Adaptor Signature Schemes with Deniable Presignatures

Adrian Cinal
NASK National Research Institute
Warsaw, Poland
adrian.cinal@nask.pl

Oliwer Sobolewski
NASK National Research Institute
Warsaw, Poland
oliwer.sobolewski@nask.pl

Abstract

An adaptor signature scheme can be seen as committing to a signature using an NP statement, in such a way that (1) the commitment, called a *presignature*, is verifiable, (2) the corresponding witness enables opening the commitment (*adapting* the presignature), and (3) seeing a valid commitment-opening (presignature-signature) pair leaks the witness. In the blockchain space, where signed transactions *must* be broadcast in order to take effect, this last property allows "forcing" a party to leak a witness for possibly multiple presignatures issued for the same NP statement. This then gives rise to many applications such as atomic swaps or payment channels. Importantly, in prevailing adaptor signature schemes, presignatures may already *technically* be signatures, in that only the party with the secret signing key could have produced them. In this case, the presignature alone constitutes evidence of intent to participate in a protocol based on adaptor signatures. Perhaps surprisingly, however, for many practical applications, this strong "binding" of presignatures turns out to be non-essential. We revisit the definitions of adaptor signature schemes and demonstrate that prevailing security requirements are often too strict. To this end, we formally define fair signature exchange (FSE) and abandon the assumption implicit in prior work that adaptor-based FSE must be "symmetric" with both parties using the same adaptor signature scheme. The resulting relaxation of security requirements leads us to the notion of *presignature deniability*, an extension to adaptor signature schemes that we define formally and construct from various assumptions.

Keywords

Adaptor signatures, Deniability, Fairness

1 Introduction

Adaptor signatures, first introduced by Poelstra [35], enable a party to commit to a signature in a manner that is verifiable and allows "reciprocation" by creating another such commitment, a *presignature* from now on, with the same trapdoor. Furthermore, this trapdoor gets leaked when one of the commitments is publicly opened, thus ensuring the other one can be opened as well. This *atomicity*, where either both commitments (presignatures) get opened or neither does, is a key property making adaptor signatures attractive.

Notably, "verifiability" of presignatures is often equated with their being unforgeable [2, 5, 20]. In this case, the presignature $\tilde{\sigma}$ is already a proof of knowledge of the secret key tied to a message, hence *technically* a signature, except with respect to some "non-standard" signature scheme $\tilde{\Sigma}$. An adaptor signature scheme can then be viewed as mangling the "actual" signature σ so that, while still technically convincing (because $\tilde{\Sigma}$ is unforgeable), it does not adhere to the format and specification of some particular signature scheme Σ, and is therefore rejected by the public. This was reflected already in the first attempt at formalizing adaptor signatures by Fournier [18], who observed that adaptor signatures make sense only in settings where "the signature itself is what has value rather than the fact that someone has signed." This reliance on consensus about what constitutes an acceptable signature made cryptocurrencies a natural setting for adaptor signatures [3, 4, 15, 22, 25, 33, 34, 36, 39, 42, 44, 45, 47]. (In cryptocurrencies, signatures also need to eventually be *published* to take effect, which will motivate a no-abort assumption in our theoretical models; see Section 4.) We show in this work, however, that, for practical purposes, presignatures *need not* be unforgeable, and, indeed, from a privacy perspective, *should not* be. To this end, we introduce the notion of *presignature deniability* and demonstrate that many existing constructions already satisfy it.

We embed our investigation into presignature deniability in the context of *fair signature exchange* (FSE) [2, 18, 24], which is where adaptor signatures have found the most adoption. We demonstrate that, when viewed as a building block for FSE, adaptor signature schemes have much *weaker* security requirements than imposed by state-of-the-art definitions. We propose a relaxation that makes security sufficient to instantiate FSE but, at the same time, affords more flexibility and efficiency in constructions (see Section 6). Finally, when complemented with presignature deniability, it lends itself naturally to a black-box security reduction (see Theorem 6.5). Many real-world protocols can be seen as instances of FSE, in particular when they involve paying for some good or service with cryptocurrency. Examples include atomic swaps [25, 43, 45], where one cryptocurrency is exchanged for another, fair issuance of tokens (such as e-cash [9] or Privacy Pass [14]), or, say, paying the notary for a qualified signature on an official document. Despite their prevalence, FSE protocols have so far, to the best of our knowledge, seen little in the way of abstract formalization in the literature,[1] a gap we also fill in this work.

In an adaptor-based FSE protocol (see Figure 1), the *initiator*, call her Alice, issues a presignature $\tilde{\sigma}$ that is "locked" with an instance (Y, y) of a hard relation R in such a way that the *responder*, call him Bob, given $\tilde{\sigma}$, the statement Y, and Alice's public key pk_A, can verify that he will be able to "unlock" $\tilde{\sigma}$ with a corresponding witness y (or, in general, some other y' with $(Y, y') \in \mathsf{R}$) to obtain a regular signature σ under Alice's key pk_A. Bob can furthermore produce a "reciprocal" presignature $\tilde{\sigma}'$ "locked" with the same statement Y. Importantly, he does not need to know the witness y to do this. This last point is crucial for practical application and shaped how adaptor signatures were formalized thus far. Alice can use y to "unlock" or *adapt* $\tilde{\sigma}'$ to get a regular signature σ' under Bob's public key pk_B. If Bob now sees σ' (because it was broadcast to the cryptocurrency network, say), he can use it together with $\tilde{\sigma}'$

[1] Notably, see the work of Asokan *et al.* [1], who introduce a general model for FSE that, however, involves a trusted third party.

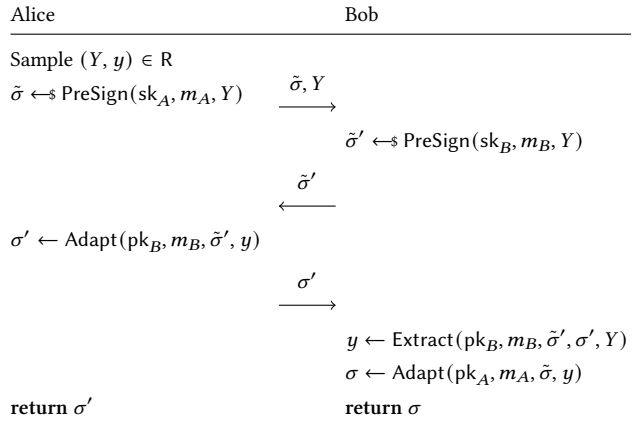


Figure 1: Adaptor-based FSE sketch

to *extract* y and likewise adapt the presignature $\tilde{\sigma}$ he originally obtained from Alice.

Note how *two* presignatures are issued, $\tilde{\sigma}$ and $\tilde{\sigma}'$, and so two adaptor signature schemes are involved. However, it has long been assumed implicitly in the literature that these are identical and have the same security requirements. This has arguably motivated the abstract syntax for adaptor signatures from the start [2]. We show in this work some of the benefits of abandoning this unnecessary assumption. For example, note how Bob is the only one to ever call Extract in this setting. Indeed, Alice already knows the (ephemeral) witness y she sampled herself for the purposes of the protocol execution. This suggests that one of the two adaptor signature schemes used does not need any security guarantees regarding extraction. We formalize this in Theorem 4.6.[2]

Before proceeding, we stress that not *all* applications of adaptor signatures can be viewed through the lens of FSE. For example, Dai *et al.* in [13] consider exclusively the application of exchanging a signature σ for the *witness* y. We focus in this work on an exchange of two signatures, where the witness y is only sampled as a means to this end (ephemeral) and is otherwise of no intrinsic value.

1.1 Contributions

The paper makes the following contributions.

(1) We formalize fair signature exchange (FSE) in a minimalist, generic setting (which, to the best of our knowledge, has not been introduced before without assuming a specific context such as availability of a distributed ledger [24] or a trusted third party [1], or as a part of a more sophisticated construction like in [2]). This leads to formulating the requirements on adaptor signature schemes used to realize FSE. Specifically, the assumption, so far implicit in the literature, that two identical adaptor signature schemes are necessary, is lifted.
(2) We formally show the two adaptor signature schemes have *different* security requirements allowing for a relaxation of definitions. In particular, we show that presignatures need not actually be "unforgeable" as often required in the literature [2, 5, 20, 26].
(3) We define presignature deniability (see Definition 5.1) and show how it removes an accountability asymmetry in FSE: now, either both parties can be held accountable or neither can. Importantly, we show that the original Schnorr-based construction of an adaptor signature scheme [35] already satisfies this property.
(4) We introduce extended adaptor signature schemes, which have the PreSign algorithm take a witness y as input instead of the statement Y. We define an appropriate security notion and show that it is "compatible" in a natural sense with the security of regular adaptor signature schemes (see Theorem 6.3). We then show that the syntactic relaxation does not compromise the security of FSE built on top of adaptor signatures (see Theorem 6.4).
(5) We show that extended adaptor signature schemes with deniable presignatures admit a clean security reduction to the underlying signature scheme's existential unforgeability (see Theorem 6.5).
(6) Finally, in Section 7, we discuss the legal and societal impact of deniability in complex cryptographic protocols.

2 Notation and Conventions

Let $\lambda \in \mathbb{N}$ denote a security parameter. An algorithm is said to be *efficient* if it is probabilistic and runs in time polynomial in the size of its input (which typically means polynomial in λ). As shorthand, we identify the logical value true with integer 1 and the logical value false with integer 0 (see, *e.g.*, Figure 3). A function $f(\lambda)$ is said to be *negligible* if it decays faster than the inverse of any polynomial in λ, *i.e.*, if $f(\lambda) = \lambda^{-\omega(1)}$. For a positive integer n, write $[n] = \{1, \ldots, n\}$. Our definitions of hard relations and signature schemes are standard and recounted for completeness in Appendix C. To each hard relation $\mathsf{R} \subseteq \{0,1\}^* \times \{0,1\}^*$, we associate a language of valid statements $\mathcal{L}_\mathsf{R} = \{x \mid (\exists w)\big((x, w) \in \mathsf{R}\big)\}$.

3 Adaptor Signatures

We recall the syntax of adaptor signature schemes originally established by Aumayr *et al.* [2]. It has been subjected to minor changes over the years, such as adding the message m as an argument to the extracting and adapting algorithms[3] [31] or accounting for *two* hard relations inherent to efficient lattice-based adaptor signature schemes [17, 47].

Definition 3.1 (Adaptor Signature Scheme). An *adaptor signature scheme* $\mathsf{AS}_{\Sigma,\mathsf{R},\mathsf{R}'}$ with respect to a signature scheme

$$\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$$

and hard relations R and $\mathsf{R}' \supseteq \mathsf{R}$ is a tuple of efficient algorithms

$$\mathsf{AS}_{\Sigma,\mathsf{R},\mathsf{R}'} = (\mathsf{PreSign}, \mathsf{PreVerify}, \mathsf{Adapt}, \mathsf{Extract})$$

such that

[2] A similar observation was independently made by Castejon-Molina *et al.* in concurrent work [8]. They develop it, however, in a completely different direction than the present work.

[3] This change makes sense in practice as, in all applications, the party doing the adapting or extracting will want to either preverify the presignature or verify the resulting signature, which they need the message for anyway.

- the *presigning* algorithm PreSign, on input a secret key sk, message m, and statement $Y \in \mathcal{L}_{\mathsf{R}}$, outputs a presignature $\tilde{\sigma}$,
- the *preverification* algorithm PreVerify, on input a public key pk, message m, presignature $\tilde{\sigma}$, and statement $Y \in \mathcal{L}_{\mathsf{R}}$, outputs a bit b,
- the *adapting* algorithm Adapt, on input a public key pk, message m, presignature $\tilde{\sigma}$, and witness y for a statement $Y \in \mathcal{L}_{\mathsf{R}}$, outputs an *adapted signature* σ,
- the *extracting* algorithm Extract, on input a public key pk, message m, presignature $\tilde{\sigma}$, signature σ, and statement $Y \in \mathcal{L}_{\mathsf{R}}$, outputs a witness y such that $(Y, y) \in \mathsf{R}'$ or $\perp$.

When the associated signature scheme Σ and the underlying hard relations R and R′ are clear from context, we simply write AS instead of $\mathsf{AS}_{\Sigma,\mathsf{R},\mathsf{R}'}$.

3.1 Correctness of Adaptor Signatures

Adaptor signature schemes of interest must satisfy basic correctness requirements. *E.g.*, honestly generated presignatures must be verifiable with PreVerify and adaptable with Adapt. This is elaborated in the following definition, originally due to [2].

Definition 3.2 (Presignature Correctness). An adaptor signature scheme AS satisfies *presignature correctness*, if, for all $\lambda \in \mathbb{N}$ and all messages m,

$$\Pr\left[\begin{array}{r|l} \mathsf{PreVerify}(\mathsf{pk}, m, \tilde{\sigma}, Y) = 1 & (\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda) \\ \mathsf{Verify}(\mathsf{pk}, m, \sigma) = 1 & (Y, y) \leftarrow\!\$\ \mathsf{Gen}_{\mathsf{R}}(1^\lambda) \\ (Y, y') \in \mathsf{R}' & \tilde{\sigma} \leftarrow\!\$\ \mathsf{PreSign}(\mathsf{sk}, m, Y) \\ & \sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, m, \tilde{\sigma}, y) \\ & y' \leftarrow \mathsf{Extract}(\mathsf{pk}, m, \tilde{\sigma}, \sigma, Y) \end{array}\right] = 1\,,$$

where the probability is taken over the random coin tosses of the algorithms involved.

Definition 3.2 only requires that Extract yield a valid witness y' for Y, which need *not* be the same as y originally sampled by $\mathsf{Gen}_{\mathsf{R}}$. We must now separately require that this witness enable adapting presignatures associated with Y. To this end, Definition 3.3 (again due to [2]) requires that, if a presignature $\tilde{\sigma}$ preverifies relative to a statement Y, then any valid witness y for Y will allow adapting $\tilde{\sigma}$ into a verifiable signature.

Definition 3.3 (Presignature Adaptability). An adaptor signature scheme AS satisfies *presignature adaptability*, if, for all $\lambda \in \mathbb{N}$, all messages m, all presignatures $\tilde{\sigma}$, all relation instances $(Y, y) \in \mathsf{R}'$, and all public keys pk, we have

$$\big(\mathsf{PreVerify}(\mathsf{pk}, m, \tilde{\sigma}, Y) = 1\big) \implies \big(\mathsf{Verify}(\mathsf{pk}, m, \mathsf{Adapt}(\mathsf{pk}, m, \tilde{\sigma}, y)) = 1\big)\,.$$

Note that presignature adaptability is a stronger notion than that implied by correctness (Definition 3.2) which only requires presignatures honestly generated by PreSign to reveal a witness upon adapting. Definition 3.3, on the other hand, requires that *any* presignature verifiable with PreVerify against $Y \in \mathcal{L}_{\mathsf{R}}$ be adaptable into a valid signature (that will hopefully leak a witness for Y as we shall separately require below).

$\mathsf{aSigForge}_{\mathcal{A},\mathsf{AS}}(\lambda)$

$\mathcal{S} \leftarrow \emptyset$
$\mathcal{T} \leftarrow \emptyset$
$C \leftarrow \emptyset$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
$(m^*, \sigma^*) \leftarrow\!\$\ \mathcal{A}^{O_{\mathsf{Sign}}(\mathsf{sk},\cdot),O_{\mathsf{PreSign}}(\mathsf{sk},\cdot,\cdot),O_{\mathsf{R}}(\lambda)}(\mathsf{pk})$
return $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*)$
$\quad \wedge\ (m^* \notin \mathcal{S})$
$\quad \wedge\ \big(\forall (Y, \tilde{\sigma}) \in \mathcal{T}[m^*]\big)(Y \in C)$

$O_{\mathsf{Sign}}(\mathsf{sk}, m)$

$\sigma \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}, m)$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{m\}$
return σ

$O_{\mathsf{PreSign}}(\mathsf{sk}, m, Y)$

$\tilde{\sigma} \leftarrow\!\$\ \mathsf{PreSign}(\mathsf{sk}, m, Y)$
$\mathcal{T}[m] \leftarrow \mathcal{T}[m] \cup \{(Y, \tilde{\sigma})\}$
return $\tilde{\sigma}$

$O_{\mathsf{R}}(\lambda)$

$(Y, y) \leftarrow\!\$\ \mathsf{Gen}_{\mathsf{R}}(1^\lambda)$
$C \leftarrow C \cup \{Y\}$
return Y

Figure 2: Existential unforgeability (aEUF-CMA) experiment

Definition 3.4 (Adaptor Signature Scheme Correctness). An adaptor signature scheme AS is *correct* if it satisfies presignature correctness and presignature adaptability.

3.2 Security of Adaptor Signatures

Precise definitions of security of adaptor signature schemes seem to be elusive, as an extensive body of work (this one included) aiming to refine them demonstrates [13, 21, 31]. In this paper, we work with the state-of-the-art definitions of *existential unforgeability* and *witness extractability* due to Liu *et al.* [31]. The model of Liu *et al.* addresses the shortcomings of the original model by Aumayr *et al.* [2], namely, lack of support for the multi-query setting, while still maintaining its modularity and simplicity. Notably, there was for a time a trend, started by Dai *et al.* [13] and picked up on by Gerhart *et al.* [21], to unify unforgeability and witness extractability into a single security notion. As Vanjani *et al.*, however, observe in [47], "not all security properties are needed always." This paper establishes this formally for FSE in Theorem 4.6.

Definition 3.5 (Existential Unforgeability). An adaptor signature scheme AS is *existentially unforgeable under adaptive chosen message attack*, if, for all efficient algorithms $\mathcal{A}$, there exists a negligible function μ such that

$$\Pr[\mathsf{aSigForge}_{\mathcal{A},\mathsf{AS}}(\lambda) = 1] \leq \mu(\lambda)\,,$$

where the experiment $\mathsf{aSigForge}_{\mathcal{A},\mathsf{AS}}$ is described in Figure 2, and the probability is taken over the random coin tosses of the algorithms involved.

$\mathsf{WitExt}_{\mathcal{A},\mathsf{AS}}(\lambda)$

$\mathcal{S} \leftarrow \emptyset$
$\mathcal{T} \leftarrow \emptyset$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
$(m^*, \sigma^*) \leftarrow\!\$\ \mathcal{A}^{O_{\mathsf{Sign}}(\mathsf{sk},\cdot), O_{\mathsf{PreSign}}(\mathsf{sk},\cdot,\cdot)}(\mathsf{pk})$
return $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*)$
$\quad \wedge\ (m^* \notin \mathcal{S})$
$\quad \wedge\ \big(\forall (Y, \tilde{\sigma}^*) \in \mathcal{T}[m^*]\big)$
$\quad\quad \big((Y, \mathsf{Extract}(\mathsf{pk}, m^*, \tilde{\sigma}^*, \sigma^*, Y)) \notin \mathsf{R}'\big)$

$O_{\mathsf{Sign}}(\mathsf{sk}, m)$
$\sigma \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}, m)$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{m\}$
return σ

$O_{\mathsf{PreSign}}(\mathsf{sk}, m, Y)$
$\tilde{\sigma} \leftarrow\!\$\ \mathsf{PreSign}(\mathsf{sk}, m, Y)$
$\mathcal{T}[m] \leftarrow \mathcal{T}[m] \cup \{(Y, \tilde{\sigma})\}$
return $\tilde{\sigma}$

Figure 3: Witness extractability experiment

Existential unforgeability of adaptor signature schemes is a simple extension of EUF-CMA security of signature schemes to the setting where a presigning oracle is available to the adversary as well as the regular signing oracle. Notably, the adversary in Figure 2 is asked to produce a *signature*, not merely a presignature. Indeed, Aumayr *et al.* define the analogous notion of *presignature unforgeability* [2, Definition 7], but, as we shall see, being able to "forge" presignatures is actually a feature in many circumstances. This will be revisited and discussed at length in Section 5.

The other security requirement recognized already by Fournier [18] and then revised by Aumayr *et al.* [2] is *witness extractability*.[4] Informally, this is the property protecting Bob the responder in FSE protocols who wants a guarantee, given a statement $Y \in \mathcal{L}_\mathsf{R}$ from Alice, that once he (Bob) issues for Alice a presignature $\tilde{\sigma}$ relative to Y, Alice will not be able to maliciously "adapt" it to a full signature σ under Bob's public key, such that the pair $(\tilde{\sigma}, \sigma)$ will *not* reveal the witness y for Y to Bob. Indeed, this would leave Alice with a signature from Bob but Bob with merely a presignature that he is unable to adapt.

Definition 3.6 (Witness Extractability). An adaptor signature scheme AS is *witness extractable*, if, for all efficient algorithms $\mathcal{A}$, there exists a negligible function μ such that

$$\Pr[\mathsf{WitExt}_{\mathcal{A},\mathsf{AS}}(\lambda) = 1] \leq \mu(\lambda)\,,$$

where the experiment $\mathsf{WitExt}_{\mathcal{A},\mathsf{AS}}$ is described in Figure 3, and the probability is taken over the random coin tosses of the algorithms involved.

Aumayr *et al.* [2] defined also an unforgeability notion for *presignatures*, which Gerhart *et al.* then strengthened in [20] to resolve a tension in their UC model. We quote both their definitions here and revisit them in Section 5, where we compare them against our new notion of presignature deniability.

[4] Fournier [18] uses the term *recoverability*.

$\mathsf{pSigForge}_{\mathcal{A},\mathsf{AS}}(\lambda)$

$Q \leftarrow \emptyset$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
$(Y, y) \leftarrow\!\$\ \mathsf{Gen}_\mathsf{R}(1^\lambda)$
$(m^*, \tilde{\sigma}^*) \leftarrow\!\$\ \mathcal{A}^{O_{\mathsf{Sign}}(\mathsf{sk},\cdot), O_{\mathsf{PreSign}}(\mathsf{sk},\cdot,\cdot)}(\mathsf{pk}, Y)$
return $(m^* \notin Q) \wedge \mathsf{PreVerify}(\mathsf{pk}, m^*, \tilde{\sigma}^*, Y)$

$O_{\mathsf{Sign}}(\mathsf{sk}, m)$
$\sigma \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}, m)$
$Q \leftarrow Q \cup \{m\}$
return σ

$O_{\mathsf{PreSign}}(\mathsf{sk}, m, Y)$
$\tilde{\sigma} \leftarrow\!\$\ \mathsf{PreSign}(\mathsf{sk}, m, Y)$
$Q \leftarrow Q \cup \{m\}$
return $\tilde{\sigma}$

Figure 4: Presignature unforgeability (pEUF-CMA) experiment due to Aumayr *et al.* [2]

Definition 3.7 (Presignature Unforgeability). An adaptor signature scheme AS satisfies *presignature unforgeability under adaptive chosen message attack*, if, for all efficient adversaries $\mathcal{A}$, there exists a negligible function μ such that

$$\Pr[\mathsf{pSigForge}_{\mathcal{A},\mathsf{AS}}(\lambda) = 1] \leq \mu(\lambda)\,,$$

where the experiment $\mathsf{pSigForge}_{\mathcal{A},\mathsf{AS}}$ is described in Figure 4, and the probability is taken over the random coin tosses of the algorithms involved.

Definition 3.8 (Strong Presignature Unforgeability). An adaptor signature scheme AS satisfies *strong presignature unforgeability under adaptive chosen message attack*, if, for all efficient adversaries $\mathcal{A}$, there exists a negligible function μ such that

$$\Pr[\mathsf{spSigForge}_{\mathcal{A},\mathsf{AS}}(\lambda) = 1] \leq \mu(\lambda)\,,$$

where the experiment $\mathsf{spSigForge}_{\mathcal{A},\mathsf{AS}}$ is described in Figure 5, and the probability is taken over the random coin tosses of the algorithms involved.

4 Fair Signature Exchange

A prevalent use case of adaptor signature schemes is *fair signature exchange* (FSE), where two parties, the initiator $\mathfrak{I}$ and the responder $\mathfrak{R}$, want to atomically exchange signatures with either both of them obtaining each other's signatures or neither of them. Indeed, as argued in the introduction already, many real-world applications are implicitly structured as FSE.

FSE can be most naturally realized using adaptor signatures (recall Figure 1). Notably, since Aumayr *et al.* [2] defined PreSign to take only the statement $Y \in \mathcal{L}_\mathsf{R}$ and not its corresponding witness y, the same adaptor signature scheme AS can be used by both parties, ignoring the fact that one of the parties can, without loss of generality, be assumed to already know y. The symmetry, while appealing, is, however, *not* required by FSE, as this section details. Indeed, FSE can be realized most generically using two, possibly distinct, adaptor signature schemes defined, in general, with respect to distinct signature schemes Σ_1 and Σ_2, as long as

$\mathsf{spSigForge}_{\mathcal{A},\mathsf{AS}}(\lambda)$

$\mathcal{S} \leftarrow \emptyset$
$\mathcal{T} \leftarrow \emptyset$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\$ \mathsf{KeyGen}(1^\lambda)$
$(m^*, \tilde{\sigma}^*, Y^*) \leftarrow\$ \mathcal{A}^{O_{\mathsf{Sign}}(\mathsf{sk},\cdot),O_{\mathsf{PreSign}}(\mathsf{sk},\cdot,\cdot)}(\mathsf{pk})$
return $\mathsf{PreVerify}(\mathsf{pk}, m^*, \tilde{\sigma}^*, Y)$
$\quad \wedge\ (m^* \notin \mathcal{S})$
$\quad \wedge\ \big((Y^*, \tilde{\sigma}^*) \notin \mathcal{T}[m^*]\big)$

$O_{\mathsf{Sign}}(\mathsf{sk}, m)$

$\sigma \leftarrow\$ \mathsf{Sign}(\mathsf{sk}, m)$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{m\}$
return σ

$O_{\mathsf{PreSign}}(\mathsf{sk}, m, Y)$

$\tilde{\sigma} \leftarrow\$ \mathsf{PreSign}(\mathsf{sk}, m, Y)$
$\mathcal{T}[m] \leftarrow \mathcal{T}[m] \cup \{(Y, \tilde{\sigma})\}$
return $\tilde{\sigma}$

Figure 5: Strong presignature unforgeability (spEUF-CMA) experiment due to Gerhart *et al.* [20]

the underlying hard relations $\mathsf{R} \subseteq \mathsf{R}'$ are the same for both (*cf.* discussion in Section 5.3). Importantly, the two adaptor signature schemes will have *distinct* objectives, therefore imposing the same security requirements on both would be unnecessarily restrictive.

We begin by formalizing signature exchange abstractly and then instantiate it using adaptor signature schemes.

4.1 Signature Exchange Model

The model we propose assumes a three-move structure with the initiator $\mathfrak{I}$ sending two messages to the responder $\mathfrak{R}$. Importantly, we assume the initiator *does not abort* and sends the second message honestly if the responder message is well-formed. This assumption is crucial for fairness (atomicity) but is also natural in real-world settings, where the initiator is *incentivized* to broadcast the signature they obtain from the responder. In cryptocurrency applications, this incentive is typically financial in nature [3, 4, 15, 22, 25, 33, 34, 36, 39, 42, 44, 45, 47].

Definition 4.1 (Signature Exchange). A *signature exchange* protocol $\mathsf{SE}_{\Sigma_1,\Sigma_2}$ with respect to two signature schemes

$\Sigma_1 = (\mathsf{KeyGen}_1, \mathsf{Sign}_1, \mathsf{Verify}_1)$ and $\Sigma_2 = (\mathsf{KeyGen}_2, \mathsf{Sign}_2, \mathsf{Verify}_2)$

is a tuple of efficient algorithms

$$\mathsf{SE} = (\mathsf{IPrep}, \mathsf{IFinal}, \mathsf{RPrep}, \mathsf{RFinal})$$

such that

- the *initiator preparation* algorithm IPrep, on input an initiator's secret key $\mathsf{sk}_\mathfrak{I}$, initiator's public key $\mathsf{pk}_\mathfrak{I}$, initiator message $m_\mathfrak{I}$, responder's public key $\mathsf{pk}_\mathfrak{R}$, and responder message $m_\mathfrak{R}$, outputs a state $\mathsf{st}_\mathfrak{I}$ and an initiator presignature $\rho_\mathfrak{I}$,
- the *responder preparation* algorithm RPrep, on input a responder's secret key $\mathsf{sk}_\mathfrak{R}$, responder's public key $\mathsf{pk}_\mathfrak{R}$, responder message $m_\mathfrak{R}$, initiator's public key $\mathsf{pk}_\mathfrak{I}$, initiator message $m_\mathfrak{I}$, and initiator presignature $\rho_\mathfrak{I}$, outputs a state $\mathsf{st}_\mathfrak{R}$ and a responder presignature $\rho_\mathfrak{R}$,

$\mathsf{Correctness}_{\mathsf{SE}}(\lambda, m_\mathfrak{I}, m_\mathfrak{R})$

$(\mathsf{pk}_\mathfrak{I}, \mathsf{sk}_\mathfrak{I}) \leftarrow\$ \mathsf{KeyGen}_1(1^\lambda)$
$(\mathsf{pk}_\mathfrak{R}, \mathsf{sk}_\mathfrak{R}) \leftarrow\$ \mathsf{KeyGen}_2(1^\lambda)$
$(\mathsf{st}_\mathfrak{I}, \rho_\mathfrak{I}) \leftarrow\$ \mathsf{IPrep}(\mathsf{sk}_\mathfrak{I}, \mathsf{pk}_\mathfrak{I}, m_\mathfrak{I}, \mathsf{pk}_\mathfrak{R}, m_\mathfrak{R})$
$(\mathsf{st}_\mathfrak{R}, \rho_\mathfrak{R}) \leftarrow\$ \mathsf{RPrep}(\mathsf{sk}_\mathfrak{R}, \mathsf{pk}_\mathfrak{R}, m_\mathfrak{R}, \mathsf{pk}_\mathfrak{I}, m_\mathfrak{I}, \rho_\mathfrak{I})$
$\sigma_\mathfrak{R} \leftarrow\$ \mathsf{IFinal}(\mathsf{st}_\mathfrak{I}, \rho_\mathfrak{R})$
$\sigma_\mathfrak{I} \leftarrow\$ \mathsf{RFinal}(\mathsf{st}_\mathfrak{R}, \sigma_\mathfrak{R})$
return $\mathsf{Verify}_1(\mathsf{pk}_\mathfrak{I}, m_\mathfrak{I}, \sigma_\mathfrak{I}) \wedge \mathsf{Verify}_2(\mathsf{pk}_\mathfrak{R}, m_\mathfrak{R}, \sigma_\mathfrak{R})$

Figure 6: Signature exchange correctness experiment

- the *initiator finalization* algorithm IFinal, on input a state $\mathsf{st}_\mathfrak{I}$ and responder presignature $\rho_\mathfrak{R}$, outputs a responder signature $\sigma_\mathfrak{R}$ or $\perp$,
- the *responder finalization* algorithm RFinal, on input a state $\mathsf{st}_\mathfrak{R}$ and responder signature $\sigma_\mathfrak{R}$, outputs an initiator signature $\sigma_\mathfrak{I}$ or $\perp$.

Definition 4.2 (Signature Exchange Correctness). A signature exchange protocol SE is *correct* if, for all pairs of messages $m_\mathfrak{I}$, $m_\mathfrak{R}$,

$$\Pr[\mathsf{Correctness}_{\mathsf{SE}}(\lambda, m_\mathfrak{I}, m_\mathfrak{R}) = 1] = 1,$$

where the experiment $\mathsf{Correctness}_{\mathsf{SE}}$ is described in Figure 6, and the probability is taken over the random coin tosses of the algorithms involved.

We are only interested in correct signature exchange protocols, which, for honest parties, lead to their exchanging signatures $\sigma_\mathfrak{I}$ and $\sigma_\mathfrak{R}$ with respect to the signature schemes, respectively, Σ_1 and Σ_2. We further require that the signature exchange be fair, *i.e.*, that honest parties cannot be wronged by malicious counterparties. This is the atomicity guarantee: if, say, the responder $\mathfrak{R}$ gets valid $\sigma_\mathfrak{I}$ in the end, then the initiator $\mathfrak{I}$ must also be able to get valid $\sigma_\mathfrak{R}$.

Definition 4.3 (Responder Fairness). A signature exchange protocol $\mathsf{SE}_{\Sigma_1,\Sigma_2}$ satisfies *responder fairness* if, for all efficient algorithms $\mathcal{A}$, there exists a negligible function μ such that

$$\Pr[\mathsf{FairnessResp}_{\mathcal{A},\mathsf{SE}}(\lambda) = 1] \leq \mu(\lambda),$$

where the experiment $\mathsf{FairnessResp}_{\mathcal{A},\mathsf{SE}}$ is defined in Figure 7, and the probability is taken over the random coin tosses of the algorithms involved.

In Figure 7, the adversary takes the role of the initiator and attempts to produce a signature σ^* relative to Σ_2 under $\mathsf{pk}_\mathfrak{R}$ in such a way that the honest responder, interaction with whom is modeled with the responder oracle $O_\mathfrak{R}$, does not obtain the corresponding signature relative to Σ_1 under *any* of the initiator public keys with which signature exchange sessions were started. In other words, the adversary wins by outputting (m^*, σ^*) such that σ^* is a valid signature that was neither obtained from the signing oracle O_{Sign}, nor does it correspond to any signature exchange session (call to $O_\mathfrak{R}$).

Definition 4.4 (Initiator Fairness). A signature exchange protocol $\mathsf{SE}_{\Sigma_1,\Sigma_2}$ satisfies *initiator fairness* if, for all efficient algorithms $\mathcal{A}$,

$\mathsf{FairnessResp}_{\mathcal{A},\mathsf{SE}}(\lambda)$

$\mathcal{S} \leftarrow \emptyset$
$\mathcal{T} \leftarrow \emptyset$
$(\mathsf{pk}_{\mathfrak{R}}, \mathsf{sk}_{\mathfrak{R}}) \leftarrow\!\!\$\ \mathsf{KeyGen}_2(1^\lambda)$
$(m^*, \sigma^*) \leftarrow\!\!\$\ \mathcal{A}^{O_{\mathfrak{R}}(\mathsf{sk}_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{R}}, \cdot, \cdot, \cdot, \cdot), O_{\mathsf{Sign}}(\mathsf{sk}_{\mathfrak{R}}, \cdot)}(\mathsf{pk}_{\mathfrak{R}})$
return $\mathsf{Verify}_2(\mathsf{pk}_{\mathfrak{R}}, m^*, \sigma^*)$
$\quad \wedge\ (m^* \notin \mathcal{S})$
$\quad \wedge\ \big(\forall (\mathsf{st}_{\mathfrak{R}}, m_{\mathfrak{I}}, \mathsf{pk}_{\mathfrak{I}}) \in \mathcal{T}[m^*]\big)$
$\qquad \big(\mathsf{Verify}_1(\mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{I}}, \mathsf{RFinal}(\mathsf{st}_{\mathfrak{R}}, \sigma^*)) \neq 1\big)$

$O_{\mathfrak{R}}(\mathsf{sk}_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{I}}, \rho_{\mathfrak{I}})$

$(\mathsf{st}_{\mathfrak{R}}, \rho_{\mathfrak{R}}) \leftarrow\!\!\$\ \mathsf{RPrep}(\mathsf{sk}_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{I}}, \rho_{\mathfrak{I}})$
if $\rho_{\mathfrak{R}} = \bot$ **then**
$\quad$ **return** $\bot$
$\mathcal{T}[m_{\mathfrak{R}}] \leftarrow \mathcal{T}[m_{\mathfrak{R}}] \cup \{(\mathsf{st}_{\mathfrak{R}}, m_{\mathfrak{I}}, \mathsf{pk}_{\mathfrak{I}})\}$
return $\rho_{\mathfrak{R}}$

$O_{\mathsf{Sign}}(\mathsf{sk}_{\mathfrak{R}}, m)$

$\sigma \leftarrow\!\!\$\ \mathsf{Sign}_2(\mathsf{sk}_{\mathfrak{R}}, m)$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{m\}$
return σ

Figure 7: Responder fairness security experiment

there exists a negligible function μ such that

$$\Pr[\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}(\lambda) = 1] \leq \mu(\lambda),$$

where the experiment $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$ is defined in Figure 8, and the probability is taken over the random coin tosses of the algorithms involved.

The initiator fairness experiment in Figure 8 is analogous to the responder fairness experiment, except, this time, the adversary takes the role of the responder. They can open multiple signature exchange sessions with the initiator via the $O^1_{\mathfrak{I}}$ oracle and finalize them in arbitrary order via $O^2_{\mathfrak{I}}$. The adversary wins if they manage to finalize all signature exchange sessions involving m^* in such a way that they obtain a verifiable signature σ^* on m^*, but the initiator's corresponding signature $\sigma_{\mathfrak{R}}$ for each finalized session involving m^* is malformed, or they produce σ^* without finalizing any sessions involving m^*. Otherwise, the initiator has successfully exchanged a signature on m^* for a signature on *some* responder message $m_{\mathfrak{R}}$.

Definition 4.5 (Fairness). A signature exchange protocol $\mathsf{SE}_{\Sigma_1,\Sigma_2}$ is *fair* if it satisfies both initiator fairness and responder fairness.

4.2 Fair Signature Exchange from Adaptor Signatures

We now present a generic fair signature exchange protocol based on adaptor signatures. Notably, as opposed to most prior work, where FSE is introduced only informally [18, 21] or is a part of a more involved protocol [2], we do not require the two adaptor signature schemes to be the same, or even the two underlying signature schemes Σ_1 and Σ_2 to be the same. All that matters is that the two adaptor signature schemes be defined with respect to the same hard relations $\mathsf{R} \subseteq \mathsf{R}'$. As the following theorem shows, the two schemes need not even enjoy identical security properties.

Theorem 4.6 ($\mathsf{AS}_{\Sigma_1,\mathsf{R},\mathsf{R}'} + \mathsf{AS}_{\Sigma_2,\mathsf{R},\mathsf{R}'} \implies \mathsf{FSE}_{\Sigma_1,\Sigma_2}$). *Let* AS_1 *be a correct and existentially unforgeable adaptor signature scheme*

$$\mathsf{AS}_1 = \mathsf{AS}_{\Sigma_1,\mathsf{R},\mathsf{R}'} = (\mathsf{PreSign}_1, \mathsf{PreVerify}_1, \mathsf{Adapt}_1, \mathsf{Extract}_1)$$

with respect to an existentially unforgeable signature scheme Σ_1 *and hard relations* $\mathsf{R} \subseteq \mathsf{R}'$. *Let* AS_2 *be a correct and witness-extractable adaptor signature scheme*

$$\mathsf{AS}_2 = \mathsf{AS}_{\Sigma_2,\mathsf{R},\mathsf{R}'} = (\mathsf{PreSign}_2, \mathsf{PreVerify}_2, \mathsf{Adapt}_2, \mathsf{Extract}_2)$$

with respect to an existentially unforgeable signature scheme Σ_2 *and the same hard relations* $\mathsf{R} \subseteq \mathsf{R}'$. *Then, the construction in Figure 9 is a correct and fair signature exchange protocol* $\mathsf{SE}_{\Sigma_1,\Sigma_2}$.

Proof. See Appendix E.1. □

Observe that it was to be expected that witness extractability (Definition 3.6) is only required for AS_2 and *not* for AS_1. Indeed, $\mathsf{Extract}_1$ is never called in Figure 9. The initiator, Alice, therefore does not care if the responder can produce a pair $(\tilde{\sigma}', \sigma')$ for which extraction would fail, because she already has the ephemeral witness y she needs. What is surprising, however, is that AS_2 does not even have to be unforgeable (unless the responder's presignature is made available to outside parties). Intuitively, this is because the initiator, Alice, selects the witness y herself anyway, and so she can simply adapt any presignature she gets from the responder, say, Bob. She really need not trouble herself with attempting any kind of forgery. The only security requirement for AS_2 is then that Bob be able to extract the witness from the resulting signature. All this gives further credibility to the split (aEUF-CMA + WitExt) security model due to Liu *et al.* [31], as the combined extractability model due to Dai *et al.* [13] "overconstrains" at least the initiator's adaptor signature scheme unnecessarily.

Finally, we briefly remark on one side issue. Note that the construction in Figure 9 calls for the responder to check that Y received from the initiator is a statement in the language $\mathcal{L}_{\mathsf{R}}$. This can be realized in a number of ways. Gerhart *et al.* [21] assume the language $\mathcal{L}_{\mathsf{R}}$ admits efficient membership testing, for example, while Esgin *et al.* [17], for their lattice-based construction, have the initiator attach to Y a zero-knowledge proof of knowledge of the witness y. Notably, one can remove the check altogether if AS_1 satisfies "preverification soundness" [21, 38], whereby producing preverifiable presignatures $\tilde{\sigma}_{\mathfrak{I}}$ relative to non-statements $Y \notin \mathcal{L}_{\mathsf{R}}$ is infeasible.

5 Presignature Deniability

Under certain circumstances, it may be desirable that presignatures only become binding once the counterparty has reciprocated, *i.e.*, committed themselves to the FSE protocol. Before that, they should be simulatable, leaving the initiator the option to deny having initiated the exchange. For example, given the legal crackdown on cryptocurrencies, it could be desirable for parties engaging in

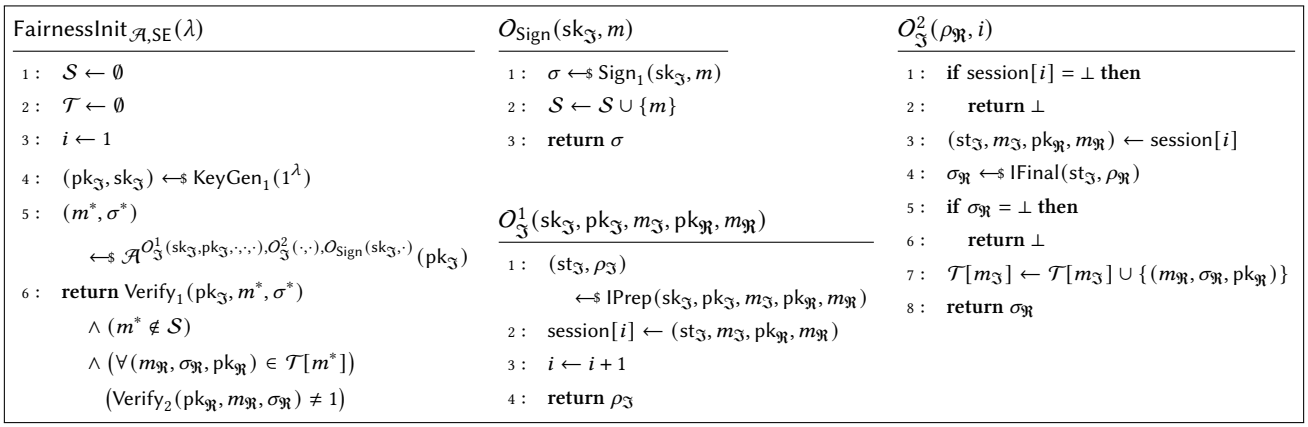


Figure 8: Initiator fairness security experiment

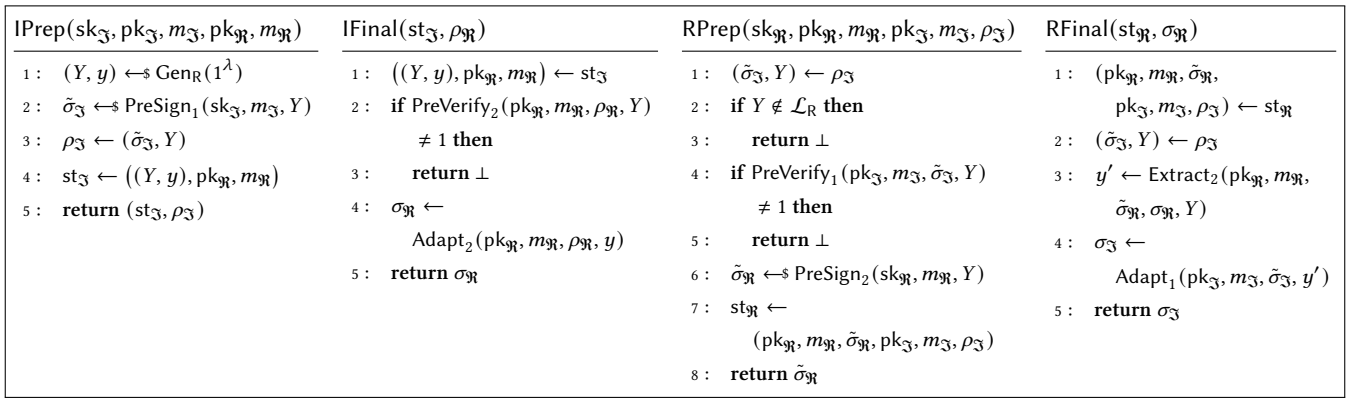


Figure 9: Fair signature exchange based on adaptor signatures

atomic swaps to have protection against undercover law enforcement agents: up until the other party (a law enforcement officer) has reciprocated, the transcript of the swap could in principle have been simulated, reducing its evidentiary power (see Section 7). This must, of course, be achieved without compromising unforgeability of the actual signatures or the atomicity guarantee of FSE (Definition 4.5).

To address this, we introduce the notion of *presignature deniability*. We formalize it using a standard approach, namely, by requiring there exist a *simulator* Sim that, on input the public key and the message (*i.e.*, without the secret key), outputs presignature-statement pairs $(\tilde{\sigma}, Y)$ that are computationally indistinguishable from pairs honestly produced via $\mathsf{Gen}_{\mathsf{R}}$ followed by PreSign. Since the simulator uses only public information, the initiator can deny having issued any presignature $\tilde{\sigma}$. The key difference between presignatures $\tilde{\sigma}$ issued honestly and those simulated, however, is whether or not the issuer knows the witness y corresponding to the statement Y relative to which $\tilde{\sigma}$ is "locked."

Definition 5.1 (Presignature Deniability). An adaptor signature scheme $\mathsf{AS}_{\Sigma,\mathsf{R},\mathsf{R}'}$ *has deniable presignatures* if there exists an efficient algorithm Sim such that, for all efficient distinguishers $\mathcal{D}$, there

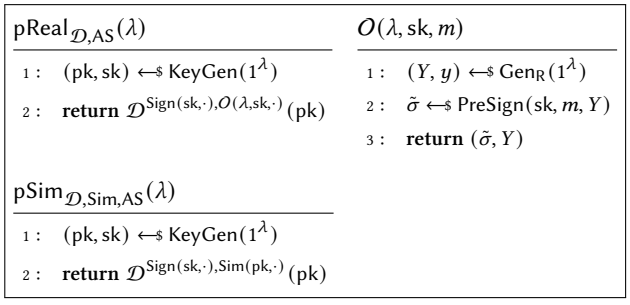


Figure 10: Presignature deniability experiments

exists a negligible function μ such that

$$\left|\Pr[\mathsf{pReal}_{\mathcal{D},\mathsf{AS}}(\lambda) = 1] - \Pr[\mathsf{pSim}_{\mathcal{D},\mathsf{Sim},\mathsf{AS}}(\lambda) = 1]\right| \leq \mu(\lambda)\,.$$

where the experiments pReal and pSim are described in Figure 10, and the probabilities are taken over the random coin tosses of the algorithms involved.

We make a number of remarks about Definition 5.1. First, observe that indistinguishability as above implies in particular that

simulated presignatures are verifiable according to PreVerify, lest PreVerify be an efficient distinguisher. In particular, each $\tilde{\sigma}$ must be "locked" relative to its corresponding Y. Second, note that, if we assume the underlying signature scheme Σ is existentially unforgeable, then the simulator Sim clearly cannot be used to extract the witness y for the instance Y that it produces.

Compare also presignature deniability with *preverification soundness* due to Gerhart *et al.* [21, Definitions 15 and 16] (later revised in [38]). Both capture some notion of "fooling" the preverifier, but are otherwise radically different. Preverification soundness protects the preverifier (say, the responder in FSE) from accepting non-statements $\tilde{Y}$ that are *not* in the language $\mathcal{L}_{\mathsf{R}}$ and therefore have no valid witnesses y that could later afford adaptation. Presignature deniability, on the other hand, is more innocuous: a simulated presignature *is* in every sense a perfectly legitimate presignature, except that nobody knows the witness y needed to adapt it, not even its issuer. And so the preverifier, call him Bob, is only "fooled," in that, if the presignature $\tilde{\sigma}$ preverifies relative to a statement $Y \in \mathcal{L}_{\mathsf{R}}$ and, say, Alice's public key pk, it does not imply that Bob is really interacting with Alice. Bob nevertheless has the guarantee that whoever he is talking to, if it is not Alice, will *not* know the witness y corresponding to Y.

Finally, compare Definition 5.1 with *presignature unforgeability* (Definitions 3.7 and 3.8). Aumayr *et al.* showed that an existentially unforgeable adaptor signature scheme AS that satisfies presignature adaptability has unforgeable presignatures according to Definition 3.7 (see [2, Lemma 10]). This result does *not* contradict our claim that secure adaptor signature schemes with deniable presignatures exist. Indeed, an adaptor signature scheme can have *both* deniable and unforgeable presignatures. This is because Aumayr *et al.*, when defining presignature unforgeability, require the forger to produce a valid presignature *relative to an honestly sampled statement* $Y \in \mathcal{L}_{\mathsf{R}}$ (see Figure 4). On the other hand, our simulator in Definition 5.1 has the freedom to sample Y on its own, a freedom it exercises in practice (see, *e.g.*, Figure 11), thus reconciling unforgeability and deniability. Presignature deniability, however, immediately contradicts the strong presignature unforgeability notion introduced by Gerhart *et al.* [20] (Definition 3.8): the simulator Sim is a successful adversary for the spEUF-CMA experiment.

If AS_1 in Figure 9 has deniable presignatures, the initiator's, Alice's, first message $\rho_{\mathfrak{I}} = (\tilde{\sigma}_{\mathfrak{I}}, Y)$ can be simulated by anyone without the knowledge of the secret key $\mathsf{sk}_{\mathfrak{I}}$. In particular, the entire transcript up to Alice's adapting the responder's reciprocal presignature $\tilde{\sigma}_{\mathfrak{R}}$ can be simulated by the responder alone. This makes convicting Alice difficult (see Section 7) and, in a sense, extends the fairness guarantee to apply to *accountability* as well: either both parties produce evidence of each other's involvement in the protocol or neither does.

5.1 Schnorr Adaptor Signatures with Deniability

Consider the relation $\mathsf{R} = \mathsf{R}' = \{(g^x, x) \mid x \in \mathbb{Z}_q\}$, where g is a fixed generator of a cyclic group $\mathbb{G}$ of prime order q in which the discrete logarithm problem is hard. Let $H : \{0,1\}^* \to \mathbb{Z}_q$ be a hash function. Poelstra's original proposal [35] for Schnorr-based adaptor signatures in this setting already has deniable presignatures, which we evidence by constructing a simulator Sim in Figure 11. It is immediate that, for all messages m and all key pairs $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{KeyGen}(1^\lambda)$,

$$\mathsf{Sim}(\mathsf{pk}, m) \equiv \big(\mathsf{PreSign}(\mathsf{sk}, m, Y), Y\big),$$

over $(Y, y) \leftarrow\!\!\$\ \mathsf{Gen}_{\mathsf{R}}(1^\lambda)$, satisfying Definition 5.1 in particular (with $\mu = 0$).

5.2 Lattice-Based Adaptor Signatures with Deniability

The (Schnorr-like) lattice-based adaptor signature scheme due to Esgin *et al.* [17] admits a simulator defined analogously.

Consider the polynomial quotient ring $\mathcal{R} = \mathbb{Z}_q[X]/(X^d + 1)$ for a prime q and d a power of two. For a positive integer c, write S_c for the set of ring elements $f \in \mathcal{R}$ with $\|f\|_\infty \leq c$, where the infinity norm applies to representatives of the coefficients f_i of f in the range $-q/2 < f_i \leq q/2$. Let $\kappa \in \mathbb{N}$ and define the challenge set:

$$C = \{c \in \mathcal{R} \mid \|c\|_1 = \kappa \wedge \|c\|_\infty = 1\}.$$

Let $H : \{0,1\}^* \to C$ be a cryptographic hash function. The presigner will hash a commitment together with the message to obtain a challenge $c \in C$, *i.e.*, a polynomial of Hamming weight κ with coefficients in $\{-1, 0, 1\}$. Let n and ℓ be integer parameters determining the hardness of Module-SIS and Module-LWE problems, respectively (see [17] for details), and let $\gamma = \kappa d(n + \ell)$. Fix a (uniformly) random $n \times \ell$ matrix $\mathbf{A}'$ over $\mathcal{R}$ and write $\mathbf{A} = [\mathbf{I}_n | \mathbf{A}'] \in \mathcal{R}^{n \times (n+\ell)}$ for $\mathbf{A}'$ left-extended with an $n \times n$ identity matrix. In this setting, Esgin *et al.* [17] introduced their lattice-based adaptor signature scheme LAS with respect to a Dilithium-like signature scheme (see [17]) and the following two relations:

$$\mathsf{R} = \{(\mathbf{t}, \mathbf{r}) \mid \mathbf{t} = \mathbf{A} \cdot \mathbf{r} \text{ and } \|\mathbf{r}\|_\infty \leq 1\}$$

and

$$\mathsf{R}' = \{(\mathbf{t}, \mathbf{r}) \mid \mathbf{t} = \mathbf{A} \cdot \mathbf{r} \text{ and } \|\mathbf{r}\|_\infty \leq 2(\gamma - \kappa)\}.$$

In Figure 12, we recall LAS and evidence that it has deniable presignatures by constructing a simulator Sim.[5] Notably, the output of Sim is only computationally indistinguishable, under the Module-LWE assumption, from $\big(\mathsf{PreSign}(\mathsf{sk}, m, Y), Y\big)$, where $Y = \mathbf{A} \cdot y$ and $\|y\|_\infty \leq 1$. Furthermore, while LAS technically satisfies Definition 5.1, the way it is actually used in practice makes the initiator's opening message non-repudiable. Indeed, as observed by Esgin *et al.* [17], the responder cannot decide themselves if $Y \in \mathcal{L}_{\mathsf{R}}$, as the language $\mathcal{L}_{\mathsf{R}}$ is, under the Module-LWE assumption, not efficiently decidable without a witness (*i.e.*, $\mathcal{L}_{\mathsf{R}} \in \mathsf{NP} \setminus \mathsf{BPP}$). Therefore, in applications (such as FSE), the presignature must be supplemented with a non-interactive zero-knowledge (NIZK) proof of knowledge π demonstrating the knowledge of (and thus also the existence of) the witness y for Y. Knowledge soundness of the NIZK scheme then makes the triple $(\tilde{\sigma}, Y, \pi)$ non-repudiable.

5.3 CL Adaptor Signatures with Deniability

We now turn to the construction due to Gerhart *et al.* [21] of an adaptor signature scheme based on the Camenisch-Lysyanskaya

[5] As a technical note, observe that PreSign in LAS is not *strictly* polynomial-time but rather *expected* polynomial-time.

PreSign(sk, m, Y)

1 : $r \leftarrow\!\!\$\ \mathbb{Z}_q$
2 : $R \leftarrow g^r$
3 : $c \leftarrow H(R \cdot Y, m)$
4 : $s \leftarrow r + c \cdot \mathsf{sk}$
5 : $\tilde{\sigma} \leftarrow (R, s)$
6 : **return** $\tilde{\sigma}$

PreVerify(pk, m, $\tilde{\sigma}$, Y)

1 : $(R, s) \leftarrow \tilde{\sigma}$
2 : $c \leftarrow H(R \cdot Y, m)$
3 : **return** $(g^s = R \cdot \mathsf{pk}^c)$

Unadapt(pk, m, σ, Y, y)

1 : $(R, s) \leftarrow \sigma$
2 : $\tilde{R} \leftarrow R \cdot Y^{-1}$
3 : $\tilde{s} \leftarrow s - y$
4 : $\tilde{\sigma} \leftarrow (\tilde{R}, \tilde{s})$
5 : **return** $\tilde{\sigma}$

Adapt(pk, m, $\tilde{\sigma}$, y)

1 : $(\tilde{R}, \tilde{s}) \leftarrow \tilde{\sigma}$
2 : $R \leftarrow \tilde{R} \cdot g^y$
3 : $s \leftarrow \tilde{s} + y$
4 : $\sigma \leftarrow (R, s)$
5 : **return** σ

Extract(pk, m, $\tilde{\sigma}$, σ, Y)

1 : $(\tilde{R}, \tilde{s}) \leftarrow \tilde{\sigma}$
2 : $(R, s) \leftarrow \sigma$
3 : $y \leftarrow s - \tilde{s}$
4 : **return** y

Sim(pk, m)

1 : $u, s \leftarrow\!\!\$\ \mathbb{Z}_q$
2 : $U \leftarrow g^u$
3 : $c \leftarrow H(U, m)$
4 : $R \leftarrow g^s \cdot \mathsf{pk}^{-c}$
5 : $Y \leftarrow U \cdot R^{-1}$
6 : $\tilde{\sigma} \leftarrow (R, s)$
7 : **return** $(\tilde{\sigma}, Y)$

Figure 11: Schnorr adaptor signature scheme due to Poelstra [35] with Unadapt **and** Sim **added**

PreSign(sk, m, Y)

1 : $\mathbf{r} \leftarrow\!\!\$\ S_\gamma^{n+\ell}$
2 : $\mathbf{w} \leftarrow \mathbf{A} \cdot \mathbf{r}$
3 : $c \leftarrow H(\mathbf{A} \cdot \mathsf{sk}, \mathbf{w} + Y, m)$
4 : $\mathbf{z} \leftarrow \mathbf{r} + c \cdot \mathsf{sk}$
5 : **if** $\|\mathbf{z}\|_\infty > \gamma - \kappa - 1$ **then**
6 : Restart
7 : $\tilde{\sigma} \leftarrow (c, \mathbf{z})$
8 : **return** $\tilde{\sigma}$

PreVerify(pk, m, $\tilde{\sigma}$, Y)

1 : $(c, \mathbf{z}) \leftarrow \tilde{\sigma}$
2 : $\mathbf{w} \leftarrow \mathbf{A} \cdot \mathbf{z} - c \cdot \mathsf{pk}$
3 : **return** $(\|\mathbf{z}\|_\infty \le \gamma - \kappa - 1)$
$\wedge\ \big(c = H(\mathsf{pk}, \mathbf{w} + Y, m)\big)$

Adapt(pk, m, $\tilde{\sigma}$, y)

1 : $(\tilde{c}, \tilde{\mathbf{z}}) \leftarrow \tilde{\sigma}$
2 : $\mathbf{z} \leftarrow \tilde{\mathbf{z}} + y$
3 : $\sigma \leftarrow (c, \mathbf{z})$
4 : **return** σ

Extract(pk, m, $\tilde{\sigma}$, σ, Y)

1 : $(\tilde{c}, \tilde{\mathbf{z}}) \leftarrow \tilde{\sigma}$
2 : $(c, \mathbf{z}) \leftarrow \sigma$
3 : $y \leftarrow \mathbf{z} - \tilde{\mathbf{z}}$
4 : **if** $Y \neq \mathbf{A} \cdot y$ **then**
5 : **return** $\perp$
6 : **return** y

Sim(pk, m)

1 : $\mathbf{z} \leftarrow\!\!\$\ S_{\gamma-\kappa-1}^{n+\ell}$
2 : $\mathbf{v} \leftarrow\!\!\$\ S_1^{n+\ell}$
3 : $\mathbf{u} \leftarrow \mathbf{A} \cdot \mathbf{v}$
4 : $c \leftarrow H(\mathsf{pk}, \mathbf{u}, m)$
5 : $\mathbf{w} \leftarrow \mathbf{A} \cdot \mathbf{z} - c \cdot \mathsf{pk}$
6 : $Y \leftarrow \mathbf{u} - \mathbf{w}$
7 : $\tilde{\sigma} \leftarrow (c, \mathbf{z})$
8 : **return** $(\tilde{\sigma}, Y)$

Figure 12: Lattice-based adaptor signature scheme, termed LAS**, due to Esgin *et al.* [17] with a simulator** Sim **added**

(CL) signature scheme [6] and show that it also has deniable presignatures.[6] Let n be a special RSA modulus $n = (2p' + 1)(2q' + 1)$, where p', q', $p = 2p' + 1$, and $q = 2q' + 1$ are all prime, $|p| = |q| = \lambda$, and $p \equiv q \equiv 3 \pmod 4$. Let $\ell_n, \ell_m, \ell_r, \ell_e \in \mathbb{N}$ be the parameters of the CL scheme with $\ell_n = 2\lambda$, $\ell_r = \ell_n + \ell_m + \ell(\lambda)$, for some security parameter $\ell(\lambda)$, and $\ell_e > \ell_r + 3$ (see [6, 21]). Figure 13 recalls (a simplified version of) the CL-based adaptor signature scheme due to Gerhart *et al.* [21], defined with respect to the following hard relation:

$$\mathsf{R} = \{(Y, y) \mid Y = b^y \bmod n\}\,.$$

The construction already has deniable presignatures, which we show by constructing a simulator Sim in Figure 13.

Importantly, however, note that the scheme is of no use in the fair signature exchange setting, as the underlying hard relation R *depends on the public key* (a, b, c, n) (specifically on b and n). On the other hand, when instantiating FSE from two adaptor signature schemes (recall Figure 9), the two schemes must be defined with respect to the *same* relation. Trying to use the CL-based scheme in this setting would imply that the two parties must share at least the public modulus n, which then implies they must share the secret key (p, q) as well, making the construction useless.

[6] Gerhart *et al.* [21] give a more complicated variant of CL adaptors, because they need SUF-CMA security of the underlying scheme for their *transparent reduction*. We sketch the simpler version here for brevity and clarity of exposition.

6 Extended Adaptor Signature Schemes

We now introduce an extension to the standard adaptor signature syntax aimed at capturing the asymmetry in who chooses the "lock" $Y \in \mathcal{L}_\mathsf{R}$ inherent to adaptor-based FSE (recall Figure 9). This asymmetry will further motivate a revision of the security model. As we will then see, the new model will be particularly compatible with presignature deniability (which is also inherently asymmetric and applies only to the initiator).

Observe that, in the setting of adaptor-based FSE (Figure 9), when the initiator calls $\mathsf{PreSign}_1$, we may, without loss of generality, assume they know not only the statement Y relative to which the two presignatures exchanged are created but also the ephemeral witness y. Indeed, the relation instance $(Y, y) \in \mathsf{R}$ is explicitly sampled by the initiator for the purpose of facilitating signature exchange. This then motivates a syntactic change to *one* (but crucially not both) of the adaptor signature schemes used, namely, AS_1, where the presigning algorithm $\mathsf{PreSign}_1$, instead of taking a statement of R

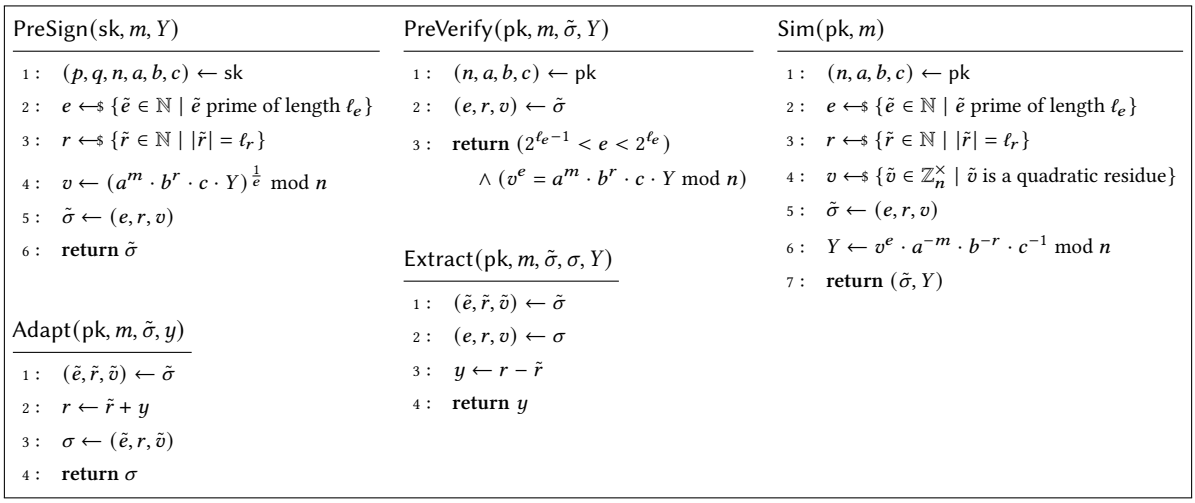


Figure 13: CL-based adaptor signature scheme due to Gerhart *et al.* [21] with a simulator Sim added

as argument, takes a witness for it.[7] We get the following definition that differs from Definition 3.1 only in the last argument to PreSign.

Definition 6.1 (Extended Adaptor Signature Scheme). An *extended adaptor signature scheme* $\mathsf{EAS}_{\Sigma,\mathsf{R},\mathsf{R}'}$ with respect to a signature scheme

$$\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$$

and hard relations R and $\mathsf{R}' \supseteq \mathsf{R}$ is a tuple of efficient algorithms

$$\mathsf{EAS}_{\Sigma,\mathsf{R},\mathsf{R}'} = (\mathsf{PreSign}, \mathsf{PreVerify}, \mathsf{Adapt}, \mathsf{Extract})$$

such that

- the *presigning* algorithm PreSign, on input a secret key sk, message m, and witness y, outputs a presignature $\tilde{\sigma}$,
- the *preverification* algorithm PreVerify, on input a public key pk, message m, presignature $\tilde{\sigma}$, and statement $Y \in \mathcal{L}_{\mathsf{R}}$, outputs a bit b,
- the *adapting* algorithm Adapt, on input a public key pk, message m, presignature $\tilde{\sigma}$, and witness y for a statement $Y \in \mathcal{L}_{\mathsf{R}}$, outputs an *adapted signature* σ,
- the *extracting* algorithm Extract, on input a public key pk, message m, presignature $\tilde{\sigma}$, signature σ, and statement $Y \in \mathcal{L}_{\mathsf{R}}$, outputs a witness y such that $(Y, y) \in \mathsf{R}'$ or $\perp$.

When the associated signature scheme Σ and the underlying hard relations R and R' are clear from context, we simply write EAS instead of $\mathsf{EAS}_{\Sigma,\mathsf{R},\mathsf{R}'}$.

Clearly, whenever one can compute statements Y of R from their corresponding witnesses y (which we can always assume without loss of generality), then any (regular) adaptor signature scheme AS with respect to R can be trivially "compiled" into an extended adaptor signature scheme EAS with respect to R by setting

$$\mathsf{PreSign}_{\mathsf{EAS}}(\mathsf{sk}, m, y) := \mathsf{PreSign}_{\mathsf{AS}}(\mathsf{sk}, m, F(y)), \tag{1}$$

where F is the one-way function associated to the hard relation R (*cf. canonical hard relations* in [21]). Direct access to the witness in PreSign, however, affords an arguably simpler variant of the Schnorr-based scheme, which we present in Figure 14.[8]

We stress that extended adaptor signature schemes cannot be expected to replace regular adaptor signature schemes in *all* applications. The property that a party doing presigning already knows y is specific to FSE (which, notably, does cover a large portion of uses of adaptor signatures). It is *not* the case, for example, when one considers the "adaptor signing" protocol from [13], which corresponds to, *e.g.*, buying the witness y for cryptocurrency. Extended adaptor signature schemes are therefore complementary to regular adaptor signature schemes rather than aiming to replace them (*cf.* Theorem 6.4).

Presignature correctness, presignature adaptability, and presignature deniability for extended adaptor signature schemes are defined analogously to the case of regular adaptor signatures with only syntactic changes. For lack of space, we omit the full definitions in the main body of this work and refer readers to Appendix D. As for security, however, since extended adaptor signature schemes can only be used by presigners that already know the witness for the statement relative to which they presign, it does not make sense to require witness extractability, and so we only consider existential unforgeability.[9] In the unforgeability experiment in Figure 15, we have the presigning oracle O_{PreSign} sample relation instances $(Y, y) \in \mathsf{R}$ honestly (since presigning now requires the witness y instead of the statement Y). We then give the adversary the capability to "close sessions," namely, choose which presignatures obtained from O_{PreSign} to adapt. This is captured by the O_{Adapt} oracle. Notably, we have O_{Adapt} also leak the witness y that O_{PreSign} originally sampled when producing $(\tilde{\sigma}, Y)$. This only gives more power to the adversary.[10] Once a presignature on a message m

[7] Compare with *weak adaptor signatures* of Bursuc and Mauw [5] and with *lockable signatures* of Thyagarajan and Malavolta [42].

[8] Compare it also to the ECDSA-based construction in [5].

[9] Compare with *non-extractable adaptor signature schemes* introduced in concurrent work [8] for a different purpose.

[10] Notably, not much more, since presignature correctness already guarantees $\mathcal{A}$ will be able to obtain *some* witness y' for Y. We require, however, that seeing the original witness y not help the adversary in any way. Compare this also with *faithful extraction* defined recently by Liu *et al.* [32].

PreSign(sk, m, y)

1 : $r \leftarrow\$ \mathbb{Z}_q$
2 : $R \leftarrow g^r$
3 : $c \leftarrow H(R, m)$
4 : $s \leftarrow r + c \cdot \mathsf{sk} + y$
5 : $\tilde{\sigma} \leftarrow (R, s)$
6 : **return** $\tilde{\sigma}$

PreVerify(pk, $m, \tilde{\sigma}, Y$)

1 : $(R, s) \leftarrow \tilde{\sigma}$
2 : $c \leftarrow H(R, m)$
3 : **return** $(g^s = R \cdot \mathsf{pk}^c \cdot Y)$

Unadapt(pk, m, σ, Y, y)

1 : $(R, s) \leftarrow \sigma$
2 : $\tilde{s} \leftarrow s + y$
3 : $\tilde{\sigma} \leftarrow (R, \tilde{s})$
4 : **return** $\tilde{\sigma}$

Adapt(pk, $m, \tilde{\sigma}, y$)

1 : $(\tilde{R}, \tilde{s}) \leftarrow \tilde{\sigma}$
2 : $s \leftarrow \tilde{s} - y$
3 : $\sigma \leftarrow (\tilde{R}, s)$
4 : **return** σ

Extract(pk, $m, \tilde{\sigma}, \sigma, Y$)

1 : $(\tilde{R}, \tilde{s}) \leftarrow \tilde{\sigma}$
2 : $(R, s) \leftarrow \sigma$
3 : $y \leftarrow \tilde{s} - s$
4 : **return** y

Sim(pk, m)

1 : $s, r \leftarrow\$ \mathbb{Z}_q$
2 : $R \leftarrow g^r$
3 : $c \leftarrow H(R, m)$
4 : $Y \leftarrow g^s \cdot \mathsf{pk}^{-c} \cdot R^{-1}$
5 : $\tilde{\sigma} \leftarrow (R, s)$
6 : **return** $(\tilde{\sigma}, Y)$

Figure 14: Schnorr extended adaptor signature scheme with deniable presignatures

has been adapted, the message is added to the query set Q shared with the signing oracle O_{Sign}. This corresponds to the view of the responder in FSE protocols: the adversary is tasked with producing a forgery on a message m^* for which no session was closed and about which the signing oracle O_{Sign} was never queried.

Definition 6.2 (Extended Existential Unforgeability). An extended adaptor signature scheme EAS is *existentially unforgeable under adaptive chosen message attack*, if, for all efficient algorithms $\mathcal{A}$, there exists a negligible function μ such that

$$\Pr[\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}(\lambda) = 1] \leq \mu(\lambda)\,,$$

where the experiment $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$ is described in Figure 15, and the probability is taken over the random coin tosses of the algorithms involved.

The following theorem shows that Definition 6.2 is "compatible" with the natural construction from Equation (1), *i.e.*, the compilation preserves existential unforgeability (necessarily defined in different ways for regular adaptor signature schemes and extended adaptor signature schemes).

Theorem 6.3. *Extended adaptor signature scheme* EAS *compiled via Equation* (1) *from a correct and existentially unforgeable adaptor signature scheme* AS *is correct and existentially unforgeable under adaptive chosen message attack.*

Proof. See Appendix E.2. □

The following theorem formalizes the observation we started this section with, namely, that the initiator in FSE may, without loss of generality, be assumed to be using an *extended* adaptor signature scheme instead of a regular one.

Theorem 6.4 ($\mathsf{EAS}_{\Sigma_1,\mathsf{R},\mathsf{R}'} + \mathsf{AS}_{\Sigma_2,\mathsf{R},\mathsf{R}'} \implies \mathsf{FSE}_{\Sigma_1,\Sigma_2}$). *Let* EAS_1 *be a correct and existentially unforgeable extended adaptor signature scheme*

$$\mathsf{EAS}_1 = \mathsf{EAS}_{\Sigma_1,\mathsf{R},\mathsf{R}'} = (\mathsf{PreSign}_1, \mathsf{PreVerify}_1, \mathsf{Adapt}_1, \mathsf{Extract}_1)$$

with respect to an existentially unforgeable signature scheme Σ_1 *and hard relations* $\mathsf{R} \subseteq \mathsf{R}'$. *Let* AS_2 *be a correct and witness-extractable adaptor signature scheme*

$$\mathsf{AS}_2 = \mathsf{AS}_{\Sigma_2,\mathsf{R},\mathsf{R}'} = (\mathsf{PreSign}_2, \mathsf{PreVerify}_2, \mathsf{Adapt}_2, \mathsf{Extract}_2)$$

with respect to an existentially unforgeable signature scheme Σ_2 *and the same hard relations* $\mathsf{R} \subseteq \mathsf{R}'$. *Then, the construction in Figure 16 is a correct and fair signature exchange protocol* $\mathsf{SE}_{\Sigma_1,\Sigma_2}$.

Proof. See Appendix E.3. □

Note the natural correspondence of the extended adaptor signature syntax and presignature deniability: both only make sense for the FSE initiator (see Figure 16). Indeed, the responder's presignature $\tilde{\sigma}_{\mathfrak{R}}$ is *not* deniable, as the responder produces it relative to a statement Y chosen at random by the initiator (see Definition 3.7). We now further show how the two notions come together with one additional assumption to afford a simple black-box reduction of aEUF-CMA security of an extended adaptor signature scheme $\mathsf{EAS} = \mathsf{EAS}_{\Sigma,\mathsf{R},\mathsf{R}'}$ (Definition 6.2) to the EUF-CMA security of the underlying signature scheme Σ (Definition C.4).

Suppose there exists an *unadapting algorithm* Unadapt for EAS such that, on input a public key pk, message m, signature σ, and relation instance $(Y, y) \in \mathsf{R}$, Unadapt outputs a presignature $\tilde{\sigma}$, and that, for all messages m and all key pairs $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{KeyGen}(1^\lambda)$, we have

$$\mathsf{Unadapt}\big(\mathsf{pk}, m, \mathsf{Sign}(\mathsf{sk}, m), Y, y\big) \equiv \mathsf{PreSign}(\mathsf{sk}, m, y)\,,$$

over $(Y, y) \leftarrow\$ \mathsf{Gen}_{\mathsf{R}}(1^\lambda)$. Note how Unadapt can be viewed as an "inverse" of Adapt. It transforms signatures into presignatures that are identically distributed to presignatures obtained by calling PreSign but does so *without* the knowledge of the secret signing key sk. We have the following result.

Theorem 6.5 (Unadapt + Sim + EUF-CMA $\implies$ aEUF-CMA). *Let* Σ *be an existentially unforgeable signature scheme. Let* $\mathsf{EAS} = \mathsf{EAS}_{\Sigma,\mathsf{R},\mathsf{R}'}$ *be a correct extended adaptor signature scheme with respect to* Σ *that has deniable presignatures. Suppose also that there exists an unadapting algorithm* Unadapt *for* EAS *as above. Then* EAS *is existentially unforgeable under adaptive chosen message attack.*

Proof. See Appendix E.4. □

Intuitively, Theorem 6.5 says that, since presignatures relative to statements with unknown witnesses can be obtained from Sim, having access to the presigning oracle O_{PreSign} does not give the

$\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}(\lambda)$

$Q \leftarrow \emptyset$
$\mathsf{session} \leftarrow \emptyset$
$i \leftarrow 1$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
$(m^*, \sigma^*) \leftarrow\!\$\ \mathcal{A}^{O_{\mathsf{Sign}}(\mathsf{sk},\cdot),O_{\mathsf{PreSign}}(\lambda,\mathsf{sk},\cdot),O_{\mathsf{Adapt}}(\mathsf{pk},\cdot)}(\mathsf{pk})$
return $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*) \wedge (m^* \notin Q)$

$O_{\mathsf{Sign}}(\mathsf{sk}, m)$

$\sigma \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}, m)$
$Q \leftarrow Q \cup \{m\}$
return σ

$O_{\mathsf{PreSign}}(\lambda, \mathsf{sk}, m)$

$(Y, y) \leftarrow\!\$\ \mathsf{Gen}_\mathsf{R}(1^\lambda)$
$\tilde{\sigma} \leftarrow\!\$\ \mathsf{PreSign}(\mathsf{sk}, m, y)$
$\mathsf{session}[i] \leftarrow (y, \tilde{\sigma}, m)$
$i \leftarrow i + 1$
return $(\tilde{\sigma}, Y)$

$O_{\mathsf{Adapt}}(\mathsf{pk}, i)$

if $\mathsf{session}[i] = \perp$ **then**
  return $\perp$
$(y, \tilde{\sigma}, m) \leftarrow \mathsf{session}[i]$
$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, m, \tilde{\sigma}, y)$
$Q \leftarrow Q \cup \{m\}$
return (σ, y)

Figure 15: Existential unforgeability experiment for an extended adaptor signature scheme

$\mathsf{IPrep}(\mathsf{sk}_\mathfrak{I}, \mathsf{pk}_\mathfrak{I}, m_\mathfrak{I}, \mathsf{pk}_\mathfrak{R}, m_\mathfrak{R})$

$(Y, y) \leftarrow\!\$\ \mathsf{Gen}_\mathsf{R}(1^\lambda)$
$\tilde{\sigma}_\mathfrak{I} \leftarrow\!\$\ \mathsf{PreSign}_1(\mathsf{sk}_\mathfrak{I}, m_\mathfrak{I}, y)$
$\rho_\mathfrak{I} \leftarrow (\tilde{\sigma}_\mathfrak{I}, Y)$
$\mathsf{st}_\mathfrak{I} \leftarrow \big((Y, y), \mathsf{pk}_\mathfrak{R}, m_\mathfrak{R}\big)$
return $(\mathsf{st}_\mathfrak{I}, \rho_\mathfrak{I})$

$\mathsf{IFinal}(\mathsf{st}_\mathfrak{I}, \rho_\mathfrak{R})$

$\big((Y, y), \mathsf{pk}_\mathfrak{R}, m_\mathfrak{R}\big) \leftarrow \mathsf{st}_\mathfrak{I}$
if $\mathsf{PreVerify}_2(\mathsf{pk}_\mathfrak{R}, m_\mathfrak{R}, \rho_\mathfrak{R}, Y) \neq 1$ **then**
  return $\perp$
$\sigma_\mathfrak{R} \leftarrow \mathsf{Adapt}_2(\mathsf{pk}_\mathfrak{R}, m_\mathfrak{R}, \rho_\mathfrak{R}, y)$
return $\sigma_\mathfrak{R}$

$\mathsf{RPrep}(\mathsf{sk}_\mathfrak{R}, \mathsf{pk}_\mathfrak{R}, m_\mathfrak{R}, \mathsf{pk}_\mathfrak{I}, m_\mathfrak{I}, \rho_\mathfrak{I})$

$(\tilde{\sigma}_\mathfrak{I}, Y) \leftarrow \rho_\mathfrak{I}$
if $Y \notin \mathcal{L}_\mathsf{R}$ **then**
  return $\perp$
if $\mathsf{PreVerify}_1(\mathsf{pk}_\mathfrak{I}, m_\mathfrak{I}, \tilde{\sigma}_\mathfrak{I}, Y) \neq 1$ **then**
  return $\perp$
$\tilde{\sigma}_\mathfrak{R} \leftarrow\!\$\ \mathsf{PreSign}_2(\mathsf{sk}_\mathfrak{R}, m_\mathfrak{R}, Y)$
$\mathsf{st}_\mathfrak{R} \leftarrow (\mathsf{pk}_\mathfrak{R}, m_\mathfrak{R}, \tilde{\sigma}_\mathfrak{R}, \mathsf{pk}_\mathfrak{I}, m_\mathfrak{I}, \rho_\mathfrak{I})$
return $\tilde{\sigma}_\mathfrak{R}$

$\mathsf{RFinal}(\mathsf{st}_\mathfrak{R}, \sigma_\mathfrak{R})$

$(\mathsf{pk}_\mathfrak{R}, m_\mathfrak{R}, \tilde{\sigma}_\mathfrak{R}, \mathsf{pk}_\mathfrak{I}, m_\mathfrak{I}, \rho_\mathfrak{I}) \leftarrow \mathsf{st}_\mathfrak{R}$
$(\tilde{\sigma}_\mathfrak{I}, Y) \leftarrow \rho_\mathfrak{I}$
$y' \leftarrow \mathsf{Extract}_2(\mathsf{pk}_\mathfrak{R}, m_\mathfrak{R}, \tilde{\sigma}_\mathfrak{R}, \sigma_\mathfrak{R}, Y)$
$\sigma_\mathfrak{I} \leftarrow \mathsf{Adapt}_1(\mathsf{pk}_\mathfrak{I}, m_\mathfrak{I}, \tilde{\sigma}_\mathfrak{I}, y')$
return $\sigma_\mathfrak{I}$

Figure 16: Fair signature exchange based on an extended adaptor signature scheme EAS_1 and an adaptor signature scheme AS_2

adversary any advantage in forging signatures: anything the oracle gives them, they could have produced themselves.

Unadapting algorithms exist for both the standard Schnorr-based construction (see Figure 11) as well as its optimized extended variant (see Figure 14).[11] Notably, however, applying the trick from Figure 11 to Figure 12 would not give us a good unadapting algorithm. Indeed, if, on input a signature $(c, \mathbf{z})$ and an instance $(Y, y) \in \mathsf{R}$, Unadapt were to output a presignature $(c, \mathbf{z} - y)$, then we would only have

$$\|\mathbf{z} - y\|_\infty \leq \|\mathbf{z}\|_\infty + \|y\|_\infty \leq \gamma - \kappa - 1 + 1 = \gamma - \kappa$$

in general, and so $(c, \mathbf{z} - y)$ would fail preverification. Furthermore, the distribution of $\mathbf{z} - y$ would be statistically non-negligibly far from a uniform distribution on $S_{\gamma-\kappa-1}$, so modifying PreVerify would not help.[12]

7 Cryptographic Deniability Meets the Real World

We now turn to assessing the impact of presignature deniability on court proceedings and law enforcement actions in the real world.

In general, cryptographic deniability is a property of a protocol that allows its participants to repudiate post factum any accusation related to their having participated in the protocol. This follows from their ability to demonstrate that the evidence, in the form of a supposedly recorded transcript of the protocol's execution, could have, in principle, been forged or *simulated* (recall Definition 5.1). While cryptographers are quick to assume participants in a deniable protocol cannot be held accountable, in many real-world settings, this is not the case. Legal practice shows that *context* matters, and the friction involved in producing a simulated transcript plays an important role [12, 30, 48]. A non-expert, for example, is less likely to have undertaken (or found and paid an expert to undertake on their behalf) the effort of forging the evidence, despite this "forgeability" being perfectly feasible in cryptographic or complexity-theoretic terms. Indeed, in this sense, simulatability of transcripts is no different from the (theoretical) capability to forge *physical* evidence, say, a handwritten signature on a paper document, a possibility (however remote it may be) the judicial system has evolved to account for. This is reflected in real-world court rulings, where (technically deniable) transcripts of WhatsApp conversations are hardly ever contested, let alone rejected, as evidence [12, 48]. Similarly, the broader society is not receptive to complexity-theoretic claims about simulatability or unforgeability as reported by Yadav *et al.* [48] and by Reitinger *et al.* [37]. Laymen

[11] Ability to unadapt Schnorr signatures into presignatures is briefly mentioned by Gerhart *et al.* in [20].

[12] We could alternatively try redefining Unadapt to let it sample its own relation instance $(Y, y) \in \mathsf{R}$ and then return it. It could then rejection-sample (Y, y) until $\mathbf{z}$ is in the correct range, but this would in turn introduce a noticeable bias in y, something the proof of Theorem 6.5 cannot afford either.

are likely to take the transcript at face value and indeed often struggle to understand the (admittedly counterintuitive at first glance) notion of cryptographic deniability in the first place [48].

It remains to be seen whether deniability as introduced by the present work for presignatures finds better success in the real world, but a case for this goes as follows. Messaging apps have been commodified to the point where their use is ubiquitous among laymen. This makes the legal argument against rejecting cryptographically deniable transcripts reasonable. Indeed, an average user is not expected to have the requisite technical knowledge and skills to simulate a transcript. Adaptor signature schemes, on the other hand, are sophisticated protocols used mostly in the sphere of blockchains and cryptocurrency. It is therefore prudent to assume, at least in the earliest stages of the technology's adoption, that persons engaging in these protocols are knowledgeable, somewhat well-versed in cryptography, and perfectly capable of developing practical tools for falsifying transcripts (*cf.* [12, 37]), particularly when equipped with AI tools for script generation. Such characterization of the cryptocurrency community should make the courts more hesitant to accept transcripts as evidence uncritically.

Note finally that, since this work considers deniability of a single message (presignature) as opposed to a longer transcript, deniability is maintained even if the responder employs trusted hardware [23].

8 Conclusions and Closing Remarks

Despite an implicit assumption prevalent in the literature, adaptor signature schemes are not "singletons," and more than one can be used at a time. Importantly, the different adaptor signature schemes have different security requirements in many applications, such as fair signature exchange (FSE). This is the observation that Castejon-Molina *et al.* made independently in recent work [8]. We took this idea a step further and formally demonstrated that, in a signature exchange protocol built from two adaptor signature schemes, AS_1 and AS_2, not only is witness extractability (Definition 3.6) of AS_1 unnecessary, unforgeability (!) of AS_2 is typically not required either. Furthermore, the initiator's presignature need not even be non-repudiable, and fairness and atomicity guarantees of FSE can still be upheld. This affords the initiator deniability and indeed extends the notion of fairness to encompass *accountability*, where neither party can produce incriminating evidence against the other without also incriminating themselves. We formalized this by introducing the notion of *presignature deniability* (Definition 5.1) and demonstrated that many existing constructions already satisfy it. Finally, we argued that AS_1 may, without loss of generality, be replaced with an *extended adaptor signature scheme* (Definition 6.1) EAS_1 that has the presigning algorithm take the witness as an argument as opposed to the corresponding statement. Indeed, it is the initiator that samples the ephemeral relation instance $(Y, y) \in \mathsf{R}$ relative to which the two presignatures are then issued, so they know the witness y. The unforgeability experiment defined for this setting (Figure 15) was shown to be compatible with the standard definition (Theorem 6.3), and the extended syntax together with presignature deniability was shown to admit a black-box reduction to EUF-CMA security of the underlying signature scheme (Theorem 6.5). At the very least, the latter makes presignature deniability a useful tool for theoreticians while we wait to see if it proves to be of use to practitioners as well. This depends, as discussed in Section 7, on whether the notion is welcomed by the courts and the society at large.

Acknowledgments

The authors would like to thank the anonymous reviewers at ACM ASIACCS 2027 for pointing out that the standard Schnorr adaptor signature scheme (Figure 11) already has deniable presignatures, and that the lattice-based construction of [17], while technically having deniable presignatures, is useless in practice unless the presignature $(\tilde{\sigma}, Y)$ is accompanied by a NIZK π showing that $Y \in \mathcal{L}_{\mathsf{R}}$. This NIZK then makes the triple $(\tilde{\sigma}, Y, \pi)$ non-repudiable. We are also grateful for pointing us to [5].

References

[1] N. Asokan, Victor Shoup, and Michael Waidner. 1998. Optimistic fair exchange of digital signatures. In *Advances in Cryptology — EUROCRYPT'98*, Kaisa Nyberg (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 591–606.

[2] Lukas Aumayr, Oguzhan Ersoy, Andreas Erwig, Sebastian Faust, Kristina Hostáková, Matteo Maffei, Pedro Moreno-Sanchez, and Siavash Riahi. 2021. Generalized Channels from Limited Blockchain Scripts and Adaptor Signatures. In *Advances in Cryptology – ASIACRYPT 2021*, Mehdi Tibouchi and Huaxiong Wang (Eds.). Springer International Publishing, Cham, 635–664.

[3] Ruben Baecker, Paul Gerhart, Jonathan Katz, and Dominique Schröder. 2025. Fair Exchange for Decentralized Autonomous Organizations via Threshold Adaptor Signatures. Cryptology ePrint Archive, Paper 2025/388. https://eprint.iacr.org/2025/388

[4] Vivek Bagaria, Joachim Neu, and David Tse. 2020. Boomerang: Redundancy Improves Latency and Throughput in Payment-Channel Networks. In *Financial Cryptography and Data Security*, Joseph Bonneau and Nadia Heninger (Eds.). Springer International Publishing, Cham, 304–324.

[5] Sergiu Bursuc and Sjouke Mauw. 2022. Contingent payments from two-party signing and verification for abelian groups. In *2022 IEEE 35th Computer Security Foundations Symposium (CSF)*. 195–210. doi:10.1109/CSF54842.2022.9919654

[6] Jan Camenisch and Anna Lysyanskaya. 2003. A Signature Scheme with Efficient Protocols. In *Security in Communication Networks*, Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 268–289.

[7] Ryann Cartor, Nathan Daly, Giulia Gaggero, Jason T. LeGrow, Andrea Sanguineti, and Silvia Sconza. 2026. Post-Quantum Adaptor Signatures with Strong Security from Cryptographic Group Actions. In *Post-Quantum Cryptography: 17th International Workshop, PQCrypto 2026, Saint-Malo, France, April 14–16, 2026, Proceedings, Part I* (Saint-Malo, France). Springer-Verlag, Berlin, Heidelberg, 370–405. doi:10.1007/978-3-032-22695-2_13

[8] Diego Castejon-Molina, Erkan Tairi, Dimitrios Vasilopoulos, and Pedro Moreno-Sanchez. 2026. Foundations of Verifiably Encrypted (Blind) Signatures. Cryptology ePrint Archive, Paper 2026/789. https://eprint.iacr.org/2026/789

[9] David Chaum. 1983. Blind Signatures for Untraceable Payments. In *Advances in Cryptology*, David Chaum, Ronald L. Rivest, and Alan T. Sherman (Eds.). Springer US, Boston, MA, 199–203.

[10] Liqun Chen, Caroline Kudla, and Kenneth G. Paterson. 2004. Concurrent Signatures. In *Advances in Cryptology - EUROCRYPT 2004*, Christian Cachin and Jan L. Camenisch (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 287–305.

[11] Michele Ciampi, Xiangyu Liu, Ioannis Tzannetos, and Vassilis Zikas. 2025. Universal Adaptor Signatures from Blackbox Multi-party Computation. In *Topics in Cryptology – CT-RSA 2025*, Arpita Patra (Ed.). Springer Nature Switzerland, Cham, 375–398.

[12] Daniel Collins, Simone Colombo, and Loïs Huguenin-Dumittan. 2025. Real-World Deniability in Messaging. *Proceedings on Privacy Enhancing Technologies* 2025, 1 (2025), 320–340. doi:10.56553/popets-2025-0018

[13] Wei Dai, Tatsuaki Okamoto, and Go Yamamoto. 2022. Stronger Security and Generic Constructions for Adaptor Signatures. In *Progress in Cryptology – INDOCRYPT 2022*, Takanori Isobe and Santanu Sarkar (Eds.). Springer International Publishing, Cham, 52–77.

[14] Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. 2018. Privacy Pass: Bypassing Internet Challenges Anonymously. In *Proceedings on Privacy Enhancing Technologies*, Vol. 2018. 164–180. doi:10.1515/popets-2018-0026

[15] Thaddeus Dryja. 2018. Discreet Log Contracts. https://adiabat.github.io/dlc.pdf Accessed: 7 February 2026.

[16] Andreas Erwig, Sebastian Faust, Kristina Hostáková, Monosij Maitra, and Siavash Riahi. 2021. Two-Party Adaptor Signatures from Identification Schemes. In *Public-Key Cryptography – PKC 2021: 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10–13, 2021, Proceedings, Part I*. Springer-Verlag, Berlin, Heidelberg, 451–480. doi:10.1007/978-3-030-75245-3_17

[17] Muhammed F. Esgin, Oğuzhan Ersoy, and Zekeriya Erkin. 2020. Post-Quantum Adaptor Signatures and Payment Channel Networks. In *Computer Security – ESORICS 2020*, Liqun Chen, Ninghui Li, Kaitai Liang, and Steve Schneider (Eds.). Springer International Publishing, Cham, 378–397.

[18] Lloyd Fournier. 2019. One-Time Verifiably Encrypted Signatures A.K.A. Adaptor Signatures. https://github.com/LLFourn/one-time-VES/blob/master/main.pdf.

[19] Juan A. Garay and Carl Pomerance. 2003. Timed Fair Exchange of Standard Signatures. In *Financial Cryptography*, Rebecca N. Wright (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 190–207.

[20] Paul Gerhart, Daniel Rausch, and Dominique Schröder. 2025. Universally Composable Adaptor Signatures. Cryptology ePrint Archive, Paper 2025/1363. https://eprint.iacr.org/2025/1363

[21] Paul Gerhart, Dominique Schröder, Pratik Soni, and Sri AravindaKrishnan Thyagarajan. 2024. Foundations of Adaptor Signatures. 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* (Zurich, Switzerland). Springer-Verlag, Berlin, Heidelberg, 161–189. doi:10.1007/978-3-031-58723-8_6

[22] Noemi Glaeser, Matteo Maffei, Giulio Malavolta, Pedro Moreno-Sanchez, Erkan Tairi, and Sri AravindaKrishnan Thyagarajan. 2022. Foundations of Coin Mixing Services. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 1259–1273. doi:10.1145/3548606.3560637

[23] Lachlan J. Gunn, Ricardo Vieitez Parra, and N. Asokan. 2019. Circumventing Cryptographic Deniability with Remote Attestation. In *Proceedings on Privacy Enhancing Technologies (PETS) (Proceedings on Privacy Enhancing Technologies, Vol. 2019)*. Stockholm, Sweden, 350–369. doi:10.2478/popets-2019-0051 Privacy Enhancing Technologies Symposium (PETS 2019), 16–20 July 2019.

[24] Hossein Hafezi, Aditi Partap, Sourav Das, and Joseph Bonneau. 2025. Fair Signature Exchange. Cryptology ePrint Archive, Paper 2025/059. https://eprint.iacr.org/archive/2025/059/1736864394.pdf Initial draft from 14 January 2025.

[25] Lucjan Hanzlik, Julian Loss, Sri AravindaKrishnan Thyagarajan, and Benedikt Wagner. 2024. Sweep-UC: Swapping Coins Privately. In *2024 IEEE Symposium on Security and Privacy (SP)*. 3822–3839. doi:10.1109/SP54263.2024.00081

[26] Pavel Hubáček, Kristýna Mašková, and Berenika Richterová. 2026. Practical Adaptor Signatures: Security and Applications. Cryptology ePrint Archive, Paper 2026/1482. https://eprint.iacr.org/2026/1482

[27] Pratima Jana, Surbhi Shaw, and Ratna Dutta. 2025. Compact Adaptor Signature from Isogenies with Enhanced Security. In *Cryptology and Network Security*, Markulf Kohlweiss, Roberto Di Pietro, and Alastair Beresford (Eds.). Springer Nature Singapore, Singapore, 77–100.

[28] Jean Belo Klamti and M. Anwar Hasan. 2022. Post-Quantum Two-Party Adaptor Signature Based on Coding Theory. *Cryptography* 6, 1 (2022). doi:10.3390/cryptography6010006

[29] Marek Klonowski, Mirosław Kutyłowski, Anna Lauks, and Filip Zagórski. 2005. Conditional Digital Signatures. In *Trust, Privacy, and Security in Digital Business*, Sokratis Katsikas, Javier López, and Günther Pernul (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 206–215.

[30] Jacob Leiken and Sunoo Park. 2025. On the Credibility of Deniable Communication in Court. Cryptology ePrint Archive, Paper 2025/1949. https://eprint.iacr.org/2025/1949

[31] Xiangyu Liu, Ioannis Tzannetos, and Vassilis Zikas. 2024. Adaptor Signatures: New Security Definition and a Generic Construction for NP Relations. In *Advances in Cryptology – ASIACRYPT 2024: 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9–13, 2024, Proceedings, Part II* (Kolkata, India). Springer-Verlag, Berlin, Heidelberg, 168–193. doi:10.1007/978-981-96-0888-1_6

[32] Xiangyu Liu, Riccardo Zanotto, and Vassilis Zikas. 2026. Non-Interactive Atomic Swaps from Standard Signatures: Lower Bounds and Constructions. Cryptology ePrint Archive, Paper 2026/2277. https://eprint.iacr.org/2026/2277

[33] Varun Madathil, Sri AravindaKrishnan Thyagarajan, Dimitrios Vasilopoulos, Lloyd Fournier, Giulio Malavolta, and Pedro Moreno-Sanchez. 2023. Cryptographic Oracle-Based Conditional Payments. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. The Internet Society. doi:10.14722/ndss.2023.23024

[34] Giulio Malavolta, Pedro Moreno-Sanchez, Clara Schneidewind, Aniket Kate, and Matteo Maffei. 2019. Anonymous Multi-Hop Locks for Blockchain Scalability and Interoperability. In *26th Annual Network and Distributed System Security Symposium (NDSS 2019)*. San Diego, CA, USA. doi:10.14722/ndss.2019.23330

[35] Andrew Poelstra. 2017. Scriptless Scripts. https://www.sosthene.net/wp-content/uploads/wpforo/default_attachments/1562671969-Poelstra-Scriptless-Scripts.pdf. Accessed: 19 April 2026.

[36] Xianrui Qin, Shimin Pan, Arash Mirzaei, Zhimei Sui, Oğuzhan Ersoy, Amin Sakzad, Muhammed F. Esgin, Joseph K. Liu, Jiangshan Yu, and Tsz Hon Yuen. 2023. BlindHub: Bitcoin-Compatible Privacy-Preserving Payment Channel Hubs Supporting Variable Amounts. In *2023 IEEE Symposium on Security and Privacy (SP)*. 2462–2480. doi:10.1109/SP46215.2023.10179427

[37] Nathan Reitinger, Nathan Malkin, Omer Akgul, Michelle L. Mazurek, and Ian Miers. 2023. Is Cryptographic Deniability Sufficient? Non-Expert Perceptions of Deniability in Secure Messaging. In *2023 IEEE Symposium on Security and Privacy (SP)*. 274–292. doi:10.1109/SP46215.2023.10179361

[38] Paul Rösler, Katharina Schambach, and Julian Thomas. 2026. Atomic Swaps with Relaxed Assumptions. *IACR Communications in Cryptology* 3, 1 (2026). doi:10.62056/abe0ivol7

[39] Erkan Tairi, Pedro Moreno-Sanchez, and Matteo Maffei. 2021. A2L: Anonymous Atomic Locks for Scalability in Payment Channel Hubs. In *2021 IEEE Symposium on Security and Privacy (SP)*. 1834–1851. doi:10.1109/SP40001.2021.00111

[40] Erkan Tairi, Pedro Moreno-Sanchez, and Matteo Maffei. 2021. Post-Quantum Adaptor Signature for Privacy-Preserving Off-Chain Payments. In *Financial Cryptography and Data Security*, Nikita Borisov and Claudia Diaz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 131–150.

[41] Erkan Tairi, Pedro Moreno-Sanchez, and Clara Schneidewind. 2023. LedgerLocks: A Security Framework for Blockchain Protocols Based on Adaptor Signatures. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 859–873. doi:10.1145/3576915.3623149

[42] Sri AravindaKrishnan Thyagarajan and Giulio Malavolta. 2021. Lockable Signatures for Blockchains: Scriptless Scripts for All Signatures. In *2021 IEEE Symposium on Security and Privacy (SP)*. 937–954. doi:10.1109/SP40001.2021.00065

[43] Sri AravindaKrishnan Thyagarajan, Giulio Malavolta, and Pedro Moreno-Sanchez. 2022. Universal Atomic Swaps: Secure Exchange of Coins Across All Blockchains. In *IEEE Symposium on Security and Privacy (SP)*. IEEE.

[44] Sri AravindaKrishnan Thyagarajan, Easwar Vivek Mangipudi, Lucjan Hanzlik, Aniket Kate, and Pratyay Mukherjee. 2025. Vitārit: Paying for Threshold Services on Bitcoin and Friends. In *2025 IEEE Symposium on Security and Privacy (SP)*. 2018–2036. doi:10.1109/SP61157.2025.00170

[45] Binbin Tu, Min Zhang, and Chen Yu. 2022. Efficient ECDSA-Based Adaptor Signature for Batched Atomic Swaps. In *Information Security*, Willy Susilo, Xiaofeng Chen, Fuchun Guo, Yudi Zhang, and Rolly Intan (Eds.). Springer International Publishing, Cham, 175–193.

[46] Nikhil Vanjani, Garrett Greiner, Sri AravindaKrishnan Thyagarajan, and Pratik Soni. 2026. New Constructions of Functional Adaptor Signatures: Broader Functions and Improved Efficiency. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. IEEE.

[47] Nikhil Vanjani, Pratik Soni, and Sri AravindaKrishnan Thyagarajan. 2024. Functional Adaptor Signatures: Beyond All-or-Nothing Blockchain-based Payments. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (Salt Lake City, UT, USA) *(CCS '24)*. Association for Computing Machinery, New York, NY, USA, 1493–1507. doi:10.1145/3658644.3690240

[48] Tarun Kumar Yadav, Devashish Gosain, and Kent Seamons. 2023. Cryptographic deniability: a multi-perspective study of user perceptions and expectations. In *Proceedings of the 32nd USENIX Conference on Security Symposium* (Anaheim, CA, USA) *(SEC '23)*. USENIX Association, USA, Article 204, 18 pages.

A Open Science

No artifact is needed to evaluate the paper's core contributions.

B Related Work

Adaptor signatures are introduced informally by Poelstra in [35], and then formalized (with increasing success) in [2, 13, 18, 21, 31]. Erwig *et al.* [16] abstract the properties of Schnorr signatures and show that any signature scheme built from an identification protocol that additionally enjoys some homomorphic properties admits an adaptor signature scheme. Privacy of adaptor signatures is first considered by Dai *et al.* [13] who formalize the notion of *unlinkability* (alluded to already by Poelstra [35]), which requires that adapted presignatures be indistinguishable from normally produced signatures. Liu *et al.* [31] then introduce a weaker property, implied by unlinkability, called *witness hiding*. Finally, they construct *universal*

adaptor signatures for arbitrary NP relations using a Karp reduction to the Hamiltonian-cycle problem. Ciampi *et al.* [11] optimize universal adaptor signatures using the MPC-in-the-head paradigm. Liu *et al.* in [32] study non-interactive atomic swaps built with adaptor signatures and introduce the notion of *faithful extraction*, which requires that adaptation followed by extraction yield the same witness as initially used for the adaptation. Tairi *et al.* in [41] and then Gerhart *et al.* in [20] study adaptor signature schemes *vis-à-vis universal composability*. Post-quantum realizations of adaptor signatures are proposed in [7, 17, 27, 28, 40].

Qin *et al.* [36] define *blind adaptor signatures* and sketch an ECDSA-based construction. Malavolta *et al.* in [34] define *anonymous multi-hop locks* as a generalization of adaptor signature-based locking in payment-channel networks. Thyagarajan and Malavolta [42] introduce *lockable signatures* as a weaker variant of adaptor signatures, where presigning is impossible without knowing the witness for the underlying hard relation. Tairi *et al.* in [39] introduce a payment channel hub protocol built from adaptor signatures. Their construction is later revised by Glaeser *et al.* in [22] who introduce blind conditional signatures and instantiate them using adaptor signatures. Tu *et al.* [45] optimize ECDSA-based adaptor signatures introduced by Aumayr *et al.* [2] to support batched atomic swaps. Bursuc and Mauw [5] introduce *weak adaptor signature schemes*, whose presigning algorithm takes the witness as input, just as in our extended adaptor signature schemes, but which also use non-standard preverification. In their paper, Bursuc and Mauw focus on the construction and formal verification of *zero-knowledge contingent payments* from weak adaptor signatures. Madathil *et al.* [33] introduce *verifiable witness encryption based on threshold signatures* and instantiate it with adaptor signatures. Bagaria *et al.* [4] devise a "straggler mitigation" scheme in payment-channel networks and propose instantiating it with adaptor signatures with threshold adaptation. Thyagarajan *et al.* [43] build an atomic swap protocol using adaptor signatures, with Hanzlik *et al.* [25] later describing a privacy-preserving version. Baecker *et al.* [3] introduce *threshold adaptor signatures* which are a hybrid of adaptor signature schemes and threshold signatures. Thyagarajan *et al.* in [44] leverage adaptor signatures to construct protocols for fair payment for threshold services. Vanjani *et al.* [47] introduce *functional adaptor signatures*, where the extracting algorithm (see Definition 3.1) does not yield a witness y directly but a function of it instead. Their primitive is revisited in [46]. Rösler *et al.* [38] construct generic n-to-m atomic swaps using adaptor signatures. Klonowski *et al.* [29] describe a *conditional signature scheme* in which Alice produces a presignature $\tilde{\sigma}$ that may be turned (adapted) into a regular signature on a message m provided some other message m' is signed by Bob. In their construction, $\tilde{\sigma}$ is itself indistinguishable from a random string and any relation to Bob or the message m must be proved separately by Alice. Furthermore, a setup procedure is required, whereby Bob issues commitments to his signatures ahead of time.

Asokan *et al.* [1] formalize fair signature exchange (FSE) in the setting where a trusted third party is available. Notably, the main role of this party is preventing premature aborts, an issue we sidestep in this work as not relevant to the most common setting of both FSE and adaptor signatures, namely, cryptocurrencies. Chen *et al.* [10] introduce *concurrent signatures* which both address the problem of fair signature exchange and are similar in spirit to deniable presignatures. In a concurrent signature scheme, Alice and Bob first exchange two *ambiguous signatures*, each of which could have been produced by either Alice or Bob (note how this is a weaker deniability notion than considered in this work). Then, once a special *keystone* is revealed, the two signatures cease being ambiguous and can be henceforth uniquely attributed to their authors. Garay and Pomerance [19] introduce an approach for gradual exchanging of signatures such that an abort at any time does not give the aborting party any significant advantage.

Real-world attitudes toward cryptographic deniability, in particular its social and legal acceptance, are surveyed in [12, 30, 37, 48]. In concurrent work [8], Castejon-Molina *et al.* also observe that, in fair signature exchange, extractability is not required for one of the adaptor signature schemes (*cf.* Theorem 4.6) and define *non-extractable adaptor signature schemes*. They also observe the tension between blind signatures and extractability.

C Preliminaries

Definition C.1. A relation $\mathsf{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ is *hard* if

(1) there exists an efficient algorithm Gen_R which, on input 1^λ, outputs a pair $(x, w) \in \mathsf{R}$, and
(2) for all efficient algorithms $\mathcal{A}$, there exists a negligible function μ such that

$$\Pr\left[(x, w') \in \mathsf{R} \;\middle|\; \begin{matrix}(x, w) \leftarrow\!\$\ \mathsf{Gen}_\mathsf{R}(1^\lambda) \\ w' \leftarrow\!\$\ \mathcal{A}(x)\end{matrix}\right] \leq \mu(\lambda)\,,$$

i.e., the probability that an efficient algorithm produces a valid *witness* w' for a random *statement* x of R is negligible.

Definition C.2 (Signature Scheme). A *signature scheme* Σ is a tuple of efficient algorithms

$$\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$$

such that

- the *key generation* algorithm KeyGen, on input the security parameter λ in unary, outputs a pair $(\mathsf{pk}, \mathsf{sk})$,
- the *signing* algorithm Sign, on input a secret key sk and message m, outputs a signature σ,
- the *verification* algorithm Verify, on input a public key pk, message m, and signature σ, outputs a bit b.

Definition C.3 (Signature Scheme Correctness). A signature scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is *correct*, if, for all $\lambda \in \mathbb{N}$ and all messages m,

$$\Pr\left[\mathsf{Verify}(\mathsf{pk}, m, \sigma) = 1 \;\middle|\; \begin{matrix}(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda) \\ \sigma \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}, m)\end{matrix}\right] = 1\,.$$

Definition C.4 (Existential Unforgeability). A signature scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is *existentially unforgeable under adaptive chosen message attack*, if, for all efficient algorithms $\mathcal{A}$, there exists a negligible function μ such that

$$\Pr[\mathsf{SigForge}_{\mathcal{A},\Sigma}(\lambda) = 1] \leq \mu(\lambda)\,,$$

where the experiment $\mathsf{SigForge}_{\mathcal{A},\Sigma}$ is described in Figure 17, and the probability is taken over the random coin tosses of the algorithms involved.

$\underline{\mathsf{SigForge}_{\mathcal{A},\Sigma}(\lambda)}$

$Q \leftarrow \emptyset$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
$(m^*, \sigma^*) \leftarrow\!\$\ \mathcal{A}^{O_{\mathsf{Sign}}(\mathsf{sk},\cdot)}(\mathsf{pk})$
return $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*)$
$\wedge\ (m^* \notin Q)$

$\underline{O_{\mathsf{Sign}}(\mathsf{sk}, m)}$

$\sigma \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}, m)$
$Q \leftarrow Q \cup \{m\}$
return σ

Figure 17: Existential unforgeability (EUF-CMA) experiment for signatures

$\underline{\mathsf{SigForge}'_{\mathcal{A},\Sigma}(\lambda)}$

$Q \leftarrow \emptyset$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
$(m^*, \sigma^*) \leftarrow\!\$\ \mathcal{A}^{O_{\mathsf{Sign}}(\mathsf{sk},\cdot)}(\mathsf{pk})$
return $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*)$
$\wedge\ \big((m^*, \sigma^*) \notin Q\big)$

$\underline{O_{\mathsf{Sign}}(\mathsf{sk}, m)}$

$\sigma \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}, m)$
$Q \leftarrow Q \cup \{(m, \sigma)\}$
return σ

Figure 18: Strong unforgeability (SUF-CMA) experiment for signatures

Definition C.5 (Strong Unforgeability). A signature scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is *strongly unforgeable under adaptive chosen message attack*, if, for all efficient algorithms $\mathcal{A}$, there exists a negligible function μ such that

$$\Pr[\mathsf{SigForge}'_{\mathcal{A},\Sigma}(\lambda) = 1] \leq \mu(\lambda)\,,$$

where the experiment $\mathsf{SigForge}'_{\mathcal{A},\Sigma}$ is described in Figure 18, and the probability is taken over the random coin tosses of the algorithms involved.

D More on Extended Adaptor Signature Schemes

Definition D.1 (Extended Presignature Correctness). An extended adaptor signature scheme EAS satisfies *presignature correctness*, if, for all $\lambda \in \mathbb{N}$ and all messages m,

$$\Pr\left[\begin{array}{r|l} \mathsf{PreVerify}(\mathsf{pk}, m, \tilde{\sigma}, Y) = 1 & (\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda) \\ \mathsf{Verify}(\mathsf{pk}, m, \sigma) = 1 & (Y, y) \leftarrow\!\$\ \mathsf{Gen}_{\mathsf{R}}(1^\lambda) \\ (Y, y') \in \mathsf{R}' & \tilde{\sigma} \leftarrow\!\$\ \mathsf{PreSign}(\mathsf{sk}, m, y) \\ & \sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, m, \tilde{\sigma}, y) \\ & y' \leftarrow \mathsf{Extract}(\mathsf{pk}, m, \tilde{\sigma}, \sigma, Y) \end{array}\right] = 1\,,$$

where the probability is taken over the random coin tosses of the algorithms involved.

Definition D.2 (Extended Presignature Adaptability). An extended adaptor signature scheme EAS satisfies *presignature adaptability*, if, for all $\lambda \in \mathbb{N}$, all messages m, all presignatures $\tilde{\sigma}$, all relation instances $(Y, y) \in \mathsf{R}'$, and all public keys pk, we have

$$\big(\mathsf{PreVerify}(\mathsf{pk}, m, \tilde{\sigma}, Y) = 1\big) \implies \big(\mathsf{Verify}(\mathsf{pk}, m, \mathsf{Adapt}(\mathsf{pk}, m, \tilde{\sigma}, y)) = 1\big)\,.$$

$\underline{\mathsf{pReal}_{\mathcal{D},\mathsf{EAS}}(\lambda)}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
return $\mathcal{D}^{\mathsf{Sign}(\mathsf{sk},\cdot), O(\lambda,\mathsf{sk},\cdot)}(\mathsf{pk})$

$\underline{O(\lambda, \mathsf{sk}, m)}$

$(Y, y) \leftarrow\!\$\ \mathsf{Gen}_{\mathsf{R}}(1^\lambda)$
$\tilde{\sigma} \leftarrow\!\$\ \mathsf{PreSign}(\mathsf{sk}, m, y)$
return $(\tilde{\sigma}, Y)$

$\underline{\mathsf{pSim}_{\mathcal{D},\mathsf{Sim},\mathsf{EAS}}(\lambda)}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
return $\mathcal{D}^{\mathsf{Sign}(\mathsf{sk},\cdot), \mathsf{Sim}(\mathsf{pk},\cdot)}(\mathsf{pk})$

Figure 19: Presignature deniability experiments for an extended adaptor signature scheme

Definition D.3 (Extended Adaptor Signature Scheme Correctness). An extended adaptor signature scheme EAS is *correct* if it satisfies extended presignature correctness and extended presignature adaptability.

Definition D.4 (Extended Presignature Deniability). An extended adaptor signature scheme $\mathsf{EAS}_{\Sigma,\mathsf{R},\mathsf{R}'}$ *has deniable presignatures* if there exists an efficient algorithm Sim such that, for all efficient distinguishers $\mathcal{D}$, there exists a negligible function μ such that

$$\big|\Pr[\mathsf{pReal}_{\mathcal{D},\mathsf{EAS}}(\lambda) = 1] - \Pr[\mathsf{pSim}_{\mathcal{D},\mathsf{Sim},\mathsf{EAS}}(\lambda) = 1]\big| \leq \mu(\lambda)\,.$$

where the experiments pReal and pSim are described in Figure 19, and the probabilities are taken over the random coin tosses of the algorithms involved.

E Proofs

E.1 Proof of Theorem 4.6

Lemma E.1. *The construction in Figure 9 is a correct signature exchange protocol.*

Proof. That $\sigma_{\mathfrak{R}}$ returned by IFinal is correct, *i.e.*, passes verification relative to Σ_2, follows immediately from presignature correctness (Definition 3.2) of AS_2.

For the other signature, $\sigma_{\mathfrak{I}}$, what follows from presignature correctness of AS_1 is only that $\mathsf{PreVerify}_1(\mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{I}}, \tilde{\sigma}_{\mathfrak{I}}, Y) = 1$. Presignature correctness of AS_2 then also gives us that $(Y, y') \in \mathsf{R}'$. Since, in general, $y \neq y'$, we must also leverage presignature adaptability (Definition 3.3) of AS_1 to argue that $\sigma_{\mathfrak{I}}$ adapted from $\tilde{\sigma}_{\mathfrak{I}}$ with y' verifies relative to Σ_1. □

Lemma E.2. *The construction in Figure 9 satisfies responder fairness.*

Proof. We reduce responder fairness to witness extractability (WitExt) of AS_2 while also relying on presignature adaptability of AS_1. Let $\mathcal{A}$ be an adversary breaking responder fairness of the construction in Figure 9, denoted SE, such that

$$\Pr[\mathsf{FairnessResp}_{\mathcal{A},\mathsf{SE}}(\lambda) = 1] = \alpha\,.$$

We construct, in Figure 20, a reduction $\mathcal{B}$ that wins the $\mathsf{WitExt}_{\mathcal{B},\mathsf{AS}_2}$ experiment (Figure 3) with probability α. Since we assume AS_2 is witness-extractable, it follows that α must be negligible, and so the construction in Figure 9 must be responder-fair. We simulate for $\mathcal{A}$ the responder oracle $O'_{\mathfrak{R}}$ and the signing oracle O'_{Sign} by using $\mathcal{B}$'s own oracles. We have the reduction maintain the query sets $\mathcal{T}_{\mathcal{B}}$ and $\mathcal{S}_{\mathcal{B}}$ to help the reader keep track of the properties of (m^*, σ^*) returned by $\mathcal{A}$. Note that they correspond to the sets $\mathcal{T}$ and $\mathcal{S}$ maintained by $\mathcal{B}$'s own oracles (see Figure 3). Also, observe that the set $\mathcal{T}$ (and, by extension, $\mathcal{T}_{\mathcal{B}}$) tracks FSE sessions *opened* by the (malicious) initiator.

Note that our simulation is perfect, *i.e.*, the view $\mathcal{A}$ has when run as a subroutine of $\mathcal{B}$ is identically distributed to the view it has in the experiment $\mathsf{FairnessResp}_{\mathcal{A},\mathsf{SE}}$. We therefore have that, with probability α, $\mathcal{A}$ (and, by extension, $\mathcal{B}$) returns (m^*, σ^*) such that, at the same time, $\mathsf{Verify}_2(\mathsf{pk}, m^*, \sigma^*)$, $m^* \notin \mathcal{S}_{\mathcal{B}}$, and

$$\big(\forall(\mathsf{st}_{\mathfrak{R}}, m_{\mathfrak{I}}, \mathsf{pk}_{\mathfrak{I}}) \in \mathcal{T}_{\mathcal{B}}[m^*]\big) \\ \big(\mathsf{Verify}_1(\mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{I}}, \mathsf{RFinal}(\mathsf{st}_{\mathfrak{R}}, \sigma^*)) \neq 1\big)\,.$$

The reduction $\mathcal{B}$, on the other hand, wins the $\mathsf{WitExt}_{\mathcal{B},\mathsf{AS}_2}$ experiment if, at the same time, $\mathsf{Verify}_2(\mathsf{pk}, m^*, \sigma^*)$, $m^* \notin \mathcal{S}$, and

$$\big(\forall(Y, \tilde{\sigma}^*) \in \mathcal{T}[m^*]\big)\big((Y, \mathsf{Extract}_2(\mathsf{pk}, m^*, \tilde{\sigma}^*, \sigma^*, Y)) \notin \mathsf{R}'\big)\,.$$

Since $\mathcal{S}_{\mathcal{B}} = \mathcal{S}$ after every call to O'_{Sign}, the first two clauses of both victory conditions describe the same event, *i.e.*, they are satisfied for $\mathcal{B}$ with probability α. Focus on the last clause in either condition. There are two possible cases for the clause in $\mathcal{A}$'s condition to be true: either (1) $\mathcal{T}_{\mathcal{B}}[m^*] = \emptyset$, or (2) not, and, for all $(\mathsf{st}_{\mathfrak{R}}, m_{\mathfrak{I}}, \mathsf{pk}_{\mathfrak{I}})$ in $\mathcal{T}_{\mathcal{B}}[m^*]$, the responder does not get a valid signature on m^* in any session.

In case (1), the oracle O_{PreSign} has not been queried on m^* (otherwise, $\mathcal{T}_{\mathcal{B}}[m^*]$ would not be empty), and so $\mathcal{T}[m^*]$ is also empty. In case (2), for every tuple $(\mathsf{st}_{\mathfrak{R}}, m_{\mathfrak{I}}, \mathsf{pk}_{\mathfrak{I}}) \in \mathcal{T}_{\mathcal{B}}[m^*]$, we have

$$\mathsf{Verify}_1\big(\mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{I}}, \mathsf{RFinal}(\mathsf{st}_{\mathfrak{R}}, \sigma^*)\big) \neq 1\,.$$

Focus on one such tuple. Recall from Figure 9 that the state $\mathsf{st}_{\mathfrak{R}}$ is a tuple $(\mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}}, \rho_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{I}}, \rho_{\mathfrak{I}})$. Furthermore, $\rho_{\mathfrak{I}}$ can be parsed as $(\tilde{\sigma}_{\mathfrak{I}}, Y)$, and we checked in RPrep that $Y \in \mathcal{L}_{\mathsf{R}}$. Next, $\mathsf{RFinal}(\mathsf{st}_{\mathfrak{R}}, \sigma^*)$ first extracts $y' \leftarrow \mathsf{Extract}_2(\mathsf{pk}, m^*, \rho_{\mathfrak{R}}, \sigma^*, Y)$, then adapts $\tilde{\sigma}_{\mathfrak{I}}$ using y' into a signature $\sigma_{\mathfrak{I}}$ and returns it. There are two possibilities here. Suppose that $(Y, y') \in \mathsf{R}'$, but verification $\mathsf{Verify}_1(\mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{I}}, \sigma_{\mathfrak{I}})$ nevertheless fails. This contradicts presignature adaptability (Definition 3.3) of AS_1. Indeed, since m^* has a "row" in $\mathcal{T}_{\mathcal{B}}$, it means that the initiator message $m_{\mathfrak{I}}$ associated with the same session passed preverification against $\mathsf{pk}_{\mathfrak{I}}$, $\tilde{\sigma}_{\mathfrak{I}}$, and Y. Therefore, RFinal adapted a preverifiable $\tilde{\sigma}_{\mathfrak{I}}$ using a valid witness y'; hence, $\sigma_{\mathfrak{I}}$ is verifiable, a contradiction. Therefore, $(Y, y') \notin \mathsf{R}'$, and this holds for every entry in $\mathcal{T}_{\mathcal{B}}[m^*]$. And so, whenever $\mathcal{A}$ wins in the responder fairness experiment, which happens with probability α, $\mathcal{B}$ wins in the witness extractability experiment. We get

$$\Pr[\mathsf{WitExt}_{\mathcal{B},\mathsf{AS}_2}(\lambda) = 1] \geq \Pr[\mathsf{FairnessResp}_{\mathcal{A},\mathsf{SE}}(\lambda) = 1] = \alpha\,,$$

and, since we assumed AS_2 is witness-extractable, it follows that α is negligible.

□

Lemma E.3. *The construction in Figure 9 satisfies initiator fairness.*

Proof. We reduce initiator fairness to existential unforgeability (aEUF-CMA) of AS_1, while also relying on presignature adaptability of AS_2. Let $\mathcal{A}$ be an adversary breaking initiator fairness of the construction in Figure 9, denoted SE, such that

$$\Pr[\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}(\lambda) = 1] = \alpha\,.$$

We construct, in Figure 21, a reduction $\mathcal{B}$ that wins the $\mathsf{aSigForge}_{\mathcal{B},\mathsf{AS}_1}$ experiment (Figure 2) with probability smaller than α by at most a polynomial factor. Since we assume AS_1 is unforgeable, it follows that α is negligible, and so the construction in Figure 9 is initiator-fair. We simulate for $\mathcal{A}$ the two initiator oracles $O'^1_{\mathfrak{I}}$ and $O'^2_{\mathfrak{I}}$, as well as the signing oracle O'_{Sign}, using calls to $\mathcal{B}$'s own oracles. We have the reduction maintain the query sets $\mathcal{T}_{\mathcal{B}}$ and $\mathcal{S}_{\mathcal{B}}$ to help the reader keep track of the properties of (m^*, σ^*) returned by $\mathcal{A}$. Observe that the set $\mathcal{T}_{\mathcal{B}}$ tracks FSE sessions *closed* by the (malicious) responder, and, after each call made by $\mathcal{A}$ to O'_{Sign}, the set $\mathcal{S}_{\mathcal{B}} = \mathcal{S}$, maintained by $\mathcal{B}$'s own oracle O_{Sign}.

The intuition behind the reduction is as follows. We simulate for $\mathcal{A}$ the ability to open FSE sessions via the oracle $O'^1_{\mathfrak{I}}$ using $\mathcal{B}$'s own presigning oracle O_{PreSign}. We want the pair (m^*, σ^*) eventually returned by $\mathcal{A}$ to be a winning pair for the existential unforgeability experiment (Figure 2), which, in particular, means that all queries to O_{PreSign} about m^* must be relative to statements Y sampled honestly via $\mathcal{B}$'s relation oracle O_{R} (*cf.* the $\big(\forall(Y, \tilde{\sigma}) \in \mathcal{T}[m^*]\big)(Y \in C)$ clause in Figure 2). However, if we were to sample all statements Y for O_{PreSign} this way, we would have, for lack of witnesses y, no hope of ever closing the corresponding FSE sessions, which $\mathcal{A}$ may want to do by calling $O'^2_{\mathfrak{I}}$. For this reason, we first argue that, if $\mathcal{A}$ would win $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$ by returning m^*, then $\mathcal{A}$ must not have queried $O^2_{\mathfrak{I}}$ about a session i involving m^* as the initiator message. Given this, we have the reduction *guess*, roughly speaking, a session in which to "embed" a challenge statement Y. The reduction will not be able to close this session, but if $\mathcal{A}$ is to be successful, it will not try closing any session involving m^*, so, if we guess correctly, $\mathcal{A}$'s view when run as $\mathcal{B}$'s subroutine will be identically distributed to its view in $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$. Since $\mathcal{A}$ opens at most a polynomial number of sessions, the security loss due to guessing will also be at most a polynomial factor. Two technical issues have to be taken into account here. First, $\mathcal{A}$ may return m^* that it never queried $O'^1_{\mathfrak{I}}$ on, in which case we must be able to close *every* session initiated by $\mathcal{A}$ to simulate the view $\mathcal{A}$ has in the $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$ experiment. Second, $\mathcal{A}$ may open multiple sessions with the same initiator message $m_{\mathfrak{I}}$. Therefore, instead of guessing a session at random, we must really guess a message at random and hope the message we guessed is the one eventually returned as m^*. We now proceed to make this intuition precise.

Observe first the following: if $\mathcal{A}$ wins $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$ by returning m^*, then at no point in the experiment could it have *successfully* queried $O^2_{\mathfrak{I}}$ on a session i involving $m_{\mathfrak{I}} = m^*$ and got anything other than $\perp$. In other words, $\mathcal{T}[m^*] = \emptyset$, since $\mathcal{T}[m^*]$ (in the $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$ game that $\mathcal{A}$ is expecting to be playing, *not* in $\mathsf{aSigForge}_{\mathcal{B},\mathsf{AS}_1}$) tracks closed sessions involving m^* as the initiator message. Indeed, assume toward contradiction there was such a session i involving m^* as the initiator message, and $\mathcal{A}$ queried $O^2_{\mathfrak{I}}(\rho_{\mathfrak{R}}, i)$ successfully. Since $O^2_{\mathfrak{I}}(\rho_{\mathfrak{R}}, i)$ did not return $\perp$,

$\mathcal{B}^{O_{\mathsf{Sign}}(\cdot),O_{\mathsf{PreSign}}(\cdot,\cdot)}(\mathsf{pk})$

$\mathcal{T}_{\mathcal{B}} \leftarrow \emptyset$
$\mathcal{S}_{\mathcal{B}} \leftarrow \emptyset$
$(m^*, \sigma^*) \leftarrow\!\$\ \mathcal{A}^{O'_{\mathfrak{R}}(\cdot,\cdot,\cdot,\cdot),O'_{\mathsf{Sign}}(\cdot)}(\mathsf{pk})$
return (m^*, σ^*)

$O'_{\mathfrak{R}}(\mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{R}}, m_{\mathfrak{I}}, \rho_{\mathfrak{I}})$

$(\tilde{\sigma}_{\mathfrak{I}}, Y) \leftarrow \rho_{\mathfrak{I}}$
if $Y \notin \mathcal{L}_{\mathsf{R}}$ **then**
  return $\bot$
if $\mathsf{PreVerify}_1(\mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{I}}, \tilde{\sigma}_{\mathfrak{I}}, Y) \neq 1$ **then**
  return $\bot$
$\rho_{\mathfrak{R}} \leftarrow\!\$\ O_{\mathsf{PreSign}}(m_{\mathfrak{R}}, Y)$
$\mathsf{st}_{\mathfrak{R}} \leftarrow (\mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}}, \rho_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{I}}, m_{\mathfrak{I}}, \rho_{\mathfrak{I}})$
$\mathcal{T}_{\mathcal{B}}[m_{\mathfrak{R}}] \leftarrow \mathcal{T}_{\mathcal{B}}[m_{\mathfrak{R}}] \cup \{(\mathsf{st}_{\mathfrak{R}}, m_{\mathfrak{I}}, \mathsf{pk}_{\mathfrak{I}})\}$
return $\rho_{\mathfrak{R}}$

$O'_{\mathsf{Sign}}(m)$

$\sigma \leftarrow\!\$\ O_{\mathsf{Sign}}(m)$
$\mathcal{S}_{\mathcal{B}} \leftarrow \mathcal{S}_{\mathcal{B}} \cup \{m\}$
return σ

Figure 20: Reduction of responder fairness to witness extractability of AS_2

$\mathcal{B}^{O_{\mathsf{Sign}}(\cdot),O_{\mathsf{PreSign}}(\cdot,\cdot),O_{\mathsf{R}}()}(\mathsf{pk})$

$\mathcal{T}_{\mathcal{B}} \leftarrow \emptyset$
$\mathcal{S}_{\mathcal{B}} \leftarrow \emptyset$
$\mathcal{M} \leftarrow \emptyset$
$i_s \leftarrow 1$
$i_m \leftarrow 1$
$j \leftarrow\!\$\ [n]$
$(m^*, \sigma^*) \leftarrow\!\$\ \mathcal{A}^{O'^1_{\mathfrak{I}}(\lambda,\cdot,\cdot,\cdot),O'^2_{\mathfrak{I}}(\cdot,\cdot),O'_{\mathsf{Sign}}(\cdot)}(\mathsf{pk})$
return (m^*, σ^*)

$O'^1_{\mathfrak{I}}(\lambda, m_{\mathfrak{I}}, \mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}})$

if $\mathcal{M}[m_{\mathfrak{I}}] = \bot$ **then**
  $\mathcal{M}[m_{\mathfrak{I}}] \leftarrow i_m$
  $i_m \leftarrow i_m + 1$
if $\mathcal{M}[m_{\mathfrak{I}}] = j$ **then**
  $Y \leftarrow\!\$\ O_{\mathsf{R}}()$
  $y \leftarrow \bot$
else
  $(Y, y) \leftarrow\!\$\ \mathsf{Gen}_{\mathsf{R}}(1^\lambda)$
$\tilde{\sigma}_{\mathfrak{I}} \leftarrow\!\$\ O_{\mathsf{PreSign}}(m_{\mathfrak{I}}, Y)$
$\rho_{\mathfrak{I}} \leftarrow (\tilde{\sigma}_{\mathfrak{I}}, Y)$
$\mathsf{st}_{\mathfrak{I}} \leftarrow ((Y, y), \mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}})$
$\mathsf{session}[i_s] \leftarrow (\mathsf{st}_{\mathfrak{I}}, m_{\mathfrak{I}})$
$i_s \leftarrow i_s + 1$
return $\rho_{\mathfrak{I}}$

$O'^2_{\mathfrak{I}}(\rho_{\mathfrak{R}}, i_s)$

if $\mathsf{session}[i_s] = \bot$ **then**
  return $\bot$
$(\mathsf{st}_{\mathfrak{I}}, m_{\mathfrak{I}}) \leftarrow \mathsf{session}[i_s]$
$(\tau, \mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}}) \leftarrow \mathsf{st}_{\mathfrak{I}}$
$(Y, y) \leftarrow \tau$
if $y = \bot$ **then**
  return $\bot$
$\sigma_{\mathfrak{R}} \leftarrow\!\$\ \mathsf{IFinal}(\mathsf{st}_{\mathfrak{I}}, \rho_{\mathfrak{R}})$
if $\sigma_{\mathfrak{R}} = \bot$ **then**
  return $\bot$
$\mathcal{T}_{\mathcal{B}}[m_{\mathfrak{I}}] \leftarrow \mathcal{T}_{\mathcal{B}}[m_{\mathfrak{I}}] \cup \{(m_{\mathfrak{R}}, \sigma_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{R}})\}$
return $\sigma_{\mathfrak{R}}$

$O'_{\mathsf{Sign}}(m)$

$\sigma \leftarrow O_{\mathsf{Sign}}(m)$
$\mathcal{S}_{\mathcal{B}} \leftarrow \mathcal{S}_{\mathcal{B}} \cup \{m\}$
return σ

Figure 21: Reduction of initiator fairness to existential unforgeability of AS_1

we can parse $\mathsf{session}[i] = (\mathsf{st}_{\mathfrak{I}}, m^*)$, $\mathsf{st}_{\mathfrak{I}} = (\tau, \mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}})$ and $\tau = (Y, y) \in \mathsf{R}$. It is straightforward to see that, if AS_2 satisfies presignature adaptability (Definition 3.3), then IFinal, on $\mathsf{st}_{\mathfrak{I}}$ containing $\tau = (Y, y) \in \mathsf{R}$, either outputs a verifiable signature $\sigma_{\mathfrak{R}}$ or $\bot$. Now, since $\mathsf{IFinal}(\mathsf{st}_{\mathfrak{I}}, \rho_{\mathfrak{R}})$ did not fail (see Figure 9), we know that $\sigma_{\mathfrak{R}}$ returned to $\mathcal{A}$ is verifiable, namely, that $\mathsf{Verify}_2(\mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}}, \sigma_{\mathfrak{R}}) = 1$, contradicting the assumption that $\mathcal{A}$ has won. On the other hand, it is easy to see that, if $\mathcal{A}$ fails, so does $\mathcal{B}$.

Write $n = \mathsf{poly}(\lambda)$ for the upper bound on the number of *unique messages* $\mathcal{A}$ queries its $O^1_{\mathfrak{I}}$ oracle about (this number is between 1 and the total number of queries $\mathcal{A}$ makes). Assume without loss of generality that $\mathcal{A}$ (1) never returns $\bot$, (2) opens sessions for (*i.e.*, queries $O^1_{\mathfrak{I}}$ about) exactly n distinct messages, including m^* it eventually returns, and (3) closes (via a corresponding call to $O^2_{\mathfrak{I}}$) every session *not* corresponding to m^*. To see that this is without loss of generality, observe that we could "wrap" $\mathcal{A}$ in another algorithm $\mathcal{A}'$ with polynomial additive overhead in running time and the same success probability α that would (1) sample random (m^*, σ^*) if $\mathcal{A}$ returned $\bot$ or a losing pair $(\hat{m}^*, \hat{\sigma}^*)$ as well as (2) make the missing queries and close any sessions left open by $\mathcal{A}$. As for (3), we have, as per the discussion above, that closing a session corresponding to m^* makes $\mathcal{A}$ lose in $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$, and if $\mathcal{A}$ loses, $\mathcal{A}'$ just samples random m^* for which it opens a session but never closes it. Also, if $\mathcal{A}$ wins by "plucking m^* out of thin air" without ever querying $O^1_{\mathfrak{I}}$ about it, $\mathcal{A}'$ can make one extra query to $O^1_{\mathfrak{I}}$ about m^* and still win. These assumptions allow us to identify the events "no sessions for some message were closed" and "that message was returned as m^*." The former will correspond to the view of $\mathcal{A}$ when run as a subroutine of our reduction being identically distributed to $\mathcal{A}$'s view in $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$, whereas the latter will correspond to $\mathcal{A}$'s being useful for the reduction.

The reduction $\mathcal{B}$ in Figure 21 works by choosing at random an index j between 1 and n and, for every query about the jth message, sampling the statement Y via O_{R}. If we guess correctly, and $\mathcal{A}$ wins by returning the jth message (encoded as $\mathcal{M}[m^*] = j$ in Figure 21), then, by the discussion above, it cannot have queried $O'^2_{\mathfrak{I}}$ about

a session involving m^* as the initiator message, and so $\mathcal{A}$'s view when run as $\mathcal{B}$'s subroutine is identically distributed to the view it has in $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$. We now make this argument precise.

Write Won to mean the event that $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}(\lambda) = 1$ and Guessed to mean the event that $\mathcal{A}$ in the $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$ game did not successfully query $O^2_{\mathfrak{J}}$ about a session involving the jth message that $O^1_{\mathfrak{J}}$ was queried about (for j uniform in $[n]$), where we mean by "successfully" that the call returned something other than $\perp$ and so a record $(m_{\mathfrak{R}}, \rho_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{R}})$ was added to $\mathcal{T}[m_{\mathfrak{I}}]$. Given our assumption about $\mathcal{A}$'s closing all "irrelevant" sessions (*i.e.*, making the corresponding call to $O^2_{\mathfrak{J}}$ for each call to $O^1_{\mathfrak{J}}$ that did not pertain to m^*), we have that the events Won and Guessed are independent. We then have

$$\Pr[\textsc{Won} \wedge \textsc{Guessed}] = \Pr[\textsc{Won}] \cdot \Pr[\textsc{Guessed}] = \frac{\alpha}{n},$$

where we used our assumption that $\mathcal{A}$ returns a message m^* that corresponds to one of the n distinct messages that $O^1_{\mathfrak{J}}$ was queried about. Write Won′ to mean the event that $\mathcal{A}$, when run as a subroutine of $\mathcal{B}$, returned (m^*, σ^*) such that

$$(m^* \notin \mathcal{S}_{\mathcal{B}}) \wedge \left(\mathsf{Verify}_1(\mathsf{pk}, m^*, \sigma^*) = 1\right) \wedge (\mathcal{T}_{\mathcal{B}}[m^*] = \emptyset)$$

and Guessed′ to mean the event that it did not query $O'^2_{\mathfrak{J}}$ about a session i_s involving the jth message that $O'^1_{\mathfrak{J}}$ was queried about and about a presignature $\rho_{\mathfrak{R}}$ such that $\mathsf{PreVerify}_2(\mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}}, \rho_{\mathfrak{R}}, Y) = 1$, for $\mathsf{pk}_{\mathfrak{R}}$, $m_{\mathfrak{R}}$, and Y obtained from $\mathsf{st}_{\mathfrak{I}}$ in $\mathsf{session}[i_s]$ (*i.e.*, any calls to $O'^2_{\mathfrak{J}}$ pertaining to the jth message would have to return $\perp$ even if the reduction knew the witness y). By our assumption that $\mathcal{A}$ makes queries to $O'^1_{\mathfrak{J}}$ about all the n messages it is allowed and finalizes (via calls to $O'^2_{\mathfrak{J}}$) all sessions that do not pertain to m^* it eventually returns, we have that Guessed′ occurs if and only if $\mathcal{M}[m^*] = j$. Finally, until $\mathcal{A}$ successfully (*i.e.*, with a preverifiable presignature $\rho_{\mathfrak{R}}$) queries $O'^2_{\mathfrak{J}}$ about a session involving the jth message, its view as a subroutine of $\mathcal{B}$ and its view in $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$ are distributed identically. Therefore,

$$\begin{aligned}\Pr[\mathsf{aSigForge}_{\mathcal{B},\mathsf{AS}_1}(\lambda) = 1] &= \Pr[\textsc{Won}' \wedge (\mathcal{M}[m^*] = j)] \\ &= \Pr[\textsc{Won}' \wedge \textsc{Guessed}'] \\ &= \Pr[\textsc{Won} \wedge \textsc{Guessed}] \\ &= \frac{\alpha}{n}.\end{aligned}$$

Since we assumed AS_1 is existentially unforgeable, it follows that

$$\alpha \leq n \cdot \mu(\lambda),$$

for a negligible function μ and n polynomial in λ, and so α itself is negligible.

□

Proof of Theorem 4.6. Follows directly from Lemmata E.1 to E.3. □

E.2 Proof of Theorem 6.3

Proof of Theorem 6.3. Correctness is immediate. Let $\mathcal{A}$ be an adversary breaking existential unforgeability of EAS (Figure 15) such that

$$\Pr[\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}(\lambda) = 1] = \alpha.$$

We construct, in Figure 22, a reduction $\mathcal{B}$ that breaks existential unforgeability of AS, *i.e.*, wins the $\mathsf{aSigForge}_{\mathcal{B},\mathsf{AS}}$ experiment (Figure 2), with probability smaller than α by at most a polynomial factor. Since we assume AS is existentially unforgeable, it follows that α is negligible, and so EAS is existentially unforgeable as well. We give $\mathcal{A}$ access to $\mathcal{B}$'s signing oracle O_{Sign} directly, and the other two oracles, O'_{PreSign} and O'_{Adapt}, we simulate as follows. For a query to O'_{PreSign} on a message m, we (most of the time) sample an instance $(F(y), y) \in \mathsf{R}$ and query $\mathcal{B}$'s own presigning oracle O_{PreSign} on m and $F(y)$. We also record the session so that it can be finalized with a later call to O'_{Adapt}. Note that we never explicitly evaluate F in the reduction and sample statements via $\mathsf{Gen}_{\mathsf{R}}$ instead. Requiring that F be one-way is then implicit in saying that R is hard (or that AS is existentially unforgeable).

In order to use $\mathcal{A}$ to try and break aEUF-CMA security of AS, we randomly choose one of $\mathcal{A}$'s queries to O'_{PreSign} and, instead of sampling the witness y ourselves and computing the corresponding statement $F(y)$, we get the statement from $\mathcal{B}$'s relation oracle O_{R}. Roughly speaking, if we guess correctly, and $\mathcal{A}$ wins by returning the message corresponding to this query, then $\mathcal{B}$ will win as well. We now make this argument formal by, among other things, accounting for $\mathcal{A}$'s making multiple queries to O'_{PreSign} about the same message.

Write $n = \mathsf{poly}(\lambda)$ for the upper bound on the number of *unique messages* $\mathcal{A}$ queries its O_{PreSign} oracle about (compare with the proof of Lemma E.3). Assume without loss of generality that $\mathcal{A}$ (1) never returns $\perp$, (2) calls O_{PreSign} for exactly n distinct messages, including m^* it eventually returns, and (3) calls O_{Adapt} for every session *not* corresponding to m^*. The argument that this is indeed without loss of generality is analogous to the one in the proof of Lemma E.3.

The reduction $\mathcal{B}$ in Figure 22 works analogously to the reduction from the proof of Lemma E.3. It samples a uniform index $j \in [n]$ and, for every query about the jth message, samples the statement Y via a call to O_{R}. Observe that $\mathcal{B}$ wins exactly when $\mathcal{A}$ wins in the $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$ game and our guess of j was correct. To see this, we keep track in Figure 22 of the sets $\mathcal{S}_{\mathcal{B}}$, $\mathcal{T}_{\mathcal{B}}$, and $C_{\mathcal{B}}$ corresponding to the sets $\mathcal{S}$, $\mathcal{T}$, and C from Figure 2, respectively. We now make this argument precise.

Write Won to mean the event $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}(\lambda) = 1$ and Guessed to mean the event that $\mathcal{A}$ in the $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$ game did not query O_{Adapt} about a session involving the jth message that O_{PreSign} was queried about (for j uniform in $[n]$). Given our assumption about $\mathcal{A}$'s closing all "irrelevant" sessions (*i.e.*, making the corresponding call to O_{Adapt} for each call to O_{PreSign} that did not pertain to m^*), we have that the events Won and Guessed are independent, and

$$\Pr[\textsc{Won} \wedge \textsc{Guessed}] = \Pr[\textsc{Won}] \cdot \Pr[\textsc{Guessed}] = \frac{\alpha}{n},$$

where we used our assumption that $\mathcal{A}$ returns a message m^* that corresponds to one of the n distinct messages that O_{PreSign} was queried about. Write Won′ to mean the event that $\mathcal{A}$, when run as a subroutine of $\mathcal{B}$, returned (m^*, σ^*) such that $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*)$ and $m^* \notin Q_{\mathcal{A}}$ and Guessed′ to mean the event that it did not query O'_{Adapt} about a session involving the jth message that O'_{PreSign} was queried about. By our assumption about $\mathcal{A}$, Guessed′ happens

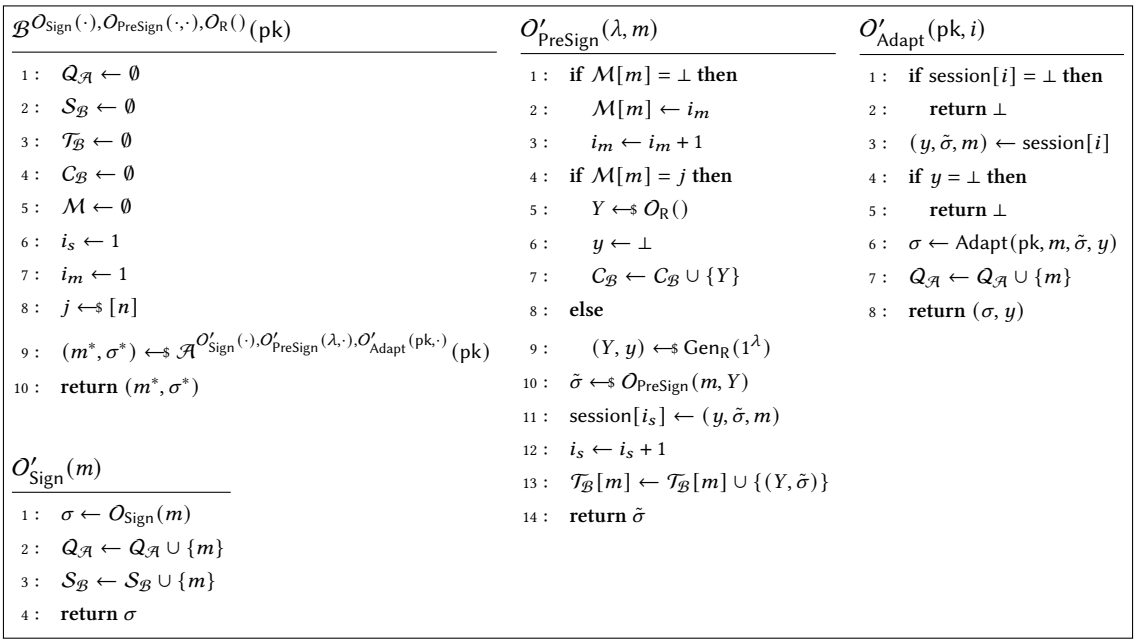


Figure 22: Reduction of aEUF-CMA **security of the compiled** EAS **to** aEUF-CMA **security of the original** AS

if and only if $\mathcal{M}[m^*] = j$, *i.e.*, the only query to O'_{PreSign} that is not "paired with" a query to O'_{Adapt} is the one for the message m^* that $\mathcal{A}$ returns in the end. Finally, until $\mathcal{A}$ queries O'_{Adapt} about a session involving the jth message, its view as a subroutine of $\mathcal{B}$ is distributed identically to its view in $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$. Therefore,

$$\begin{aligned}\Pr[\mathsf{aSigForge}_{\mathcal{B},\mathsf{AS}}(\lambda) = 1] &= \Pr[\textsc{Won}' \wedge (\mathcal{M}[m^*] = j)] \\ &= \Pr[\textsc{Won}' \wedge \textsc{Guessed}'] \\ &= \Pr[\textsc{Won} \wedge \textsc{Guessed}] \\ &= \frac{\alpha}{n}\,.\end{aligned}$$

Since we assumed AS is existentially unforgeable,

$$\alpha \leq n \cdot \mu(\lambda)\,,$$

for a negligible function μ. Since $n = \mathsf{poly}(\lambda)$, it follows that α is negligible. □

E.3 Proof of Theorem 6.4

Lemma E.4. *The construction in Figure 16 is a correct signature exchange protocol.*

Proof. The proof is exactly analogous to that of Lemma E.1 and relies on presignature correctness of both EAS_1 and AS_2 and on presignature adaptability of EAS_1. □

Lemma E.5. *The construction in Figure 16 satisfies responder fairness.*

Proof. The proof is identical to that of Lemma E.2 and relies on witness extractability of AS_2 and presignature adaptability (defined exactly the same way for extended adaptor signature schemes and regular adaptor signature schemes) of EAS_1. □

Lemma E.6. *The construction in Figure 16 satisfies initiator fairness.*

Proof. We reduce initiator fairness to existential unforgeability (aEUF-CMA) of EAS_1 while also relying on presignature adaptability of AS_2. Let $\mathcal{A}$ be an adversary breaking initiator fairness of the construction in Figure 16, denoted SE, such that

$$\Pr[\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}(\lambda) = 1] = \alpha\,.$$

We construct, in Figure 23, a reduction $\mathcal{B}$ that wins the $\mathsf{aSigForge}_{\mathcal{B},\mathsf{EAS}_1}$ experiment (Figure 15) with probability α. Since we assume EAS_1 is unforgeable, it follows that α is negligible, and so the construction in Figure 16 is initiator-fair. We simulate for $\mathcal{A}$ the two initiator oracles $O'^1_{\mathfrak{I}}$ and $O'^2_{\mathfrak{I}}$, as well as the signing oracle O'_{Sign}, using calls to $\mathcal{B}$'s own oracles. We have the reduction maintain the query sets $\mathcal{T}_{\mathcal{B}}$ and $\mathcal{S}_{\mathcal{B}}$ to help the reader keep track of the properties of (m^*, σ^*) returned by $\mathcal{A}$. Observe that the set $\mathcal{T}_{\mathcal{B}}$ tracks FSE sessions *closed* by the (malicious) responder.

Note that the simulation is perfect. Indeed, $O'^1_{\mathfrak{I}}$ exactly simulates IPrep using the call to $\mathcal{B}$'s own presigning oracle O_{PreSign} which honestly samples a relation instance $(Y, y) \in \mathsf{R}$ and presigns relative to Y. Similarly, $O'^2_{\mathfrak{I}}$ perfectly simulates IFinal: since the session indices i in the reduction and in the $\mathsf{aSigForge}_{\mathcal{B},\mathsf{EAS}_1}$ experiment (Figure 15) match exactly, the adaptation oracle O_{Adapt} calls Adapt_1 on $m_{\mathfrak{I}}$, $\tilde{\sigma}_{\mathfrak{I}}$, and y corresponding to the ith session opened, *i.e.*, such that $\tilde{\sigma}_{\mathfrak{I}}$ was produced via $\mathsf{PreSign}_1(\mathsf{sk}, m_{\mathfrak{I}}, y)$. Presignature adaptability (Definition 3.3) of AS_2 then implies that $\sigma_{\mathfrak{R}}$ is a verifiable signature (since $O'^2_{\mathfrak{I}}$ checked that $\rho_{\mathfrak{R}}$ preverifies before adapting it into $\sigma_{\mathfrak{R}}$ using a valid witness y leaked by O_{Adapt}).

Since the simulation is perfect, we have that, with probability α, $\mathcal{A}$ (and, by extension, $\mathcal{B}$) returns (m^*, σ^*) such that, at the same

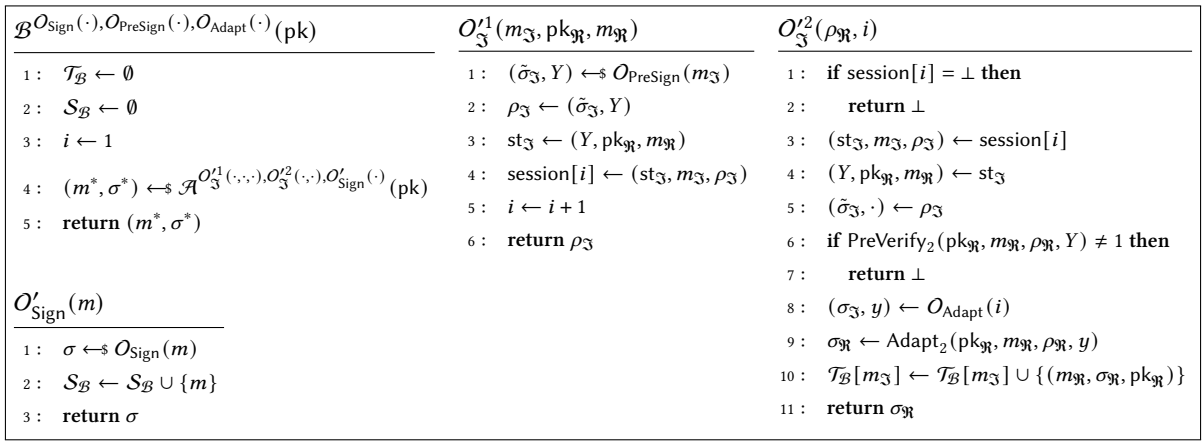


Figure 23: Reduction of initiator fairness to existential unforgeability of EAS_1

time $\mathsf{Verify}_1(\mathsf{pk}, m^*, \sigma^*)$, $m^* \notin \mathcal{S}_{\mathcal{B}}$ and

$$\big(\forall (m_{\mathfrak{R}}, \sigma_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{R}}) \in \mathcal{T}_{\mathcal{B}}[m^*]\big)\big(\mathsf{Verify}_2(\mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}}, \sigma_{\mathfrak{R}}) \neq 1\big)\,.$$

The first clause corresponds exactly to the first clause of $\mathcal{B}$'s victory condition. It suffices to show how the remaining two clauses above relate to the final clause in the victory condition of $\mathcal{B}$, namely, that $m^* \notin Q$, where Q is the set of messages a signature on which was produced by O_{Sign} *or* a presignature on which was adapted by O_{Adapt}. Note that $\mathcal{S}_{\mathcal{B}} \subseteq Q$ is the set of messages added to Q by virtue of querying the signing oracle O_{Sign} about them. Assuming $\mathcal{A}$ wins, $m^* \notin \mathcal{S}_{\mathcal{B}}$, and so we only have to analyze if m^* has been added to Q in O_{Adapt}, which is only called when a session is finalized involving m^*. Consider again two cases: either (1) $\mathcal{T}_{\mathcal{B}}[m^*] = \emptyset$, or (2) $\mathcal{T}_{\mathcal{B}}[m^*]$ is not empty, and, for all entries $(m_{\mathfrak{R}}, \sigma_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{R}})$ in it, $\mathsf{Verify}_2(\mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}}, \sigma_{\mathfrak{R}})$ fails. The latter case is impossible. Indeed, presignature adaptability of AS_2 implies that, whenever $(m_{\mathfrak{R}}, \sigma_{\mathfrak{R}}, \mathsf{pk}_{\mathfrak{R}})$ gets added to $\mathcal{T}_{\mathcal{B}}[m^*]$, it must satisfy $\mathsf{Verify}_2(\mathsf{pk}_{\mathfrak{R}}, m_{\mathfrak{R}}, \sigma_{\mathfrak{R}})$. Assume therefore that $\mathcal{T}_{\mathcal{B}}[m^*] = \emptyset$. Then, no session i involving m^* has been finalized, but, since session indices maintained by the reduction $\mathcal{B}$ and those maintained by $\mathcal{B}$'s oracles O_{PreSign} and O_{Adapt} match exactly, we have that $m^* \notin Q$.

The reduction $\mathcal{B}$ hence wins exactly when $\mathcal{A}$ wins, and $\mathcal{A}$'s views when run as $\mathcal{B}$'s subroutine and in the $\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}$ experiment are distributed identically, yielding,

$$\Pr[\mathsf{aSigForge}_{\mathcal{B},\mathsf{EAS}_1}(\lambda) = 1] = \Pr[\mathsf{FairnessInit}_{\mathcal{A},\mathsf{SE}}(\lambda) = 1] = \alpha\,.$$

Since EAS_1 is assumed existentially unforgeable, we have that α is negligible. □

Proof of Theorem 6.4. Follows directly from Lemmata E.4 to E.6. □

E.4 Proof of Theorem 6.5

Proof of Theorem 6.5. The proof is analogous to that of Lemma E.3. Let $\mathcal{A}$ be an efficient adversary such that

$$\Pr[\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}(\lambda) = 1] = \alpha\,.$$

The proof follows by constructing, in Figure 24, a polynomial-time reduction $\mathcal{B}$ that, given oracle access to $\mathcal{A}$, wins the EUF-CMA experiment for the underlying signature scheme Σ with probability smaller than α by at most a polynomial factor.

The reduction uses its signing oracle O_{Sign} to simulate the signing, presigning, and adapting oracles for $\mathcal{A}$. The signing oracle O'_{Sign} is simulated by simply forwarding the call to $\mathcal{B}$'s own signing oracle. The presigning oracle O'_{PreSign} is more interesting, however. It maintains a "reverse mapping" table $\mathcal{M}$ that maps each unique message m the oracle gets queried on to a unique index i_m. The reduction $\mathcal{B}$ selects one such index, call it j, ahead of time and, for queries about messages other than the jth message, produces the presignatures by "unadapting" signatures obtained from O_{Sign}.

The difficult part in leveraging $\mathcal{A}$ to win the EUF-CMA experiment is that $\mathcal{A}$ is allowed to query O_{PreSign} on the message m^* it eventually returns, so $\mathcal{B}$ cannot respond to all queries this way. This is exactly why it singles out the jth message. The adaptor signature adversary $\mathcal{A}$, when victorious in the $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$ game, will not have asked for the adaptation of any presignatures it obtained for m^* (since any such call to O_{Adapt} would add m^* to Q in Figure 15). The reduction $\mathcal{B}$ therefore will not need the witness y for any of these presignatures and can instead simulate them using the simulator Sim. And so it guesses one of the messages that $\mathcal{A}$ will query O'_{PreSign} about to be the message m^* that $\mathcal{A}$ will eventually return and, for that message, responds by simulating the presignature $\tilde{\sigma}$ via Sim. Since $\mathcal{A}$ can make at most polynomially-many queries, $\mathcal{B}$ will guess correctly with non-negligible probability and, when it does, will win the EUF-CMA experiment for the underlying signature scheme Σ.

Write $n = \mathsf{poly}(\lambda)$ for the upper bound on the number of *unique messages* $\mathcal{A}$ queries its O_{PreSign} oracle about (this number is between 1 and the total number of queries $\mathcal{A}$ makes). Assume without loss of generality that $\mathcal{A}$ behaves as follows: it (1) always returns a message-signature pair (m^*, σ^*), (2) queries O_{PreSign} about exactly n distinct messages, including m^* it eventually returns, and (3) calls O_{Adapt} on all sessions i *not* involving m^*. To see this is without loss of generality, observe that we could "wrap" any $\mathcal{A}$ in another algorithm $\mathcal{A}'$ with polynomial additive overhead in running time and the same success probability α that would sample random (m^*, σ^*)

$\mathcal{B}^{O_{\mathsf{Sign}}(\cdot)}(\mathsf{pk})$

$i_s \leftarrow 1$
$i_m \leftarrow 1$
$j \leftarrow\!\$\ [n]$
$\mathcal{M} \leftarrow \emptyset$
$\mathsf{session} \leftarrow \emptyset$
$(m^*, \sigma^*) \leftarrow\!\$\ \mathcal{A}^{O'_{\mathsf{Sign}}(\cdot), O'_{\mathsf{PreSign}}(\lambda, \mathsf{pk}, \cdot), O'_{\mathsf{Adapt}}(\mathsf{pk}, \cdot)}(\mathsf{pk})$
return (m^*, σ^*)

$O'_{\mathsf{Sign}}(m)$

$\sigma \leftarrow\!\$\ O_{\mathsf{Sign}}(m)$
return σ

$O'_{\mathsf{Adapt}}(\mathsf{pk}, i)$

if $\mathsf{session}[i] = \bot$ **then**
  return $(\bot, \bot)$
$(m, \sigma, y) \leftarrow \mathsf{session}[i]$
return (σ, y)

$O'_{\mathsf{PreSign}}(\lambda, \mathsf{pk}, m)$

if $\mathcal{M}[m] = \bot$ **then**
  $\mathcal{M}[m] \leftarrow i_m$
  $i_m \leftarrow i_m + 1$
if $\mathcal{M}[m] = j$
  $(\tilde{\sigma}, Y) \leftarrow\!\$\ \mathsf{Sim}(\mathsf{pk}, m)$
  $(\sigma, y) \leftarrow (\bot, \bot)$
else
  $\sigma \leftarrow\!\$\ O_{\mathsf{Sign}}(m)$
  $(Y, y) \leftarrow\!\$\ \mathsf{Gen}_{\mathsf{R}}(1^\lambda)$
  $\tilde{\sigma} \leftarrow\!\$\ \mathsf{Unadapt}(\mathsf{pk}, m, \sigma, Y, y)$
$\mathsf{session}[i_s] \leftarrow (m, \sigma, y)$
$i_s \leftarrow i_s + 1$
return $(\tilde{\sigma}, Y)$

Figure 24: Reduction of aEUF-CMA **security of** EAS **with deniable presignatures and an unadapting algorithm to** EUF-CMA **security of the underlying signature scheme** Σ

whenever $\mathcal{A}$ returned either $\bot$ or a losing pair $(\hat{m}^*, \hat{\sigma}^*)$ as well as make any missing queries. These assumptions then allow us to identify the events "no sessions were closed via O_{Adapt} for some message" and "that message was returned as m^*." The former will correspond to the distribution of $\mathcal{A}$'s view when run as a subroutine of our reduction being indistinguishable (under the presignature deniability assumption) to $\mathcal{A}$'s view in $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$. The latter, on the other hand, will correspond to $\mathcal{A}$'s being useful for the reduction.

Let the reduction $\mathcal{B}$ choose the index j at which to "embed" the challenge presignature $\tilde{\sigma}$ obtained from Sim uniformly from $[n]$. To bridge the gap between $\mathcal{A}$'s being run as a subroutine of $\mathcal{B}$ and $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$, we construct, in Figure 25, a distinguisher $\mathcal{D}$ that uses $\mathcal{A}$ to tell simulated presignatures from genuine ones. When $\mathcal{D}$ is playing the $\mathsf{pSim}_{\mathcal{D},\mathsf{Sim},\mathsf{EAS}}$ game, it is running exactly $\mathcal{B}$'s simulation for $\mathcal{A}$ and outputs 1 only if $\mathcal{B}$ also wins in its $\mathsf{SigForge}_{\mathcal{B},\Sigma}$ game, *i.e.*, we have:

$$\Pr[\mathsf{pSim}_{\mathcal{D},\mathsf{Sim},\mathsf{EAS}}(\lambda) = 1] \leq \Pr[\mathsf{SigForge}_{\mathcal{B},\Sigma}(\lambda) = 1]\,.$$

On the other hand, when $\mathcal{D}$ is playing the $\mathsf{pReal}_{\mathcal{D},\mathsf{EAS}}$ game, it is simulating for $\mathcal{A}$ exactly the $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$ experiment except when $\mathcal{A}$ queries O'_{Adapt} about the jth message (in which case it receives $(\bot, \bot)$ when run as a subroutine of $\mathcal{D}$ but a legitimate $(\tilde{\sigma}, y)$ pair in $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$). This is fine, since we only ever care about the cases when $\mathcal{A}$ does not query O'_{Adapt} about a session involving the jth message. We now make the argument precise.

Write Won to mean the event that $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}(\lambda) = 1$ and Guessed to mean the event that $\mathcal{A}$ in the $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$ game did not query O_{Adapt} about a session involving the jth message that O_{PreSign} was queried about (for j uniform in $[n]$). Given our assumption about $\mathcal{A}$'s closing all "irrelevant" sessions (*i.e.*, making the corresponding call to O_{Adapt} for each call to O_{PreSign} that did not pertain to m^*), we have that the events Won and Guessed are independent. Therefore,

$$\Pr[\textsc{Won} \wedge \textsc{Guessed}] = \Pr[\textsc{Won}] \cdot \Pr[\textsc{Guessed}] = \frac{\alpha}{n}\,,$$

where we used our assumption that $\mathcal{A}$ returns a message m^* that corresponds to one of the n distinct messages that O_{PreSign} was queried about. Write Won′ to mean the event that $\mathcal{A}$ returns, in $\mathsf{pReal}_{\mathcal{D},\mathsf{EAS}}$, a pair (m^*, σ^*) such that $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*)$ and $m^* \notin Q$ and Guessed′ to mean the event that it did not query O'_{Adapt} about a session involving the jth message that O'_{PreSign} was queried about. By our assumption that $\mathcal{A}$ makes queries to O'_{PreSign} about all the n messages it is allowed and finalizes (via calls to O'_{Adapt}) all sessions that do not pertain to m^* it eventually returns, we have that Guessed′ occurs if and only if $\mathcal{M}[m^*] = j$, which $\mathcal{D}$ checks explicitly. Finally, until $\mathcal{A}$ queries O'_{Adapt} about a session involving the jth message, its view in $\mathsf{aSigForge}_{\mathcal{A},\mathsf{EAS}}$ is distributed identically to its view when run as a subroutine of $\mathcal{D}$ in $\mathsf{pReal}_{\mathcal{D},\mathsf{EAS}}$ (since Unadapt produces presignatures distributed identically to the output of PreSign for fresh instances $(Y, y) \leftarrow\!\$\ \mathsf{Gen}_{\mathsf{R}}(1^\lambda)$ sampled independently each time). Therefore,

$$\begin{aligned}\Pr[\mathsf{pReal}_{\mathcal{D},\mathsf{EAS}}(\lambda) = 1] &= \Pr[\textsc{Won}' \wedge (\mathcal{M}[m^*] = j)] \\ &= \Pr[\textsc{Won}' \wedge \textsc{Guessed}'] \\ &= \Pr[\textsc{Won} \wedge \textsc{Guessed}] \\ &= \Pr[\textsc{Won}] \cdot \Pr[\textsc{Guessed}] \\ &= \frac{\alpha}{n}\,.\end{aligned}$$

By the assumption that EAS has deniable presignatures, there exists a negligible function ε such that

$$\Pr[\mathsf{pReal}_{\mathcal{D},\mathsf{EAS}}(\lambda) = 1] \leq \Pr[\mathsf{pSim}_{\mathcal{D},\mathsf{Sim},\mathsf{EAS}}(\lambda) = 1] + \varepsilon\,.$$

It follows that

$$\alpha \leq n \cdot \Pr[\mathsf{SigForge}_{\mathcal{B},\Sigma}(\lambda) = 1] + n \cdot \varepsilon\,.$$

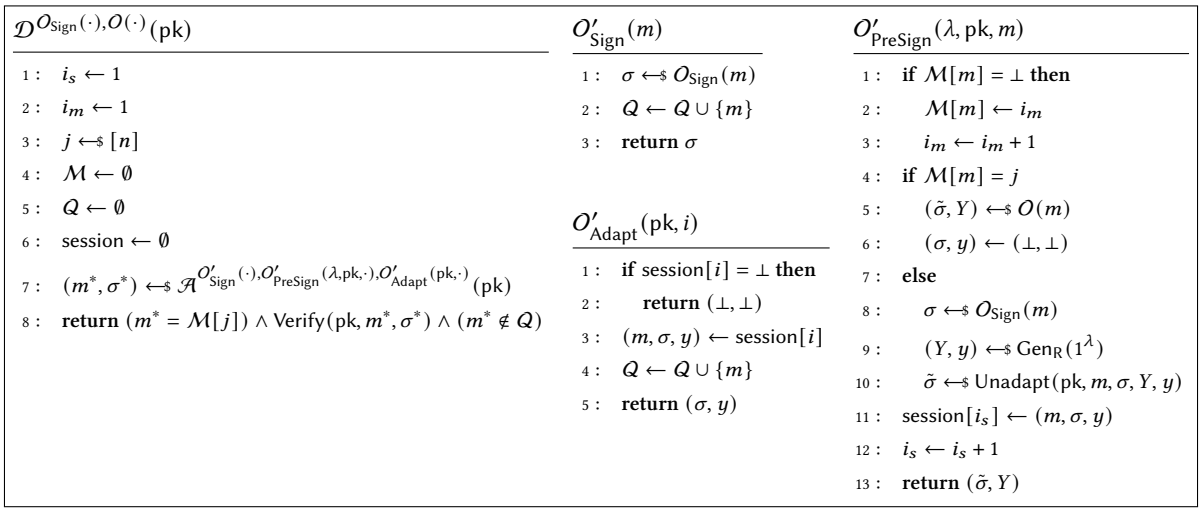


Figure 25: Distinguisher of $\mathsf{pReal}_{\mathcal{D},\mathsf{EAS}}$ from $\mathsf{pSim}_{\mathcal{D},\mathsf{Sim},\mathsf{EAS}}$ built from an aEUF-CMA adversary $\mathcal{A}$ for EAS

Since we assumed Σ is existentially unforgeable, it follows that

$$\alpha \leq n \cdot \mu(\lambda) + n \cdot \varepsilon(\lambda)\,,$$

for negligible functions μ and ε, and n polynomial in λ, and so α itself is negligible. □

Received 21 August 2026; revised TBD; accepted TBD