Generic Double-Authentication Preventing Signatures and a Post-Quantum Instantiation[⋆]

David Derler[1], Sebastian Ramacher[1], and Daniel Slamanig[2]

[1] IAIK, Graz University of Technology, Austria
[2] AIT Austrian Institute of Technology GmbH, Vienna, Austria
cryptsec@derler.info, sebastian.ramacher@tugraz.at,
daniel.slamanig@ait.ac.at

Abstract. Double-authentication preventing signatures (DAPS) are a variant of digital signatures which have received considerable attention recently (Derler et al. EuroS&P 2018, Poettering Africacrypt 2018). They are unforgeable signatures in the usual sense and sign messages that are composed of an address and a payload. Their distinguishing feature is the property that signatures on *two different* payloads with respect to the *same* address allow to publicly extract the secret signing key. Thus, they are a means to disincentivize double-signing and are a useful tool in various applications.

DAPS are known in the factoring, the discrete logarithm and the lattice setting. The majority of the constructions are ad-hoc. Only recently, Derler et al. (EuroS&P 2018) presented the first generic construction that allows to extend *any* discrete logarithm based secure signature scheme to DAPS. However, their scheme has the drawback that the number of potential addresses (the address space) used for signing is polynomially bounded (and in fact small) as the size of secret and public keys of the resulting DAPS are linear in the address space. In this paper we overcome this limitation and present a generic construction of DAPS with constant size keys and signatures. Our techniques are not tailored to a specific algebraic setting and in particular allow us to construct the first DAPS without structured hardness assumptions, i.e., from symmetric key primitives, yielding a candidate for post-quantum secure DAPS.

Keywords: digital signatures, double-authentication prevention, Shamir secret sharing, provable-security, generic construction, exponential size address space

1 Introduction

Digital signatures are an important cryptographic primitive used to provide strong integrity and authenticity guarantees for digital messages. Among many other applications, they are used to issue digital certificates for public keys within public-key infrastructures, to guarantee the origin of executable code, to sign

[⋆] All authors have been supported by EU H2020 project Prismacloud, grant agreement n°644962.

digital documents such as PDF documents (in a legally binding way), as well as in major cryptographic protocols such as TLS. Recently, signatures also emerged to be a cornerstone of distributed cryptocurrencies such as Bitcoin, i.e., are used to bind coins to users (by means of public keys) and to sign transactions.

Double-authentication preventing signatures (DAPS) are a variant of digital signatures used to sign messages of the form $m = (a, p)$ with a being the so called address and p the payload. They provide unforgeability guarantees in the sense of conventional signatures but have the special property that signing two different payloads $p \neq p'$ using the same address a allows to publicly extract the secret signing key from the respective signatures. In the literature, various compelling applications for DAPS have been proposed. Those applications include penalizing double spending attacks in cryptocurrencies [RKS15] or penalizing certification authorities for issuing two certificates with respect to the same domain name, but for two different public keys [PS14], for example. In this work we purely focus on DAPS constructions and we refer the reader to [PS14,PS17] for a comparison with other types of self-enforcing digital signatures.

Currently, DAPS are known in the factoring [PS14,PS17,BPS17], the discrete logarithm [RKS15,DRS18b,Poe18] and the lattice setting [BKN17]. The majority of the constructions (the only exception being [DRS18b]) are ad-hoc. Unfortunately, such an approach yields very specific constructions, whose security may not be well understood. Having generic DAPS constructions, in contrast, yields much more flexibility, as it allows to plug in building blocks whose security is well understood. In addition, this yields simplicity and modularity in the security analysis. Only recently, Derler et al. (EuroS&P 2018) presented the first generic construction that allows to extend *any* discrete logarithm based EUF-CMA secure signatures scheme to DAPS. However, their scheme has the drawback that the number of potential addresses (the address space) used for signing is polynomially bounded (and in fact small) as the size of secret and the public keys of the resulting DAPS are linear in the address space. We ask whether we can come up with a generic construction without this drawback.

Somewhat orthogonal to the motivational discussion above, our work is also driven by the question whether it is possible to construct DAPS without relying on structured hardness assumptions, i.e., solely from symmetric key primitives (following up on a very recent line of work [CDG$^+$17a,DRS18a,BEF18,KKW18]). This is interesting, because symmetric key primitives are conjectured to remain secure in the advent of sufficiently powerful quantum computers. Such quantum computers would break all discrete log and RSA based public key cryptosystems [Sho97].

1.1 Existing DAPS Constructions

DAPS have been introduced by Poettering and Stebila [PS14,PS17] in a factoring-based setting. Ruffing, Kate and Schröder later introduced the notion of accountable assertions (AS) in [RKS15], being a related but weaker primitive than DAPS. In addition they present one AS that also is a DAPS (RKS henceforth). The RKS construction is based on Merkle tress and chameleon hash functions in the discrete logarithm setting. Very recently, Bellare, Poettering and Ste-

Approach	Address space	Extraction	Setting	Generic
[PS14,PS17]	exponential	DSE	factoring	×
[RKS15]	exponential	DSE	DLOG	×
[BPS17]	exponential	DSE	factoring	×
[BKN17]	exponential	DSE	lattices	×
[DRS18b]	small	wDSE*	DLOG	✓
[Poe18]	small	DSE	DLOG	×
Construction 1	exponential	wDSE	symmetric	✓
Construction 2	exponential	DSE	any	✓

Table 1: Overview of DAPS constructions

bila [BPS17] proposed new factoring-based DAPS from trapdoor identification-schemes using an adaption and extension of a transform from [BPS16]. Their two transforms applied to the Guillou-Quisquater (GQ) [GQ88] and Micali-Reyzin (MR) [MR02] identification scheme yield signing and verification times as well as signature sizes comparable (or slightly above) standard RSA signatures. Boneh et al. [BKN17] propose constructions of DAPS from lattices. They consider DAPS as a special case of what they call predicate-authentication-preventing signatures (PAPS). In PAPS one considers a k-ary predicate on the message space and given any k valid signatures that satisfy the predicate reveal the signing key. Consequently, DAPS are PAPS for a specific 2-ary predicate. Derler, Ramacher and Slamanig (DRS henceforth) in [DRS18b] recently provided the first black-box construction of DAPS from digital signatures schemes and demonstrate how this approach can be used to construct N-times-authentication-preventing signatures (NAPS) (a notion called k-way DAPS in [BKN17]). In addition, they introduced weaker extraction notions, where the focus of the extraction is on the signing key of the underlying signature scheme only. A drawback of their work is that the constructions have $O(n)$ secret and public key size where n is the size of the address space. So their constructions are only suitable for small message spaces. In a follow up work Poettering [Poe18], also focusing on DAPS for small address spaces, showed how for a certain class of signature schemes (obtained via Fiat-Shamir from certain identification schemes), the DRS approach can be improved by reducing the signature size by a factor of five and the size of the secret key from $O(n)$ to $O(1)$. However, this comes at the cost of no longer being able to do a black-box reduction to the underlying signature scheme. In Table 1 we provide a comparison of existing DAPS approaches with the ones presented in this paper regarding address space, extraction capabilities, algebraic setting as well as their characteristic as either being tailored to a specific setting or generic.

1.2 Contribution

Our contributions can be summarized as follows:

– We propose a generic DAPS, respectively NAPS, construction building upon DRS' secret-sharing approach, which resolves the address-space limitation in the DRS construction, and, in particular, supports an exponentially large address space. This improvement is achieved by deriving the coefficients of the secret sharing polynomial from the address using a carefully chosen pseudorandom function with an output domain being compatible with the secret key space of the underlying signature scheme. Consequently, the overhead in the public-key reduces to a constant factor. Like the DRS approach, our generic approach satisfies a relaxed notion of extractability. Interestingly, we can instantiate this construction solely from symmetric-key primitives, yielding a candidate for post-quantum secure DAPS/NAPS.

– While the aforementioned construction thus closes an important gap in the literature, the signature sizes are somewhat large compared to signatures in the discrete log or RSA setting. To this end, we additionally follow a different direction which basically targets the extension of any digital signature scheme (such as ECDSA or EdDSA, for example) to a DAPS. Essentially, we present a compiler which uses an arbitrary DAPS scheme to extend any given signature scheme to a DAPS. While this might sound somewhat odd at first sight, we want to stress that all existing DAPS which have compact keys and exponentially large address space are ad-hoc constructions, whereas practical applications most likely will use standardized signature schemes. Using our construction it is possible to generically bring extraction to any signature scheme. Hence we obtain more efficient DAPS being compatible with standardized signature schemes such as ECDSA or EdDSA.

2 Preliminaries

In this section we firstly present a formal model for the security of signature and DAPS schemes, recall non-interactive zero-knowledge proof systems and Shamir's secret sharing.

2.1 Digital Signature Schemes

Subsequently we formally recall the notion of digital signature schemes.

Definition 1 (Signature Scheme). *A signature scheme* Σ *is a triple* $(\mathsf{KGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$ *of PPT algorithms, which are defined as follows:*

$\mathsf{KGen}_\Sigma(1^\kappa)$: *This algorithm takes a security parameter* κ *as input and outputs a secret (signing) key* sk_Σ *and a public (verification) key* pk_Σ *with associated message space* $\mathcal{M}$ *(we may omit to make the message space* $\mathcal{M}$ *explicit).*

$\mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)$: *This algorithm takes a secret key* sk_Σ *and a message* $m \in \mathcal{M}$ *as input and outputs a signature* σ*.*

$\mathsf{Verify}_\Sigma(\mathsf{pk}_\Sigma, m, \sigma)$: *This algorithm takes a public key* pk_Σ*, a message* $m \in \mathcal{M}$ *and a signature* σ *as input and outputs a bit* $b \in \{0, 1\}$*.*

We require a signature scheme to be correct and to provide existential unforgeability under adaptively chosen message attacks ($\mathsf{EUF\text{-}CMA}$ security). For correctness we require that for all $\kappa \in \mathbb{N}$, for all $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \mathsf{KGen}_\Sigma(1^\kappa)$ and for all $m \in \mathcal{M}$ it holds that

$$\Pr\left[\mathsf{Verify}_\Sigma(\mathsf{pk}_\Sigma, m, \mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)) = 1\right] = 1.$$

Definition 2 (EUF-CMA). *For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of* EUF-CMA *as*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\kappa) = 1\right]$$

where the corresponding experiment is depicted in Figure 1. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\kappa) \leq \varepsilon(\kappa)$$

we say that Σ is EUF-CMA *secure.*

```
$\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\Sigma}(\kappa)$:
  $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \mathsf{KGen}_\Sigma(1^\kappa)$
  $\mathcal{Q} \leftarrow \emptyset$
  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'_\Sigma(\mathsf{sk}_\Sigma, \cdot)}(\mathsf{pk})$
    where oracle $\mathsf{Sign}'_\Sigma$ on input $m$:
      $\sigma \leftarrow \mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)$, $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
      return $\sigma$
  return 1, if $\mathsf{Verify}_\Sigma(\mathsf{pk}_\Sigma, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{Q}$
  return 0
```

Fig. 1: EUF-CMA **security.**

2.2 Double-Authentication-Preventing Signatures

Double-authentication-preventing signatures (DAPS) are signature schemes being capable of signing messages from a message space $\mathcal{M}$ of the form $\mathsf{A} \times \mathsf{P}$. Each message $m = (a, p) \in \mathcal{M}$ thereby consists of an address a in address space A and a payload p from payload space P. In addition to the algorithms provided by conventional signature schemes, a DAPS scheme provides a fourth algorithm Ex_D that extracts the secret key from signatures on two colliding messages, i.e., two different messages sharing the same address. Formally, a pair of colliding messages is defined as follows:

Definition 3 **(Colliding Messages).** *We call two messages $m_1 = (a_1, p_1)$ and $m_2 = (a_2, p_2)$ colliding if $a_1 = a_2$, but $p_1 \neq p_2$.*

Below, we now formally define DAPS following [PS14,PS17].

Definition 4 **(DAPS).** *A double-authentication-preventing signature scheme* DAPS *is a tuple $(\mathsf{KGen}_\mathsf{D}, \mathsf{Sign}_\mathsf{D}, \mathsf{Verify}_\mathsf{D}, \mathsf{Ex}_\mathsf{D})$ of PPT algorithms, which are defined as follows:*

$\mathsf{KGen}_\mathsf{D}(1^\kappa)$: *This algorithm takes a security parameter κ as input and outputs a secret (signing) key sk_D and a public (verification) key pk_D with associated message space $\mathcal{M}$ (we may omit to make the message space $\mathcal{M}$ explicit).*

$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)$: *This algorithm takes a secret key sk_D and a message $m \in \mathcal{M}$ as input and outputs a signature σ.*

$\mathsf{Verify_D}(\mathsf{pk_D}, m, \sigma)$: *This algorithm takes a public key* $\mathsf{pk_D}$*, a message* $m \in \mathcal{M}$ *and a signature* σ *as input and outputs a bit* $b \in \{0, 1\}$*.*

$\mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$: *This algorithm takes a public key* $\mathsf{pk_D}$*, two colliding messages* m_1 *and* m_2 *and signatures* σ_1 *for* m_1 *and* σ_2 *for* m_2 *as inputs and outputs a secret key* $\mathsf{sk_D}$*.*

Note that the algorithms $\mathsf{KGen_D}$, $\mathsf{Sign_D}$, and $\mathsf{Verify_D}$ match the definition of the algorithms of a conventional signature scheme. For DAPS one requires a restricted but otherwise standard notion of unforgeability [PS14,PS17], where adversaries can adaptively query signatures for messages but only on distinct addresses. Figure 2 details the unforgeability security experiment.

Definition 5 (EUF-CMA [**PS14**])**.** *For a PPT adversary* $\mathcal{A}$*, we define the advantage function in the sense of* EUF-CMA *as*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right]$$

where the corresponding experiment is depicted in Figure 2. If for all PPT adversaries $\mathcal{A}$ *there is a negligible function* $\varepsilon(\cdot)$ *such that*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) \leq \varepsilon(\kappa)$$

we say that DAPS *is* EUF-CMA *secure.*

```
$\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{sk_D}, \mathsf{pk_D}) \leftarrow \mathsf{KGen_D}(1^\kappa)$
  $\mathcal{Q} \leftarrow \emptyset$, $\mathcal{R} \leftarrow \emptyset$
  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}'_\mathsf{D}(\mathsf{sk_D}, \cdot)}(\mathsf{pk}_\Sigma)$
    where oracle $\mathsf{Sign}'_\mathsf{D}$ on input $m$:
      $(a, p) \leftarrow m$
      if $a \in \mathcal{R}$, return $\perp$
      $\sigma \leftarrow \mathsf{Sign_D}(\mathsf{sk_D}, m)$, $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$, $\mathcal{R} \leftarrow \mathcal{R} \cup \{a\}$
      return $\sigma$
  return 1, if $\mathsf{Verify_D}(\mathsf{pk_D}, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{Q}$
  return 0
```

Fig. 2: EUF-CMA **security for** DAPS**.**

The interesting property of a DAPS scheme is the notion of double-signature extractability (DSE). It requires that whenever one obtains signatures on two colliding messages, one should be able to extract the signing key using the extraction algorithm $\mathsf{Ex_D}$. We present the security definition denoted as DSE in Figure 3. Thereby, we consider the common notion which requires extraction to work if the key pair has been generated honestly. In this game, the adversary is given a key pair and outputs two colliding messages and corresponding signatures. The adversary wins the game if the key produced by $\mathsf{Ex_D}$ is different from the signing key, although extraction should have succeeded, i.e, the messages were colliding and their signatures were valid.

Definition 6 (DSE [PS14])**.** *For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of double-signature extraction (*DSE*) as*

$$\mathsf{Adv}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right]$$

where the corresponding experiment is depicted in Figure 3. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) \leq \varepsilon(\kappa),$$

then DAPS *provides* DSE*.*

```
$\mathsf{Exp}^{\mathsf{DSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{sk_D}, \mathsf{pk_D}) \leftarrow \mathsf{KGen_D}(1^\kappa)$
  $(m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(\mathsf{sk_D}, \mathsf{pk_D})$
  return 0, if $m_1$ and $m_2$ are not colliding
  return 0, if $\mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i) = 0$ for any $i \in [2]$
  $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$
  return 1, if $\mathsf{sk}'_\mathsf{D} \neq \mathsf{sk_D}$
  return 0
```

Fig. 3: DSE **security for** DAPS**.**

In Appendix A we recall the strong variant of extractability under malicious keys (denoted as DSE*), where the adversary is allowed to generate the key arbitrarily. The DSE* notion is very interesting from a theoretical perspective, but no practically efficient DAPS construction can achieve this notion so far.

DRS in [DRS18b] argue that when DAPS are constructed by extending a conventional signature scheme Σ, extraction of the part of the signing key corresponding to Σ is already sufficient to disincentivizes double-authentication for many applications. Hence, Derler et al. [DRS18b] defined two weaker double-signature extraction notions that cover extraction of the signing key of the underlying signature scheme for honestly and maliciously generated DAPS keys. The security games for weak double-signature extraction (wDSE) and weak double-signature extraction under malicious keys (wDSE*) are depicted in Figure 4 and Figure 5. DSE and DSE* imply their weaker counterparts and wDSE* implies wDSE.

Definition 7 ($T \in \{\mathsf{wDSE}, \mathsf{wDSE}^*\}$)**.** *For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of weak double-signature extraction ($T =$ wDSE) and weak double-signature extraction under malicious keys ($T =$ wDSE*), as*

$$\mathsf{Adv}^{T}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = \Pr\left[\mathsf{Exp}^{T}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right]$$

where the corresponding experiments are depicted in Figure 4 and Figure 5 respectively. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{T}_{\mathcal{A},\mathsf{DAPS}}(\kappa) \leq \varepsilon(\kappa),$$

then DAPS *provides* T.

```
$\mathsf{Exp}^{\mathsf{wDSE}}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
    $(\mathsf{sk_D}, \mathsf{pk_D}) \leftarrow \mathsf{KGen_D}(1^\kappa)$ with $\mathsf{sk_D} = (\mathsf{sk}_\Sigma, \dots)$
    $(m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(\mathsf{sk_D}, \mathsf{pk_D})$
    return 0, if $m_1$ and $m_2$ are not colliding
    return 0, if $\mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i) = 0$ for any $i \in [2]$
    $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$ where $\mathsf{sk}'_\mathsf{D} = (\mathsf{sk}'_\Sigma, \dots)$
    return 1, if $\mathsf{sk}'_\Sigma \neq \mathsf{sk}_\Sigma$
    return 0
```

Fig. 4: wDSE **security for** DAPS**.**

```
$\mathsf{Exp}^{\mathsf{wDSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
    $(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(1^\kappa)$ where $\mathsf{pk_D} = (\mathsf{pk}_\Sigma, \dots)$
    return 0, if $m_1$ and $m_2$ are not colliding
    return 0, if $\mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i) = 0$ for any $i \in [2]$
    $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$ where $\mathsf{sk}'_\mathsf{D} = (\mathsf{sk}'_\Sigma, \dots)$
    return 1, if $\mathsf{sk}'_\Sigma$ is not the secret key corresponding to $\mathsf{pk}_\Sigma$
    return 0
```

Fig. 5: wDSE* **security for** DAPS**.**

Finally, for our constructions we may sometimes require a very mild additional property of DAPS which we call *verifiability of secret keys*. Informally it requires that there is an additional efficient algorithm VKey which, given a key pair, outputs 1 if the given secret key is the key corresponding to the given public key. Formally we define verifiability of keys as follows:

Definition 8 (Verifiability of Keys). *We say that a DAPS scheme* $\mathsf{DAPS} = (\mathsf{KGen_D}, \mathsf{Sign_D}, \mathsf{Verify_D}, \mathsf{Ex_D})$ *provides verifiability of keys, if it provides an additional efficient algorithm* VKey *so that for all* $\kappa \in \mathbb{N}$*, for all* $(\mathsf{sk}, \mathsf{pk})$ *it holds that*

$$\mathsf{VKey}(\mathsf{sk}, \mathsf{pk}) = 1 \implies (\mathsf{sk}, \mathsf{pk}) \in \mathsf{KGen_D}(1^\kappa).$$

2.3 Non-Interactive ZK Proof Systems (NIZK)

We recall a standard definition of non-interactive zero-knowledge proof systems. Let $L \subseteq \mathsf{X}$ be an **NP**-language with associated witness relation R so that $L = \{x \mid \exists w : R(x, w) = 1\}$.

Definition 9 (Non-Interactive Zero-Knowledge Proof System). *A non-interactive proof system* Π *is a tuple of algorithms* $(\mathsf{Setup}_\Pi, \mathsf{Proof}_\Pi, \mathsf{Verify}_\Pi)$, *which are defined as follows:*

$\mathsf{Setup}_\Pi(1^\kappa)$: *This algorithm takes a security parameter* κ *as input, and outputs a common reference string* crs.

$\mathsf{Proof}_\Pi(\mathsf{crs}, x, w)$: *This algorithm takes a common reference string* crs*, a statement* x*, and a witness* w *as input, and outputs a proof* π*.*

$\mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi)$: *This algorithm takes a common reference string* crs*, a statement* x*, and a proof* π *as input, and outputs a bit* $b \in \{0, 1\}$*.*

From a non-interactive zero-knowledge proof system we require *completeness*, *soundness* and *adaptive zero-knowledge* and *simulation-sound extractability*. In Appendix C we recall formal definitions of those properties.

NIZK from Σ-protocols. A Σ-protocol for language L is an interactive three move protocol between a prover and a verifier, where the prover proves knowledge of a witness w to the statement $x \in L$. We recall the formal definition of Σ-protocols in Appendix B. One can obtain a non-interactive proof system with the above properties by applying the Fiat-Shamir transform [FS86] to any Σ-protocol where the min-entropy μ of the commitment a sent in the first message of the Σ-protocol is so that $2^{-\mu}$ is negligible in the security parameter κ and its challenge space C is exponentially large in the security parameter. Essentially, the transform removes the interaction between the prover and the verifier by using a hash function H (modelled as a random oracle) to obtain the challenge. That is, the algorithm $\mathsf{Challenge}$ obtains the challenge as $H(\mathsf{a}, x)$. Due to the lack of space we postpone a formal presentation to Appendix C.1.

Efficient NIZK Proof Systems for General Circuits. Over the last few years NIZK proof systems for general circuits have seen significant progress improving their overall efficiency. Based on the MPC-in-the-head paradigm by Ishai et al. [IKOS09], ZKBoo [GMO16] and the optimized version ZKB++ [CDG+17a] are zero-knowledge proof systems covering languages over arbitrary circuits. They roughly work as follows: The prover simulates all parties of a multiparty computation (MPC) protocol implementing the joint evaluation of some function, say $y = \text{SHA-3}(x)$, and computes commitments to the states of all players. The verifier then randomly corrupts a subset of the players and checks whether those players performed the computation correctly. Following the same paradigm, Katz et al. [KKW18] recently proposed to use a MPC protocol with a preprocessing phase, which allows to significantly reduce the proof sizes. This proof system, denoted as KKW, allows one to choose a larger number of players then in the case of ZKBoo and ZKB++, where larger numbers lead to smaller proofs. For all three proof systems, the number of binary multiplication gates is the main factor influencing the proof size, as the proof size grows linearly with the number of those gates.

Finally, Ames et al. [AHIV17] introduced Ligero, which offers proofs of logarithmic size in the number of multiplication gates if the circuit is represented using a prime field. When considering binary circuits, the number of addition respectively XOR gates has also to be accounted for in the proof size. But, as noted by Katz et al. in [KKW18], especially for large circuits with more than 100,000 gates Ligero beats ZKBoo, ZKB++ and KKW in term of proof size.

2.4 Shamir's Secret Sharing

Shamir's (k, ℓ)-threshold secret sharing [Sha79] is a secret sharing scheme which allows to information-theoretically share a secret s among a set of ℓ parties so that any collection of at least k shares allow to reconstruct s. Let s be the constant term of an otherwise randomly chosen $k-1$ degree polynomial

$$f(X) = \rho_{k-1}X^{k-1} + \cdots + \rho_1 X + s$$

over a finite field $\mathbb{F}$. A share is computed as $f(i)$ for party i, $1 \leq i \leq \ell$. Let $\mathcal{S}$ be any set of cardinality at least k of these ℓ shares and let $I_{\mathcal{S}}$ be the set of indices corresponding to shares in $\mathcal{S}$. Using Lagrange interpolation one can then can reconstruct the secret s by computing $s = f(0)$ as

$$s = \sum_{j \in I_{\mathcal{S}}} \lambda_j f(j) \quad \text{with} \quad \lambda_j = \prod_{i \in I_{\mathcal{S}} \setminus \{j\}} \frac{j}{j-i}.$$

As long as only $k-1$ or less shares are available the secret s is information-theoretically hidden.

3 DAPS without Structured Hardness Assumptions

For our first construction we follow the basic idea of Derler et al. [DRS18b] and build DAPS by including secret shares of the signing key in the signatures. To resolve the address space limitation of their approach, however, we derive the coefficients of the sharing polynomial using a pseudorandom function (PRF). By then additionally proving the correct evaluation of the PRF, it is no longer necessary to store encrypted versions of the coefficients in the public key. The only issue which remains, is to additionally prove consistency with respect to a "commitment" to the PRF secret key contained in the public key (we commit to it using a fixed-value key-binding PRF as defined in Appendix D). To bind the message to the proof, we use a signature-of-knowledge style methodology [CL06].

More precisely, we start from a one-way function $f : S \to P$, which we use to define the relation between public and secret keys, i.e., so that $\mathsf{pk}_\Sigma = f(\mathsf{sk}_\Sigma)$. In addition we carefully choose a PRF $\mathcal{F}$, which maps to the secret key space S. At the core of our DAPS construction we use a NIZK proof to prove consistency of the secret signing key, as well as the correctness of the secret sharing. For this proof we define an language L with associated witness relation R in the following way:

$$\begin{aligned}&((\mathsf{pk}_\Sigma, \beta, c, a, z), (\mathsf{sk}_\Sigma, \mathsf{sk}_{\mathsf{PRF}}, \rho)) \in R \iff \\ &\qquad \rho = \mathcal{F}(\mathsf{sk}_{\mathsf{PRF}}, a) \ \wedge \ z = \rho p + \mathsf{sk}_\Sigma \ \wedge \ c = \mathcal{F}(\mathsf{sk}_{\mathsf{PRF}}, \beta) \ \wedge \ \mathsf{pk}_\Sigma = f(\mathsf{sk}_\Sigma)\end{aligned}$$

In this statement we cover three aspects: First, we prove that the polynomial for Shamir's secret sharing is derived from the address and that the secret share is correctly calculated. Second, we prove the relation between the secret and public key of the signature scheme. Third, we "commit" to the PRF secret key using a fixed-value key-binding PRF. The full scheme is depicted in Scheme 1.

```
$\mathsf{KGen_D}(1^\kappa)$: Fix a signature scheme $\Sigma = (\mathsf{KGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$, a value-key-binding
    PRF $\mathcal{F} : \mathcal{S} \times D \to \mathsf{R}$ with respect to $\beta \in D$. Let $\mathsf{sk_{PRF}} \xleftarrow{R} \mathcal{S}$, and $\mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa)$.
    Let $c = \mathcal{F}(\mathsf{sk_{PRF}}, \beta)$. Set $\mathsf{sk_D} \leftarrow (\mathsf{sk}_\Sigma, \mathsf{sk_{PRF}})$, $\mathsf{pk_D} \leftarrow (\mathsf{pk}_\Sigma, \mathsf{crs}, \beta, c)$.
$\mathsf{Sign_D}(\mathsf{sk_D}, m)$: Parse $\mathsf{sk_D}$ as $(\mathsf{sk}_\Sigma, \mathsf{sk_{PRF}})$ and $m$ as $(a, p)$.
    1. $\rho \leftarrow \mathcal{F}(\mathsf{sk_{PRF}}, a)$
    2. $z \leftarrow \rho p + \mathsf{sk}_\Sigma$
    3. $\pi \leftarrow \mathsf{Proof}_\Pi(\mathsf{crs}, (\mathsf{pk}_\Sigma, \beta, c, a, z, m), (\mathsf{sk}_\Sigma, \mathsf{sk_{PRF}}, \rho))$
    4. Return $(z, \pi)$.
$\mathsf{Verify_D}(\mathsf{pk_D}, m, \sigma)$: Parse $\mathsf{pk_D}$ as $(\mathsf{pk}_\Sigma, \mathsf{crs}, \beta, c)$, $m$ as $(a, p)$ and $\sigma$ as $(z, \pi)$.
    1. Return $\mathsf{Verify}_\Pi(\mathsf{crs}, (\mathsf{pk}_\Sigma, \beta, c, a, z, m), \pi)$.
$\mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$: Parse $\sigma_i$ as $(z_i, \cdot)$, $m_i$ as $(a_i, p_i)$.
    1. If $m_1$ and $m_2$ are not colliding, return $\perp$
    2. if $\mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i) = 0$ for any $i$, return $\perp$
    3. let $\mathsf{sk}_\Sigma \leftarrow \frac{z_1 p_2 - z_2 p_1}{p_2 - p_1}$
    4. return $\mathsf{sk}_\Sigma$
```

Scheme 1: Generic DAPS from Σ.

It is important to note that the PRF needs to be compatible with the signature scheme, in the sense that secret-key space of Σ, i.e., S, and R match. For simplicity, we assume that $\mathsf{R} = S$. Additionally, the domain and codomain of the PRF also define the message space of the DAPS. In the following theorem we prove that Scheme 1 is an EUF-CMA-secure DAPS.

Theorem 1. *If the NIZK proof system* Π *is simulation-sound extractable,* $\mathcal{F}$ *is a PRF, and* f *is an OWF, then Scheme 1 provides* EUF-CMA *security.*

Proof. We prove this theorem using a sequence of games. We denote the winning event of game G_i as S_i. We let Q_Σ be the number of signing oracle queries.

Game 0: The original game.

Game 1: As before, but we modify $\mathsf{KGen_D}$ as follows:

$\mathsf{KGen_D}(1^\kappa)$: As before, but let $\boxed{(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_{1,\Pi}(1^\kappa)}$ and store $\boxed{\tau}$.

Transition $0 \Rightarrow 1$**:** Both games are indistinguishable under adaptive zero-knowledge of the proof system, i.e. $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{Adv}^{\mathsf{Sim}}_{\mathcal{A},\mathsf{S},\Pi}(\kappa)$.

Game 2: As Game 1, but we modify $\mathsf{Sign_D}$ as follows:

$\mathsf{Sign_D}(\mathsf{sk}, m)$: As before, but let $\boxed{\pi \leftarrow \mathcal{S}_{2,\Pi}(\mathsf{crs}, \tau, (\mathsf{pk}_\Sigma, \beta, c, a, z, m))}$.

Transition $1 \Rightarrow 2$**:** Both games are indistinguishable under adaptive zero-knowledge of the proof system, i.e. $|\Pr[S_1] - \Pr[S_2]| \leq \mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A},\mathsf{S},\Pi}(\kappa)$.

Game 3: As before, but we modify $\mathsf{KGen_D}$ and $\mathsf{Sign_D}$ as follows.

$\mathsf{KGen_D}(1^\kappa)$: As before, but let $\boxed{c \xleftarrow{R} \mathsf{R}}$.

$\mathsf{Sign_D}(\mathsf{sk_D}, m)$: As before, but let $\boxed{\rho \xleftarrow{R} \mathsf{R}}$.

Transition $2 \Rightarrow 3$**:** We engage with a PRF challenger $\mathcal{C}$ against $\mathcal{F}$. We modify $\mathsf{Sign_D}$ as follows:

$\mathsf{KGen_D}(1^\kappa)$: As before, but let $\boxed{c \xleftarrow{R} \mathcal{C}(\beta)}$.

$\mathsf{Sign_D}(\mathsf{sk_D}, m)$: As before, but let $\boxed{\rho \xleftarrow{R} \mathcal{C}(a)}$.

Thus an adversary distinguishing the two games also distinguishes the PRF from a random function, i.e. $|\Pr[S_4] - \Pr[S_3]| \leq \mathsf{Adv}_{\mathcal{D},F}(\kappa)$.

Game 4: As before, but we modify Sign_D as follows.

$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)\colon$ As before, but track all (a, ρ) pairs in $\mathcal{Q}$.
We abort if there exists $(a_1, \rho), (a_2, \rho) \in \mathcal{Q}$ such that $a_1 \neq a_2$.

Transition $3 \Rightarrow 4$**:** Both games proceed identically, unless the abort event happens. The probability of the abort event is bounded by $^1/_{|\mathsf{R}|}$, i.e. $|\Pr[S_5] - \Pr[S_4]| \leq {}^{Q_\Sigma}/_{|\mathsf{R}|}$.

Game 5: As before, but we modify Sign_D as follows.

$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)\colon$ As before, but let $\boxed{z \xleftarrow{R} \mathsf{R}}$.

Transition $4 \Rightarrow 5$**:** This change is conceptional. Note that ρ is uniformly random and not revealed, and thus z is uniformly random.

Game 6: As before, but we modify KGen_D as follows:

$\mathsf{KGen}_\mathsf{D}(1^\kappa)\colon$ As before, but let $\boxed{(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_{1,\Pi}(1^\kappa)}$ and store $\boxed{(\tau, \xi)}$.

Transition $5 \Rightarrow 6$**:** Both games are indistinguishable under simulation-sound extractability of the proof system, i.e. $|\Pr[S_6] - \Pr[S_5]| \leq \mathsf{Adv}^{\mathsf{Ext}_1}_{\mathcal{A},\mathcal{E},\Pi}(\kappa)$.

Game 7: As before, but we now use the extractor to obtain $\mathsf{sk}^*_\Sigma \leftarrow \mathcal{E}_{2,\Pi}(\mathsf{crs}, \xi, (\mathsf{pk}_\Sigma, \beta, c, a, z, m), \pi)$ and abort in case the extraction fails.

Transition $6 \Rightarrow 7$**:** Both games proceed identically, unless we abort. The probability of that happening is bounded by the simulation-sound extractablity of the proof system, i.e. $|\Pr[S_7] - \Pr[S_6]| \leq \mathsf{Adv}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa)$.

Reduction. Now we are ready to present a reduction which engages with an OWF challenger $\mathcal{C}$. In particular, we obtain a challenge and embed it in the public key, i.e.

$\mathsf{KGen}_\mathsf{D}(1^\kappa)\colon$ As before, but $\boxed{\mathsf{pk}_\Sigma \leftarrow \mathcal{C}}$.

Once the adversary returns a forgery, we extract sk^*_Σ and forward the solution to the OWF challenger. Hence $\Pr[S_7] \leq \mathsf{Adv}^{\mathsf{OWF}}_{\mathcal{A},f}(\kappa)$, which concludes the proof. □

We now show that Scheme 1 also provides wDSE security. We note that in the proof of Theorem 2 we do not need to simulate proofs, so a weaker extraction notion would suffice. The proof of Theorem 1, however, already requires simulation-sound extractability which is why we directly resort to simulation-sound extractability.

Theorem 2. *If the NIZK proof system* Π *is simulation-sound extractable and the PRF* $\mathcal{F}$ *is computationally fixed-value-key-binding, then Scheme 1 provides* wDSE *security.*

Proof. We prove this theorem using a sequence of games. We denote the winning event of game G_i as S_i. Let $m_1, m_2, \sigma_1, \sigma_2$ denote the output of $\mathcal{A}$. For simplicity we write $m_j = (a, p_j)$, $\sigma_j = (z_j, \pi_j)$ for $j \in [2]$. Now, we have proofs attesting that $z_j = \rho p_j + \mathsf{sk}_\Sigma$ for $j \in [2]$.

Game 0: The original game.

Game 1: As before, but we modify KGen_D as follows:

$\mathsf{KGen}_\mathsf{D}(1^\kappa)$: As before, but let $\boxed{(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_{1,\Pi}(1^\kappa)}$ and store $\boxed{\tau}$.

Transition $0 \Rightarrow 1$**:** Both games are indistinguishable under adaptive zero-knowledge of the proof system, i.e. $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{Adv}^{\mathsf{Sim}}_{\mathcal{A},\mathsf{S},\Pi}(\kappa)$.

Game 2: As before, but we modify KGen_D as follows:

$\mathsf{KGen}_\mathsf{D}(1^\kappa)$: As before, but let $\boxed{(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_{1,\Pi}(1^\kappa)}$ and store $\boxed{\xi}$.

Transition $1 \Rightarrow 2$**:** Both games are indistinguishable under simulation-sound extractability of the proof system, i.e. $|\Pr[S_2] - \Pr[S_1]| \leq \mathsf{Adv}^{\mathsf{Ext}_1}_{\mathcal{A},\mathcal{E},\Pi}(\kappa)$.

Game 3: As before, but we now use the extractor to obtain $(\mathsf{sk}^*_{\Sigma,j}, \mathsf{sk}^*_{\mathsf{PRF},j}) \leftarrow \mathcal{E}_{2,\Pi}(\mathsf{crs}, \xi, (\mathsf{pk}_\Sigma, \beta, c, a, z_j, m_j), \pi)$ for $j \in [2]$ and abort if the extraction fails.

Transition $2 \Rightarrow 3$**:** Both games proceed identically, unless we abort. The probability of that happening is bounded by the simulation-sound extractablity of the proof system, i.e. $|\Pr[S_3] - \Pr[S_2]| \leq 2 \cdot \mathsf{Adv}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa)$.

Game 4: As before, but we abort if $\mathsf{sk}_\mathsf{PRF} \neq \mathsf{sk}^*_{\mathsf{PRF},j}$ for any $j \in [2]$.

Transition $3 \Rightarrow 4$**:** Both games proceed identically, unless we abort. Let $j \in [2]$ be such that $\mathsf{sk}_\mathsf{PRF} \neq \mathsf{sk}^*_{\mathsf{PRF},j}$. We bound the abort probability using $\mathcal{F}$. Let $\mathcal{C}$ be a computational fixed-value-key-binding challenger. We modify KGen_D as follows:

$\mathsf{KGen}_\mathsf{D}(1^\kappa)$: As before, but let $\boxed{(\mathsf{sk}_\mathsf{PRF}, \beta)} \leftarrow \mathcal{C}$.

Then we have that $\mathcal{F}(\mathsf{sk}_\mathsf{PRF}, \beta) = \mathcal{F}(\mathsf{sk}^*_{\mathsf{PRF},j}, \beta)$, hence we forward $\mathsf{sk}^*_{\mathsf{PRF},j}$ to $\mathcal{C}$. Thus we built an adversary $\mathcal{B}$ against fixed-value-key-binding of $\mathcal{F}$, i.e. $|\Pr[S_4] - \Pr[S_3]| \leq \mathsf{Adv}^{\mathsf{cFKVB}}_{\mathcal{B},\mathcal{F}}(\kappa) = \varepsilon(\kappa)$.

As we have now ensured that the correct PRF secret key was used to generate ρ from a, sk_Σ is now uniquely determined via the secret sharing. Thus the adversary can no longer win, i.e. $\Pr[S_4] = 0$. □

Extension to NAPS. Following the ideas outlined in [DRS18b], Scheme 1 can be extended to an N-time authentication-preventing signature scheme by changing the sharing polynomial $\rho X + \mathsf{sk}_\Sigma$ to a polynomial of degree $N-1$ with coefficients $\rho_1, \ldots, \rho_{N-1}$ obtained from the PRF via $\rho_i = \mathcal{F}(\mathsf{sk}_\mathsf{PRF}, a \| i)$.

Instantiations. The requirement on the signature scheme are very weak, yet finding a suitable combination of primitives can be difficult. Thus we discuss some possible instantiations. One candidate scheme on top of which the DAPS extension can be applied is Picnic [CDG$^+$17a,CDG$^+$17b]. In Picnic the public key pk_Σ is the image of the secret key sk_Σ under a one-way function built from LowMC [ARS$^+$15,ARS$^+$16]. Signatures are then generated by proving this relation using a NIZK from ZKB++ made non-interactive. In this case it is straight forward to use the block cipher LowMC (denoted by $\mathcal{E}$) as PRF by setting $\mathcal{F}(s, x) = \mathcal{E}(s, x) \oplus x$. We argue that this PRF can also be considered a computational fixed-value-key-binding PRF, since it is reasonable to assume that finding a new key which maps one particular input to one particular output is no easier than generic key search. Furthermore, when increasing the block size of LowMC relative to the key size, the existence of second key mapping to the same output becomes increasingly unlikely.

The circuit for the secret sharing can either be implemented using a binary circuit realizing the required arithmetic, or, more efficiently, by computing the sharing bit-wise. For the latter, we consider ρ, p and sk_Σ as n bit values, and compute secret shares $z_i = \rho_i p_i + \mathsf{sk}_{\Sigma,i}$ for each bit $i \in [n]$. Thus only n ANDs are required to implemented the secret sharing. All in all Picnic signatures can be easily extended to a DAPS without requiring extensive changes. We also note that the Fiat-Shamir transformed ZKB++ is in fact simulation-sound extractable NIZK proof systems as confirmed in [DRS18a]. Using the signature size formulas, we can estimate DAPS signatures sizes at around 408 KB, meaning there is a overhead of 293 KB compared to Picnic signatures requiring roughly 115 KB in the ROM targeting 256 bit classical security. Analogously to the QROM security of Picnic, Unruh's transform [Unr12,Unr15,Unr16] can be used to obtain QROM security for the DAPS construction.

Also hash-based signatures such as SPHINCS [BHH+15] are well suited for this construction. Similar to the case of Picnic, the PRF can be instantiated using LowMC. However, the consistency proof is more expensive, as computing the public key requires multiple evaluations of hash functions.

Relying on Structured Hardness Assumptions. The situation is different for signature schemes relying on structured hardness assumptions, e.g., those in the discrete logarithm setting such as Schnorr signatures [Sch89], ECDSA and EdDSA [BDL+12]. While they would fulfill the requirement for the secret-key-to-public-key relation, i.e., here working in a group $\mathbb{G}$ with generator g the OWF is of the form $f(x) := g^x$, the problem is finding an efficient NIZK proof system to prove statements over $\mathbb{Z}_p$ and in a prime order group $\mathbb{G}$ simultaneously. Furthermore the NIZK proof system would also need to support statements over binary circuits for the PRF evaluation. Recently, Agrawal et al. [AGM18] made progress in this direction, enabling non-interactive proofs of composite statements for relations over multiple groups and binary circuits. Using these techniques to construct DAPS is an interesting open problem.

4 Extending Any Signature Scheme Using DAPS

Finally, we follow a different direction for our second approach. Here we start from an already existing DAPS and use it to extend *any* unforgeable signature scheme to a DAPS. Interestingly, both the unforgeability and extraction follow in a black-box way from the signature scheme and the underlying DAPS, respectively. In this construction, the secret key consists of the secret keys of the underlying DAPS and signature scheme. To guarantee extraction of the full secret key, we apply the technique of Bellare et al. [BPS17] and encrypt the key of the signature scheme using a one-time pad derived from the secret key of the DAPS scheme. The public key then consists of that encrypted key and the public keys of the underlying DAPS and signature scheme. However, for extraction of maliciously generated keys, i.e., DSE^*-security, this means that public keys need to be extended with a NIZK proof that the encryption was performed correctly. For the sake of simplicity, we thus concentrate on the DSE security of the scheme. We present the compiler in Scheme 2.

In the following theorem we formally state that the DAPS construction in Scheme 2 yields an $\mathsf{EUF\text{-}CMA}$-secure DAPS.

$\mathsf{KGen}_\mathsf{D}(1^\kappa)$: Fix some signature scheme $\Sigma = (\mathsf{KGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Verify}_\Sigma)$ and some DAPS $\mathsf{DAPS} = (\mathsf{KGen}_\mathsf{D}, \mathsf{Sign}_\mathsf{D}, \mathsf{Verify}_\mathsf{D}, \mathsf{Ex}_\mathsf{D})$ with verifiability of keys. Let $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \Sigma.\mathsf{KGen}_\Sigma(1^\kappa)$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{DAPS}.\mathsf{KGen}_\mathsf{D}(1^\kappa)$, $Y \leftarrow \mathsf{sk}_\Sigma \oplus H(\mathsf{sk})$, and return $(\mathsf{sk}_\mathsf{D}, \mathsf{pk}_\mathsf{D}) := ((\mathsf{sk}_\Sigma, \mathsf{sk}), (\mathsf{pk}_\Sigma, \mathsf{pk}, Y))$.

$\mathsf{Sign}_\mathsf{D}(\mathsf{sk}_\mathsf{D}, m)$: Parse sk_D as $(\mathsf{sk}_\Sigma, \mathsf{sk})$.

1. $\sigma_0 \leftarrow \Sigma.\mathsf{Sign}_\Sigma(\mathsf{sk}_\Sigma, m)$
2. $\sigma_1 \leftarrow \mathsf{DAPS}.\mathsf{Sign}_\mathsf{D}(\mathsf{sk}, m)$
3. Return $\sigma = (\sigma_0, \sigma_1)$

$\mathsf{Verify}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m, \sigma)$: Parse pk_D as $(\mathsf{pk}_\Sigma, \mathsf{pk}, \cdot)$, and return 1 if all of the following checks hold and 0 otherwise:

- $\Sigma.\mathsf{Verify}_\Sigma(\mathsf{pk}, (a, p)) = 1$
- $\mathsf{DAPS}.\mathsf{Verify}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, (a, p)) = 1$

$\mathsf{Ex}_\mathsf{D}(\mathsf{pk}_\mathsf{D}, m_1, m_2, \sigma_1, \sigma_2)$: Parse pk_D as $(\mathsf{pk}_\Sigma, \mathsf{pk}, Y)$, obtain $\mathsf{sk} \leftarrow \mathsf{DAPS}.\mathsf{Ex}_\mathsf{D}(\mathsf{pk}, m_1, m_2, \sigma_1, \sigma_2)$ and $\mathsf{sk}_\Sigma \leftarrow Y \oplus H(\mathsf{sk})$, and return $\mathsf{sk}_\mathsf{D} = (\mathsf{sk}_\Sigma, \mathsf{sk})$.

Scheme 2: Black-Box Extension of any Signature Scheme to DAPS.

Theorem 3. *If Σ is unforgeable,* DAPS *is unforgeable and provides verifiability of keys, then the DAPS construction in Scheme 2 is unforgeable in the ROM.*

The theorem above is proven in Appendix E.1. Additionally, Scheme 1 provides DSE-security if the underlying DAPS provides it as well.

Theorem 4. *If* DAPS *provides* DSE*-security, then the construction of DAPS in Scheme 2 provides* DSE*-security as well.*

The theorem above is proven in Appendix E.2.

5 Conclusion

In this work, we close two important gaps in the literature on DAPS. First, we present a generic DAPS construction, which, in contrast to [DRS18b], does not come with the drawback of a polynomially bounded address space. Our construction only relies on assumptions related to symmetric key primitives, which is why we also obtain a candidate for a post-quantum DAPS construction. Second, we also present an alternative generic construction of DAPS which basically shows how to bring DAPS features to any signature scheme. This is of particular practical importance, as it allows to extend arbitrary signature schemes with double signature extraction features. As our compiler works by using an arbitrary DAPS scheme to extend a given signature scheme in a black-box way, this yields more efficient DAPS than previously known for standardized and widely used signature schemes such as ECDSA or EdDSA.

References

[AGM18] Shashank Agrawal, Chaya Ganesh, and Payman Mohassel. Non-interactive zero-knowledge proofs for composite statements. In *CRYPTO (3)*, volume 10993 of *Lecture Notes in Computer Science*, pages 643–673. Springer, 2018.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *CCS*, pages 2087–2104. ACM, 2017.

[ARS$^+$15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454. Springer, 2015.

[ARS$^+$16] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. *IACR Cryptology ePrint Archive*, 2016:687, 2016.

[BDL$^+$12] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *J. Cryptographic Engineering*, 2(2):77–89, 2012.

[BEF18] Dan Boneh, Saba Eskandarian, and Ben Fisch. Post-quantum group signatures from symmetric primitives. *IACR Cryptology ePrint Archive*, 2018:261, 2018.

[BHH$^+$15] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 368–397. Springer, 2015.

[BKN17] Dan Boneh, Sam Kim, and Valeria Nikolaenko. Lattice-based DAPS and generalizations: Self-enforcement in signature schemes. In *ACNS*, volume 10355 of *Lecture Notes in Computer Science*, pages 457–477. Springer, 2017.

[BPS16] Mihir Bellare, Bertram Poettering, and Douglas Stebila. From identification to signatures, tightly: A framework and generic transforms. In *ASIACRYPT (2)*, volume 10032 of *Lecture Notes in Computer Science*, pages 435–464, 2016.

[BPS17] Mihir Bellare, Bertram Poettering, and Douglas Stebila. Deterring certificate subversion: Efficient double-authentication-preventing signatures. In *Public Key Cryptography (2)*, volume 10175 of *Lecture Notes in Computer Science*, pages 121–151. Springer, 2017.

[CDG$^+$17a] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *CCS*, pages 1825–1842. ACM, 2017.

[CDG$^+$17b] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. The Picnic Signature Algorithm Specification, 2017. https://github.com/Microsoft/Picnic/blob/master/spec.pdf.

[CL06] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 78–96. Springer, 2006.

[CMR98] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *STOC*, pages 131–140. ACM, 1998.

[DRS18a] David Derler, Sebastian Ramacher, and Daniel Slamanig. Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In *PQCrypto*, volume 10786 of *Lecture Notes in Computer Science*, pages 419–440. Springer, 2018.

[DRS18b] David Derler, Sebastian Ramacher, and Daniel Slamanig. Short double- and n-times-authentication-preventing signatures from ECDSA and more. In *EuroS&P*, pages 273–287. IEEE, 2018.

[Fis99] Marc Fischlin. Pseudorandom function tribe ensembles based on one-way permutations: Improvements and applications. In *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 432–445. Springer, 1999.

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the fiat-shamir transform. In *INDOCRYPT*, volume 7668 of *Lecture Notes in Computer Science*, pages 60–79. Springer, 2012.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[GMO16] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *USENIX Security Symposium*, pages 1069–1083. USENIX Association, 2016.

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231. Springer, 1988.

[IKOS09] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM J. Comput.*, 39(3):1121–1152, 2009.

[KKW18] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. *IACR Cryptology ePrint Archive*, 2018:475, 2018.

[MR02] Silvio Micali and Leonid Reyzin. Improving the exact security of digital signature schemes. *J. Cryptology*, 15(1):1–18, 2002.

[Poe18] Bertram Poettering. Shorter double-authentication preventing signatures for small address spaces. In *AFRICACRYPT*, volume 10831 of *Lecture Notes in Computer Science*, pages 344–361. Springer, 2018.

[PS14] Bertram Poettering and Douglas Stebila. Double-authentication-preventing signatures. In *ESORICS (1)*, volume 8712 of *Lecture Notes in Computer Science*, pages 436–453. Springer, 2014.

[PS17] Bertram Poettering and Douglas Stebila. Double-authentication-preventing signatures. *Int. J. Inf. Sec.*, 16(1):1–22, 2017.

[RKS15] Tim Ruffing, Aniket Kate, and Dominique Schröder. Liar, liar, coins on fire!: Penalizing equivocation by loss of bitcoins. In *ACM Conference on Computer and Communications Security*, pages 219–230. ACM, 2015.

[Sch89] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1989.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 135–152. Springer, 2012.

[Unr15] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *EUROCRYPT (2)*, volume 9057 of *Lecture Notes in Computer Science*, pages 755–784. Springer, 2015.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 497–527. Springer, 2016.

[Unr17] Dominique Unruh. Post-quantum security of fiat-shamir. In *ASIACRYPT (1)*, volume 10624 of *Lecture Notes in Computer Science*, pages 65–95. Springer, 2017.

A DSE* Security of DAPS

We recall the DSE^* security notion of DAPS. The game is depicted in Figure 6, where in contrast to Figure 3 the keys are generated by the adversary.

Definition 10 (DSE^* [PS14]). *For a PPT adversary $\mathcal{A}$, we define the advantage function in the sense of double-signature extraction under malicious keys (DSE^*) as*

$$\mathsf{Adv}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa) = 1\right]$$

where the corresponding experiment is depicted in Figure 6. If for all PPT adversaries $\mathcal{A}$ there is a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa) \leq \varepsilon(\kappa),$$

then DAPS *provides* DSE^*.

```
$\mathsf{Exp}^{\mathsf{DSE}^*}_{\mathcal{A},\mathsf{DAPS}}(\kappa)$:
  $(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2) \leftarrow \mathcal{A}(1^\kappa)$
  return 0, if $m_1$ and $m_2$ are not colliding
  return 0, if $\mathsf{Verify_D}(\mathsf{pk_D}, m_i, \sigma_i) = 0$ for any $i \in [2]$
  $\mathsf{sk}'_\mathsf{D} \leftarrow \mathsf{Ex_D}(\mathsf{pk_D}, m_1, m_2, \sigma_1, \sigma_2)$
  return 1, if $\mathsf{sk}'$ is not the secret key corresponding to $\mathsf{pk_D}$
  return 0
```

Fig. 6: DSE^* **security for** DAPS.

B Σ-Protocols

Let $L \subseteq \mathsf{X}$ be an **NP**-language with associated witness relation R so that $L = \{x \mid \exists w : R(x, w) = 1\}$. A Σ-protocol for language L is defined as follows.

Definition 11. *A Σ-protocol for language L is an interactive three-move protocol between a PPT prover $\mathsf{P} = (\mathsf{Commit}, \mathsf{Prove})$ and a PPT verifier $\mathsf{V} = (\mathsf{Challenge}, \mathsf{Verify})$, where P makes the first move and transcripts are of the form $(\mathsf{a}, \mathsf{c}, \mathsf{s}) \in \mathsf{A} \times \mathsf{C} \times \mathsf{S}$. Additionally they satisfy the following properties:*

Completeness *A Σ-protocol for language L is complete, if for all security parameters κ, and for all $(x, w) \in R$, it holds that*

$$\Pr[\langle \mathsf{P}(1^\kappa, x, w), \mathsf{V}(1^\kappa, x)\rangle = 1] = 1.$$

Special Soundness *A Σ-protocol for language L is special sound, if there exists a PPT extractor $\mathcal{E}$ so that for all x, and for all sets of accepting transcripts $\{(\mathsf{a}, \mathsf{c}_i, \mathsf{s}_i)\}_{i\in[2]}$ with respect to x where $\mathsf{c}_1 \neq \mathsf{c}_2$, generated by any algorithm with polynomial runtime in κ, it holds that*

$$\Pr\left[w \leftarrow \mathcal{E}(1^\kappa, x, \{(\mathsf{a}, \mathsf{c}_i, \mathsf{s}_i)\}_{i\in[2]}) \ : \ (x, w) \in R\right] \geq 1 - \varepsilon(\kappa).$$

Special Honest-Verifier Zero-Knowledge *A Σ-protocol is special honest-verifier zero-knowledge, if there exists a PPT simulator $\mathcal{S}$ so that for every $x \in L$ and every challenge c from the challenge space, it holds that a transcript $(\mathsf{a}, \mathsf{c}, \mathsf{s})$, where $(\mathsf{a}, \mathsf{s}) \leftarrow \mathcal{S}(1^\kappa, x, \mathsf{c})$ is indistinguishable from a transcript resulting from an honest execution of the protocol.*

C NIZK Security Properties

Definition 12 (Completeness). *A non-interactive proof system for language L is complete, if for all $\kappa \in \mathbb{N}$, for all $\mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa)$, for all $x \in L$, for all w such that $R(x, w) = 1$, and for all $\pi \leftarrow \mathsf{Proof}_\Pi(\mathsf{crs}, x, w)$, we have that $\mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi) = 1$.*

This captures perfect completeness.

Definition 13 (Soundness). *For an efficient adversary $\mathcal{A}$, we define the advantage function in the sense of soundness as*

$$\mathsf{Adv}^{\mathsf{Sound}}_{\mathcal{A},\Pi}(\kappa) = \Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa), \\ (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs})\end{array} : \begin{array}{r}\mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi) = 1 \\ \wedge\ x \notin L\end{array}\right].$$

If for any efficient adversary $\mathcal{A}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{Sound}}_{\mathcal{A},\Pi}(\kappa) \leq \varepsilon(\kappa),$$

Π *is sound.*

Definition 14 (Adaptive Zero-Knowledge). *For an efficient simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ and an efficient adversary $\mathcal{A}$, we define the advantage functions in the sense of zero-knowledge as*

$$\mathsf{Adv}^{\mathsf{Sim}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) = \left|\begin{array}{l}\Pr\left[\mathsf{crs} \leftarrow \mathsf{Setup}_\Pi(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1\right] - \\ \Pr\left[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1\right]\end{array}\right|$$

and

$$\mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) = \left|\Pr\left[\mathsf{Exp}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) = 1\right] - \frac{1}{2}\right|$$

where the corresponding experiment is depicted in Figure 7. If there exists an efficient simulator $\mathsf{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that for any efficient adversary $\mathcal{A}$ there exist negligible functions $\varepsilon_1(\cdot)$ and $\varepsilon_2(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{Sim}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) \leq \varepsilon_1(\kappa) \text{ and } \mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S},\Pi}(\kappa) \leq \varepsilon_2(\kappa)$$

then Π provides adaptive zero-knowledge.

```
$\mathsf{Exp}^{\mathsf{ZK}}_{\mathcal{A},\mathsf{S},\Pi}(\kappa)$:
  $b \leftarrow \{0,1\}$
  $(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{P}_b(\cdot,\cdot)}(\mathsf{crs})$
    where oracle $\mathsf{P}_0$ on input $(x, w)$:
      return $\pi \leftarrow \mathsf{Proof}_\Pi(\mathsf{crs}, x, w)$, if $(x, w) \in R$
      return $\bot$
    and oracle $\mathsf{P}_1$ on input $(x, w)$:
      return $\pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, x)$, if $(x, w) \in R$
      return $\bot$
  return 1, if $b = b^*$
  return 0
```

Fig. 7: Adaptive Zero-Knowledge

Definition 15 (Simulation-Sound Extractability). *For an adaptively zero-knowledge non-interactive proof system Π, for an efficient extractor extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ and an efficient adversary $\mathcal{A}$, we define the advantage functions in the sense of simulation-sound extractability as*

$$\mathsf{Adv}^{\mathsf{Ext}_1}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) = \left| \begin{array}{l} \Pr\left[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1\right] - \\ \Pr\left[(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_1(1^\kappa) : \mathcal{A}(\mathsf{crs}) = 1\right] \end{array} \right|$$

and

$$\mathsf{Adv}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) = 1\right] - \frac{1}{2}$$

where the corresponding experiment is depicted in Figure 8. If there exists an efficient extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ such that for any efficient adversary $\mathcal{A}$ there exist negligible functions $\varepsilon_1(\cdot)$ and $\varepsilon_2(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{Ext}_1}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) \leq \varepsilon_1(\kappa) \text{ and } \mathsf{Adv}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa) \leq \varepsilon_2(\kappa)$$

then Π provides simulation-sound extractactability.

C.1 NIZK from Σ-Protocols

To convert a Σ-protocol to a NIZK, $\mathsf{Setup}_\Pi(1^\kappa)$ fixes a hash function $H : \mathsf{A} \times \mathsf{X} \to \mathsf{C}$, sets $\mathsf{crs} \leftarrow (\kappa, H)$ and returns crs. The algorithms Proof_Π and Verify_Π are defined as follows:

$\mathsf{Proof}_\Pi(\mathsf{crs}, x, w)$: Start P on $(1^\kappa, x, w)$, obtain the first message a, answer with $\mathsf{c} \leftarrow H(\mathsf{a}, x)$. Finally obtain s and return $\pi \leftarrow (\mathsf{a}, \mathsf{s})$.

$\mathsf{Verify}_\Pi(\mathsf{crs}, x, \pi)$: Parse π as (a, s). Start V on $(1^\kappa, x)$ and send a as first message to the verifier. When V outputs c, reply with s and output 1 if V accepts and 0 otherwise.

Combining [FKMV12, Thm. 1, Thm. 2, Thm. 3, Prop. 1] (among others) shows that a so-obtained proof system is complete, sound, adaptively zero-knowledge,

```
$\mathsf{Exp}^{\mathsf{Ext}_2}_{\mathcal{A},\mathcal{E},\Pi}(\kappa)$:
  $(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_1(1^\kappa)$
  $\mathcal{Q}_\mathcal{S} = \emptyset$
  $(x^*, w^*) \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot)}(\mathsf{crs})$
    where oracle $\mathcal{S}$ on input $(x, w)$:
      $\mathcal{Q}_\mathcal{S} \leftarrow \mathcal{Q}_\mathcal{S} \cup \{(x, w)\}$
      return $\pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, x)$, if $(x, w) \in R$
      return $\perp$
  $w \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi, x^*, \pi^*)$
  return 1, if $\mathsf{Verify}_\Pi(\mathsf{crs}, x^*, \pi^*) = 1 \wedge (x^*, \pi^*) \notin \mathcal{Q}_\mathcal{S} \wedge (x^*, w) \notin R$
  return 0
```

Fig. 8: Simulation-sound extractability

if the underlying Σ-protocol is special sound and the commitments sent in the first move are unconditionally binding. Security of the Fiat-Shamir transform in the quantum-accessible ROM (QROM) requires stronger properties of the Σ-protocols [Unr17], however Unruh's transform [Unr12,Unr15,Unr16] can be used to obtain QROM-secure NIZKs from Σ-protocols.

D One-way Functions and Pseudorandom Function Families

We recall the definitions of one-way functions and pseudorandom function (families).

Definition 16 (OWF). *Let $f : S \to P$ be a function. For a PPT adversary $\mathcal{A}$ we define the advantage function as*

$$\mathsf{Adv}^{\mathsf{OWF}}_{\mathcal{A},f}(\kappa) = \Pr\left[x \xleftarrow{R} S, x^* \leftarrow \mathcal{A}(1^\kappa, f(x)) : f(x) = f(\mathcal{A}^*)\right].$$

The function f is one-way function (OWF) if it is efficiently computable and for all PPT adversaries $\mathcal{A}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{OWF}}_{\mathcal{A},f}(\kappa) \leq \varepsilon(\kappa).$$

Definition 17 (PRF). *Let $\mathcal{F} : \mathcal{S} \times D \to \mathsf{R}$ be a family of functions and let Γ be the set of all functions $D \to \mathsf{R}$. For a PPT distinguisher $\mathcal{D}$ we define the advantage function as*

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{D},\mathcal{F}}(\kappa) = \left|\Pr\left[s \xleftarrow{R} \mathcal{S}, \mathcal{D}^{\mathcal{F}(s,\cdot)}(1^\kappa)\right] - \Pr[f \xleftarrow{R} \Gamma, \mathcal{D}^{f(\cdot)}(1^\kappa)]\right|.$$

$\mathcal{F}$ is a pseudorandom function (family) if it is efficiently computable and for all PPT distinguishers $\mathcal{D}$ there exists a negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{D},\mathcal{F}}(\kappa) \leq \varepsilon(\kappa).$$

Below, we provide a slightly stronger variant of a definition of a notion introduced in [CMR98,Fis99].

Definition 18 (Fixed-Value-Key-Binding PRF). *A PRF family $\mathcal{F} : \mathcal{S} \times D \to \mathsf{R}$ and a $\beta \in D$, is fixed-value-key-binding if for all adversaries $\mathcal{A}$*

$$\Pr\left[s \xleftarrow{R} \mathcal{S}, s' \leftarrow \mathcal{A}(s, \beta) : \mathcal{F}(s, \beta) = \mathcal{F}(s', \beta) \ \wedge \ s \neq s'\right] = 0.$$

Moreover, we present a relaxed (computational) version of the above definition.

Definition 19 (Computational Fixed-Value-Key-Binding PRF). *For a PRF family $\mathcal{F} : \mathcal{S} \times D \to \mathsf{R}$ and a $\beta \in D$, we define the advantage function of a PPT adversary $\mathcal{A}$ as*

$$\mathsf{Adv}^{\mathsf{cFKVB}}_{\mathcal{A},\mathcal{F}}(\kappa) = \Pr\left[s \xleftarrow{R} \mathcal{S}, s' \leftarrow \mathcal{A}(1^\kappa, s, \beta) : \mathcal{F}(s, \beta) = \mathcal{F}(s', \beta) \ \wedge \ s \neq s'\right].$$

$\mathcal{F}$ is computationally fixed-value-key-binding if for all PPT adversaries there exists as negligible function $\varepsilon(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{cFKVB}}_{\mathcal{A},\mathcal{F}}(\kappa) = \varepsilon(\kappa).$$

E Security Proofs

E.1 Proof of Theorem 3

Proof. To prove the theorem above, we proceed in a sequence of games where we play $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{DAPS},\mathcal{A}}(\kappa)$ with the DAPS in Scheme 1 and adversary $\mathcal{A}$.

Game 0: The original unforgeability game.

Game 1: As Game 0, but we choose Y uniformly at random and abort as soon as $\mathcal{A}$ queries the random oracle H on sk with $\mathsf{VKey}(\mathsf{sk}, \mathsf{pk}) = 1$.

Transition $0 \Rightarrow 1$: Let this event be called E. The distributions in Game 0 and Game 1 are identical unless E happens. We bound the probability of E to happen by constructing an adversary $\mathcal{B}$ with

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B},\mathsf{DAPS}}(\kappa) \geq \Pr[E].$$

To do so, we honestly generate $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma)$ and engage in an experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B},\mathsf{DAPS}}(\kappa)$ to obtain pk for DAPS. We choose Y uniformly at random, and set $(\mathsf{sk}_\mathsf{D}, \mathsf{pk}_\mathsf{D}) \leftarrow ((\mathsf{sk}_\Sigma, \perp), (\mathsf{pk}_\Sigma, \mathsf{pk}_\mathsf{D}, Y))$. Whenever a signature for DAPS is required, we use the signing oracle provided by $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B},\mathsf{DAPS}}(\kappa)$. If E happens, we have that $\mathsf{VKey}(\mathsf{sk}, \mathsf{pk}) = 1$, which—by the correctness of DAPS—means that we can choose an arbitrary unqueried message m from the message space of DAPS which satisfies the winning condition, and output $(m, \mathsf{DAPS.Sign}_\mathsf{D}(\mathsf{sk}, m))$ as a forgery for DAPS. All in all, we thus have that $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B},\mathsf{DAPS}}(\kappa)$.

Reduction. Now we are ready to show that the winning probability in Game 1 is bounded by $\max\{\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_1,\Sigma}(\kappa), \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_2,\mathsf{DAPS}}(\kappa)\}$. To do so, we construct two reductions which use $\mathcal{A}$ to construct $\mathcal{B}_1$ or $\mathcal{B}_2$ respectively. Both $\mathcal{B}_1$ and $\mathcal{B}_2$ will succeed whenever $\mathcal{A}$ succeeds.

$\mathcal{B}_1$: In this case, we engage in an experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_1,\Sigma}(\kappa)$ to obtain pk_Σ. We choose Y uniformly at random, obtain $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{DAPS.KGen}_\mathsf{D}(1^\kappa)$ and set $(\mathsf{sk}_\mathsf{D}, \mathsf{pk}_\mathsf{D}) \leftarrow ((\perp, \mathsf{sk}), (\mathsf{pk}_\Sigma, \mathsf{pk}, Y))$. Whenever a $\boldsymbol{\Sigma}$ signature is required, the signature is obtained using the oracle provided by the experiment. If the adversary eventually outputs a forgery $(m^*, \sigma^*) = (m^*, (\sigma_0^*, \sigma_1^*))$ we output (m^*, σ_0^*) as a forgery to win $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_1,\Sigma}(\kappa)$. Clearly, $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathrm{Game}\ 1}(\kappa) \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_1,\Sigma}(\kappa)$.

$\mathcal{B}_2$: In this case, we engage in an experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_2,\mathsf{DAPS}}(\kappa)$ to obtain pk. We choose Y uniformly at random, obtain $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \Sigma.\mathsf{KGen}_\Sigma(1^\kappa)$ and set $(\mathsf{sk}_\mathsf{D}, \mathsf{pk}_\mathsf{D}) \leftarrow ((\mathsf{sk}_\Sigma, \perp), (\mathsf{pk}_\Sigma, \mathsf{pk}, Y))$. Whenever a DAPS signature is required, the signature is obtained using the oracle provided by the experiment. If the adversary eventually outputs a forgery $(m^*, \sigma^*) = (m^*, (\sigma_0^*, \sigma_1^*))$ we output (m^*, σ_1^*) as a forgery to win $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_2,\mathsf{DAPS}}(\kappa)$. Clearly, $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathrm{Game}\ 1}(\kappa) \leq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_2,\mathsf{DAPS}}(\kappa)$.

All in all, we now have $\Pr[S_0] = \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{A},\mathsf{DAPS}}(\kappa) \leq \max\{\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_1,\Sigma}(\kappa), \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_2,\mathsf{DAPS}}(\kappa)\} + \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{DAPS},\mathcal{B}}(\kappa)$ which concludes the prove. □

E.2 Proof of Theorem 4

Proof. We prove this theorem using a reduction. Assume that $\mathcal{A}$ breaks DSE-security of Scheme 1. We build a DSE adversary $\mathcal{B}$ against DAPS: When $\mathcal{B}$ is started on the secret key sk and public key pk of DAPS, we compute the key pair of Σ honestly, i.e., $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow \Sigma.\mathsf{KGen}_\Sigma(1^\kappa)$. Then, we compute the combined public key by extending it with $Y \leftarrow \mathsf{sk}_\Sigma \oplus H(\mathsf{sk})$. Now, we start $\mathcal{A}$ on the combined key-pair $(\mathsf{sk}_\Sigma, \mathsf{sk}), (\mathsf{pk}_\Sigma, \mathsf{pk}, Y)$. Once $\mathcal{A}$ returns colliding messages m_1, m_2 and signatures $\sigma_1 = (\sigma_{1,0}, \sigma_{1,1})$, $\sigma_2 = (\sigma_{2,0}, \sigma_{2,1})$, forward the messages with the corresponding DAPS signatures $\sigma_{1,1}, \sigma_{2,1}$ to $\mathcal{B}$. Let $(\mathsf{sk}_\Sigma^*, \mathsf{sk}^*) \leftarrow \mathsf{Ex}_\mathsf{D}((\mathsf{pk}_\Sigma, \mathsf{pk}, Y), m_1, m_2, \sigma_1, \sigma_2)$. Since, by definition, the adversary needs to output $(\mathsf{sk}_\Sigma^*, \mathsf{sk}^*) \neq (\mathsf{sk}_\Sigma, \mathsf{sk})$, it follows that $\mathsf{sk}_\Sigma^* \neq \mathsf{sk}_\Sigma$ or $\mathsf{sk}^* \neq \mathsf{sk}$. If we have $\mathsf{sk}^* = \mathsf{sk}$, we have that $\mathsf{sk}_\Sigma^* = Y \oplus H(\mathsf{sk}) = \mathsf{sk}_\Sigma$ since Y was set up honestly. Hence we have $\mathsf{sk}^* \neq \mathsf{sk}$, so $\mathcal{B}$ wins the DSE-security game if $\mathcal{A}$ wins it, which concludes the proof. □