a scheme TS that is S-CEO and ES-DEO but not ES-CEO. Necessarily, the scheme cannot be ES-UEO, and hence neither simultaneously M-S-UEO and RB-CMA, because the former would imply ES-CEO and ES-DEO by Proposition 3.6, and the latter would imply ES-UEO by Proposition 3.7. The scheme preserves unforgeability and M-S-UEO if the underlying scheme TS' has these properties.

The main idea is to prepend a bit $b = 0$ to each (honest) partial signature and to let the combine algorithm prepend the logical AND over all such bits of the partial signatures and to prepend this AND to the combined signatures. Then, any honest contribution of a partial signature (as in S-CEO) enforces a 0-bit in the combined signature, whereas contributions from exclusively dishonest parties (as in ES-CEO) can easily make this bit 1. The final step is to allow the transfer to a public key pk^* of such signatures starting with 1. To ensure ES-DEO, we simply include the message in the signature. More formally, consider an arbitrary threshold scheme TS' (where KeyGen' for given parameters pp always outputs public keys pk' of a fixed length and $\mathsf{PartComb}'$ for a given pk' always outputs signatures Σ'_m of a fixed length) and modify it to scheme TS as follows:

1. $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: Outputs $\mathsf{Setup}'(1^\lambda) \to \mathsf{pp}'$.

2. $\mathsf{KeyGen}(\mathsf{pp}, n, t) \to (\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{sk}_i\}_{i\in[n]})$: Computes the underlying keys $\mathsf{KeyGen}'(\mathsf{pp}', n, t) \to (\mathsf{pk}', \{\mathsf{pk}'_i\}_{i\in[n]}, \{\mathsf{sk}'_i\}_{i\in[n]})$ and returns $\mathsf{pk} = 0||\mathsf{pk}'$, as well as $\{\mathsf{pk}_i\}_{i\in[n]}$, $\{\mathsf{sk}_i\}_{i\in[n]}$ for $\mathsf{pk}_i = \mathsf{pk}'_i$ and $\mathsf{sk}_i = \mathsf{sk}'_i$.

3. $\mathsf{PartSign}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_i, m) \to \sigma_{m,i}$. Runs $\mathsf{PartSign}'(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_i, m) \to \sigma'_{m,i}$ and outputs $\sigma_{m,i} = 0||\sigma'_{m,i}$.

4. $\mathsf{PartVer}(\mathsf{pp}, \mathsf{pk}, m, \mathsf{pk}_i, \sigma_{m,i}) \to 0/1$. Runs $\mathsf{PartVer}'(\mathsf{pp}, \mathsf{pk}', m, \mathsf{pk}_i, \sigma'_{m,i}) \to 0/1$ for $\mathsf{pk}', \sigma'_{m,i}$ with $\mathsf{pk} = b||\mathsf{pk}'$ and $\sigma_{m,i} = b_i||\sigma'_{m,i}$.

5. $\mathsf{PartComb}(\mathsf{pp}, \mathsf{pk}, \mathsf{S}, m, \{\mathsf{pk}_i, \sigma_{m,i}\}_{i\in\mathsf{S}}) \to \Sigma_m$: Runs $\mathsf{PartComb}'(\mathsf{pp}, \mathsf{pk}', \mathsf{S}, m, \{\mathsf{pk}_i, \sigma'_{m,i}\}_{i\in\mathsf{S}}) \to \Sigma'_m$ for $\mathsf{pk}', \sigma'_{m,i}$ with $\mathsf{pk} = b||\mathsf{pk}'$ and $\sigma_{m,i} = b_i||\sigma'_{m,i}$. Returns $\Sigma_m = (\bigwedge_{i\in\mathsf{S}} b_i)||\mathsf{pk}'||\Sigma'_m||m$.

6. $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma_m) \to 0/1$: Accepts if $\mathsf{Ver}'(\mathsf{pp}, \mathsf{pk}', m, \Sigma'_m) \to 0/1$ for $\mathsf{pk} = b||\mathsf{pk}'$ $\Sigma_m = b||\mathsf{pk}'||\Sigma'_m||m$. (If Σ_m does not parse correctly with leading values $b||\mathsf{pk}'$ and Σ'_m of fixed length, followed by m, then Ver also returns 0.)

Via the equivalence of UEO to CEO and DEO (both in the strong and extra-strong versions), we immediately obtain a separation for the UEO case as part of our result for CEO:

Proposition A.1. *If* TS' *is a correct and* UF-CMA *and* M-S-UEO *threshold signature scheme, then* TS *as defined above is a correct and* UF-CMA *and* M-S-UEO *threshold signature scheme which satisfies* S-CEO *and* ES-DEO *(and thus* S-UEO*) but not* ES-CEO *(and thus not* ES-UEO*).*

Proof. Correctness and unforgeability of TS follow straightforwardly from the correctness properties of TS' because genuinely combined partial signatures (all with leading bit $b_i = 0$) result in a valid signature Σ'_m which verify under an honestly generated key pk with a leading 0-bit. Unforgeability also holds because one can give a straightforward reduction for genuine public keys of the form $\mathsf{pk} = 0||\mathsf{pk}'$.

We first argue that M-S-UEO holds. Note that a valid signature Σ_m for public keys must contain the encoding of the public key, including the leading bit. Hence, no signature can satisfy verification for different keys simultaneously, and M-S-UEO must thus hold. The property ES-DEO is equally easy to show since any signature Σ_m must contain the message m (starting at the fixed position) such that each signature can only work for one message.

Next, consider S-CEO. Any signature Σ_m for $\mathcal{I}\backslash\mathcal{C} \neq \emptyset$ created in a query in set $\mathcal{Q}$ is at the front of the form $0||\mathsf{pk}'$, because any partial signature by an honest party starts with a bit 0, resulting in a leading 0-bit via the logical AND. This rules out the possibility for the adversary to output a key pk^* starting with a 1-bit for a successful attack. But any key $\mathsf{pk}^* \neq \mathsf{pk}$ for the genuine key $\mathsf{pk} = 0||\mathsf{pk}'$ then cannot match the entry $0||\mathsf{pk}'$ in Σ_m, showing that the scheme achieves S-CEO.

We finally argue that the scheme is not ES-CEO. For this, let the adversary on input $\mathsf{pk} = 0||\mathsf{pk}'$ corrupt t parties $\mathcal{J}$ and run the partial signature generation algorithm for some message m to create $\sigma_{m,j} = 0||\sigma'_{m_j}$. Hand over m, $\mathcal{I} = \emptyset$, $\mathcal{J}$ and $\{\mathsf{pk}_j, 1||\sigma'_{m,j}\}_{j\in\mathcal{J}}$ to oracle $\mathcal{O}_{\mathsf{augmSign}}$ to get a signature Σ_m. Output $m^* = m$, $\Sigma^* = \Sigma_m$ and $\mathsf{pk}^* = 1||\mathsf{pk}'$. Note that the signature Σ_m created by PartComb starts with the AND of all bits in the set of partial signatures, which is now 1, and thus creates a valid signature starting with a 1-bit. This signature matches the modified public key $\mathsf{pk}^* = 1||\mathsf{pk}'$ starting with 1, thus breaking ES-CEO with probability 1. □

We next separate ES-DEO from ES-DEO. We use the same scheme TS, based on a scheme TS', as in the CEO case, with one change in signature combining and verification, where we flip the message bits m in the signature depending on the AND bit:

1. $\mathsf{PartComb}(\mathsf{pp}, \mathsf{pk}, \mathsf{S}, m, \{\mathsf{pk}_i, \sigma_{m,i}\}_{i\in\mathsf{S}}) \to \Sigma_m$: Runs $\mathsf{PartComb}'(\mathsf{pp}, \mathsf{pk}', \mathsf{S}, m, \{\mathsf{pk}_i, \sigma'_{m,i}\}_{i\in\mathsf{S}}) \to \Sigma'_m$ for $\mathsf{pk}', \sigma'_{m,i}$ with $\mathsf{pk} = b||\mathsf{pk}'$ and $\sigma_{m,i} = b_i||\sigma'_{m,i}$. Let $c = \bigwedge_{i\in\mathsf{S}} b_i$. Returns $\Sigma_m = c||\mathsf{pk}'||\Sigma'_m||(c^{|m|} \oplus m)$.
2. $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma_m) \to 0/1$: Accepts if $\mathsf{Ver}'(\mathsf{pp}, \mathsf{pk}', b^{|m|} \oplus m, \Sigma'_m) \to 0/1$ for $\mathsf{pk} = b||\mathsf{pk}'$ $\Sigma_m = b||\mathsf{pk}'||\Sigma'_m||(b^{|m|} \oplus m)$. (If Σ_m does not parse correctly with leading values $b||\mathsf{pk}'$ and Σ'_m of fixed length, followed by $b^{|m|} \oplus m$, then Ver also returns 0.)

Proposition A.2. *If* TS' *is a correct and* UF-CMA *and* M-S-UEO *threshold signature scheme, then* TS *as defined above is a correct and* UF-CMA *and* M-S-UEO *threshold signature scheme which satisfies* S-CEO *and* ES-DEO *but not* ES-UEO.

Proof. Correctness, unforgeability, and M-S-UEO follow as in the previous case. The extra-strong version ES-CEO of S-CEO holds because each signature Σ_m for some message m in the query set $\mathcal{Q}$ must start with $c||\mathsf{pk}'$ for some bit c and the honest public key $\mathsf{pk} = 0||\mathsf{pk}'$. Hence, at most two keys can match such a signature: $\mathsf{pk} = 0||\mathsf{pk}'$ and $\mathsf{pk}^* = 1||\mathsf{pk}$. However, verification of the signature Σ_m also checks that the message part encodes either m or the flipped version $1^{|m|} \oplus m$, depending on the first bit of the public key. But then the signature Σ_m cannot verify positively for both keys pk and pk^*, yielding that ES-CEO holds.

Next, consider $\mathsf{S\text{-}DEO}$. Any signature Σ_m for $\mathcal{I} \backslash \mathcal{C} \neq \emptyset$ created in a query in set $\mathcal{Q}$ is at the front of the form $0||\mathsf{pk}'$, because any partial signature by an honest party starts with a bit 0, resulting in a leading 0-bit via the logical AND. This rules out the possibility for the adversary to output a key pk^* starting with a 1-bit for a successful attack. But any key $\mathsf{pk}^* \neq \mathsf{pk}$ for the genuine key $\mathsf{pk} = 0||\mathsf{pk}'$ then cannot match the entry $0||\mathsf{pk}'$ in Σ_m, showing that the scheme achieves $\mathsf{S\text{-}DEO}$.

We show that the scheme TS is not $\mathsf{ES\text{-}DEO}$. The adversary receives as input $\mathsf{pk} = 0||\mathsf{pk}'$, corrupts t parties $\mathcal{J}$, and runs the partial signature generation algorithm for some message m to create $\sigma_{m,j} = 0||\sigma'_{m_j}$. Hand over m, $\mathcal{I} = \emptyset$, $\mathcal{J}$ and $\{\mathsf{pk}_j, 1||\sigma'_{m,j}\}_{j \in \mathcal{J}}$ to oracle $\mathcal{O}_{\mathsf{augmSign}}$ to get a signature Σ_m. Note that this signature is of the form $\Sigma_m = 1||\mathsf{pk}'||\Sigma'_m||(1^{|m|} \oplus m)$. The adversary outputs $m^* = 1^{|m|} \oplus m$, $\Sigma^* = \Sigma_m$ and $\mathsf{pk}^* = 1||\mathsf{pk}'$. It follows that Ver' checks the validity of the signature Σ'_m for message $b^{|m^*|} \oplus m^* = m$ and accepts, and that Σ_m correctly encodes the data for the modified public key $\mathsf{pk}^* = 1||\mathsf{pk}'$ starting with 1, The adversary thus breaks $\mathsf{ES\text{-}DEO}$ with probability 1. □

A.2 Separating M-S-UEO and S-UEO in the Non-Robust Setting

In this section, we discuss that $\mathsf{M\text{-}S\text{-}UEO}$ only implies $\mathsf{S\text{-}UEO}$ for robust schemes. That is, we present a scheme TS that is $\mathsf{M\text{-}S\text{-}UEO}$ (and unforgeable and correct if the underlying scheme TS' already is) but not $\mathsf{RB\text{-}CMA}$, and for which we show that $\mathsf{S\text{-}UEO}$ can be broken. This in particular means that the extra-strong notion $\mathsf{ES\text{-}UEO}$ cannot hold either. Consider an arbitrary threshold scheme TS' and modify it to scheme TS as follows:

1. $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: Outputs $\mathsf{Setup}'(1^\lambda) \to \mathsf{pp}'$.

2. $\mathsf{KeyGen}(\mathsf{pp}, n, t) \to (\mathsf{pk}, \{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{sk}_i\}_{i\in[n]})$: Computes the underlying keys $\mathsf{KeyGen}'(\mathsf{pp}', n, t)$ $\to (\mathsf{pk}', \{\mathsf{pk}'_i\}_{i\in[n]}, \{\mathsf{sk}'_i\}_{i\in[n]})$ and returns $\mathsf{pk} = 0||\mathsf{pk}'$, as well as $\{\mathsf{pk}_i\}_{i\in[n]}$, $\{\mathsf{sk}_i\}_{i\in[n]}$ for $\mathsf{pk}_i = \mathsf{pk}'_i$ and $\mathsf{sk}_i = \mathsf{sk}'_i$.

3. $\mathsf{PartSign}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_i, m) \to \sigma_{m,i}$. Runs $\mathsf{PartSign}'(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_i, m) \to \sigma'_{m,i}$ and outputs $\sigma_{m,i} = 0||\sigma'_{m,i}$.

4. $\mathsf{PartVer}(\mathsf{pp}, \mathsf{pk}, m, \mathsf{pk}_i, \sigma_{m,i}) \to 0/1$. Runs $\mathsf{PartVer}'(\mathsf{pp}, \mathsf{pk}', m, \mathsf{pk}_i, \sigma'_{m,i}) \to 0/1$ for $\mathsf{pk}', \sigma'_{m,i}$ with $\mathsf{pk} = b||\mathsf{pk}'$ and $\sigma_{m,i} = b_i||\sigma'_{m,i}$.

5. $\mathsf{PartComb}(\mathsf{pp}, \mathsf{pk}, \mathsf{S}, m, \{\mathsf{pk}_i, \sigma_{m,i}\}_{i\in\mathsf{S}}) \to \Sigma_m$: Runs $\mathsf{PartComb}'(\mathsf{pp}, \mathsf{pk}', \mathsf{S}, m, \{\mathsf{pk}_i, \sigma'_{m,i}\}_{i\in\mathsf{S}}) \to$ Σ'_m for $\mathsf{pk}', \sigma'_{m,i}$ with $\mathsf{pk} = b||\mathsf{pk}'$ and $\sigma_{m,i} = b_i||\sigma'_{m,i}$. Returns $\Sigma_m = 1||\mathsf{pk}'$ if there exists b_i with $b_i = 1$, and $\Sigma_m = 0||\mathsf{pk}||\Sigma'_m$ otherwise.

6. $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}, m, \Sigma_m) \to 0/1$: Accepts if $\mathsf{pk} = 1||\mathsf{pk}'$ and $\Sigma_m = 1||\mathsf{pk}'$, else $\mathsf{Ver}(\mathsf{pp}, \mathsf{pk}', m, \Sigma'_m) \to$ $0/1$ for $\mathsf{pk} = 0||\mathsf{pk}'$ and $\Sigma_m = 0||\mathsf{pk}'||\Sigma'_m$. (In the latter case, if pk does not parse correctly with a leading 0-bit or Σ_m does not parse correctly with leading bits $0||\mathsf{pk}'$, then Ver also returns 0.)

Proposition A.3. *If* TS' *is a correct and* $\mathsf{UF\text{-}CMA}$ *threshold signature scheme for* $n \geq 2$*, then* TS *as defined above is a correct and* $\mathsf{UF\text{-}CMA}$ *threshold signature scheme which satisfies* $\mathsf{M\text{-}S\text{-}UEO}$ *but not* $\mathsf{RB\text{-}CMA}$ *nor* $\mathsf{S\text{-}UEO}$.

Proof. Correctness and unforgeability of TS follow straightforwardly from the correctness properties of TS' because genuinely combined partial signatures (all with leading bit $b_i = 0$) result in a valid signature Σ'_m which verify under an honestly generated key pk with a leading 0-bit. Unforgeability also holds because one can give a straightforward reduction for genuine public keys of the form $\mathsf{pk} = 0||\mathsf{pk}'$.

We first argue that $\mathsf{M\text{-}S\text{-}UEO}$ holds. Note that a valid signature Σ_m for public keys starting with a 0-bit must contain the encoding of $0||\mathsf{pk}'$ at the beginning. Hence, one cannot find different public keys $\mathsf{pk} \neq \mathsf{pk}^*$ of this type that are valid for the same signature Σ_m. Furthermore, the same holds for exceptional keys $\mathsf{pk} = 1||\mathsf{pk}'$, which only accept one signature $\Sigma_m = 1||\mathsf{pk}'$ as valid. Crossover cases with one public key starting with 1 and one with 0 in the $\mathsf{M\text{-}S\text{-}UEO}$ attack cannot work for the same signature either since the leading bit of valid signatures must always match the ones of the public keys.

It remains to argue that the scheme is neither $\mathsf{S\text{-}UEO}$ nor $\mathsf{RB\text{-}CMA}$. For $\mathsf{S\text{-}UEO}$, we let the adversary initially corrupt one of the at least two users ($n \geq 2$), say, $j \in \mathcal{C}$. Then, for the honestly generated public key $\mathsf{pk} = 0||\mathsf{pk}'$ it calls oracle $\mathcal{O}_{\mathsf{augmSign}}$ for an arbitrary message m and $\mathcal{I} = [n]\backslash\{j\}$, $\mathcal{J} = \{j\}$ and a created partial signature $\sigma_{m,j} = 1||\sigma'_{m,j}$. Here, $\sigma'_{m,j}$ is computed with knowledge of the corrupted secret key $\mathsf{sk}_j = \mathsf{sk}'_j$. The result of the call to $\mathcal{O}_{\mathsf{augmSign}}$ is a signature $\Sigma_m = 1||\mathsf{pk}'$ and some entry $(m, \Sigma_m, \mathcal{I})$ in the query set $\mathcal{Q}$. The attacker finally outputs $\mathsf{pk}^* = 1||\mathsf{pk}'$ together with $m^* = m$ and signature $\Sigma^*_m = \Sigma_m = 1||\mathsf{pk}'$. Note that this is a valid signature for $\mathsf{pk}^* \neq \mathsf{pk}$, and we have an entry in $\mathcal{Q}$, such that the adversary wins the $\mathsf{S\text{-}UEO}$ game with probability 1.

To break property $\mathsf{RB\text{-}CMA}$ our adversary corrupts all except for one party i, generates partial signatures $0||\sigma'_{m,j}$ with the help of the secret keys for some message m for $j \neq i$, but changes them to $\sigma_{m,j} = 1||\sigma'_{m,j}$. The attacker outputs (m, i) as the challenge to create partial signature $\sigma_{m,i}$ for the uncorrupt user i. It then submits $([n], m, \{\sigma_{m,i}\}_{i\in[n]})$ in the second round, resulting in a signature of the form $\Sigma_m = 1||\mathsf{pk}'$ due to the leading 1-bits in the (at least one) corrupt signatures (using $n \geq 2$). But then verification Ver rejects this signature Σ_m with leading 1-bit due to the public key pk starting with a bit 0. Hence, our adversary breaks robustness with probability 1. □