A preliminary version of this paper appears in the proceedings of NordSec 2018. This is the full version.

The Fiat-Shamir Zoo: Relating the Security of Different Signature Variants

Matilda Backendal[1] Mihir Bellare[2] Jessica Sorrell[3] Jiahao Sun[4]

September, 2018

Abstract

The Fiat-Shamir paradigm encompasses many different ways of turning a given identification scheme into a signature scheme. Security proofs pertain sometimes to one variant, sometimes to another. We systematically study three variants that we call the challenge (signature is challenge and response), commit (signature is commitment and response), and transcript (signature is challenge, commitment and response) variants. Our framework captures the variants via transforms that determine the signature scheme as a function of not only the identification scheme and hash function (to cover both standard and random oracle model hashing), but also what we call a signing algorithm, to cover both classical and with-abort signing. We relate the security of the signature schemes produced by these transforms, giving minimal conditions under which uf-security of one transfers to the other. To apply this comprehensively, we formalize linear identification schemes, show that many schemes in the literature are linear, and show that any linear scheme meets our conditions for the signature schemes given by the three transforms to have equivalent uf-security. Our results give a comprehensive picture of the Fiat-Shamir zoo and allow proofs of security in the literature to be transferred automatically from one variant to another.

[1]Department of Computer Science and Engineering (CSE), University of California San Diego (UCSD), USA, and Faculty of Engineering, Lund University, Sweden. Supported in part by Scott Klemmer and the CSE Undergraduate Summer Research Internship program at UCSD.

[2]Department of Computer Science and Engineering (CSE), University of California San Diego (UCSD), USA. Supported in part by NSF grants CNS-1717640 and CNS-1526801, a gift from Microsoft corporation and ERC Project ERCC (FP7/615074).

[3]Department of Computer Science and Engineering (CSE), University of California San Diego (UCSD), USA. Supported in part by NSF grant CNS-1528068.

[4]Department of Computer Science and Engineering (CSE), University of California San Diego (UCSD), USA. Supported in part by Scott Klemmer and the CSE Undergraduate Summer Research Internship program at UCSD.

Contents

1 Introduction

Ed25519 [12] is a fast signature scheme with widespread usage including in TLS 1.3, SSH, Signal, and Tor [21]. It is derived via the Fiat-Shamir paradigm [16] applied to the Schnorr identification scheme [28]. It is not alone; over the last three decades the Fiat-Shamir paradigm has been a popular way to obtain signature schemes, for reasons including the following: *Speed.* It yields some of our most efficient signature schemes. *Proofs.* The paradigm is backed by proofs of security [27, 1, 20]. *Extendability.* Classically used with number-theoretic schemes [16, 19, 28, 26], extensions of the paradigm now provide lattice-based schemes, some of which are proposed to NIST for post-quantum standards [22, 2, 15, 13].

However, referring, above, to "the" Fiat-Shamir paradigm is misleading, for the paradigm is not monolithic: It encompasses variant methods that, starting from a given identification scheme, yield different signature schemes. This creates some confusion, with proofs in the literature pertaining sometimes to one variant, sometimes to another, yet being quoted without regard to which variant is being considered. Extensions such as signing with aborts [22, 2, 15, 13] bring further variants.

This paper aims to provide a systematic and comprehensive picture of the variants in a general setting, and give results relating their security under minimal assumptions. This allows us to leverage existing security proofs given for one variant [27, 1, 20], automatically transferring them to another, rather than prove security of different variants from scratch.

Background. An identification scheme ID is a 3-move interactive protocol. The prover, having public key pk and secret key sk, sends a commitment $\textsc{Ct}$, the verifier sends a random challenge $\textsc{Ch}$, the prover sends a response $\textsc{Rp}$, and the verifier computes a decision $d \leftarrow \mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp})$ to accept or reject, where 1^λ is the unary representation of the security parameter λ. In a signature scheme based on ID, the prover, now the signer, given message M, computes $\textsc{Ct}$ as before, sets $\textsc{Ch} \leftarrow \mathsf{F}(1^\lambda, pk, (\textsc{Ct}, M))$ to a hash of the commitment and message, computes $\textsc{Rp}$ and then returns a signature σ. We distinguish three variants with regard to what σ consists of. **(1)** In what we call the *transcript* variant [27], σ is $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp})$. It is verified by checking that $\mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp}) = \mathsf{true}$ and $\textsc{Ch} = \mathsf{F}(1^\lambda, pk, (\textsc{Ct}, M))$. **(2)** In what we call the *commitment* variant [25, 1], σ is $(\textsc{Ct}, \textsc{Rp})$. It is verified by setting $\textsc{Ch} \leftarrow \mathsf{F}(1^\lambda, pk, (\textsc{Ct}, M))$ and checking that $\mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp}) = \mathsf{true}$. **(3)** In what we call the *challenge* variant [16, 28, 19, 26], σ is $(\textsc{Ch}, \textsc{Rp})$. This usually yields the shortest signatures but requires a *commitment reproducing algorithm* $\mathsf{ID.CR}$ that allows the verifier to reproduce $\textsc{Ct} \leftarrow \mathsf{ID.CR}(1^\lambda, pk, \textsc{Ch}, \textsc{Rp})$ and then check that $\textsc{Ch} = \mathsf{F}(1^\lambda, pk, (\textsc{Ct}, M))$.

The history of the various transforms is interesting. Fiat and Shamir (FS) [16], GQ [19], Schnorr [28] and Okamoto [26] all gave challenge-style signatures. However, the first security proofs, by Pointcheval and Stern (PS) [27], were for transcript-style signatures, which seem to originate with them. The proofs of Abdalla, An, Bellare and Namprempre (AABN) [1] were for commitment-style signatures, which seem to originate with Ohta and Okamoto (OO) [25]. The changes are (mostly) made silently: PS, OO, AABN (and subsequent literature) tend to refer to their results as establishing security of the FS, GQ, Schnorr and Okamoto schemes, but the proofs pertain to variants not only different from the original ones but in some cases also different from each other.

Questions. We would like a fuller picture, that given an identification scheme ID tells us, for each of the three variant signature schemes derived from ID, whether or not the variant is secure. The above-mentioned results do not directly yield this information. One approach to filling this gap would be to return to the techniques in prior proofs and directly try to prove security of each variant signature scheme. Given the complexity of the techniques, this would be tedious. Instead,

Signature Scheme	**Signature σ**	**To verify σ, check this:**
$\mathsf{DS}_{\mathrm{tr}} = \mathbf{gFS}_{\mathrm{tr}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$	$(\mathrm{Ct}, \mathrm{Ch}, \mathrm{Rp})$	$\mathsf{ID.V}(1^{\lambda}, pk, \mathrm{Ct}, \mathrm{Ch}, \mathrm{Rp}) = \mathsf{true}$ $\mathrm{Ch} = \mathsf{F}^{\mathrm{H}}(1^{\lambda}, pk, (\mathrm{Ct}, M))$
$\mathsf{DS}_{\mathrm{ct}} = \mathbf{gFS}_{\mathrm{ct}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$	$(\mathrm{Ct}, \mathrm{Rp})$	$\mathsf{ID.V}(1^{\lambda}, pk, \mathrm{Ct}, \mathsf{F}^{\mathrm{H}}(1^{\lambda}, pk, (\mathrm{Ct}, M)), \mathrm{Rp}) = \mathsf{true}$
$\mathsf{DS}_{\mathrm{ch}} = \mathbf{gFS}_{\mathrm{ch}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$	$(\mathrm{Ch}, \mathrm{Rp})$	$\mathrm{Ch} = \mathsf{F}^{\mathrm{H}}(1^{\lambda}, pk, (\mathsf{ID.CR}(1^{\lambda}, pk, \mathrm{Ch}, \mathrm{Rp}), M))$

$$\begin{tikzcd}
\mathsf{DS}_{\mathrm{ch}} \arrow[r, shift left, "\text{Prop. 3}", "/" marking] \arrow[r, bend right=40, "\text{CNS, \textbf{Th. 4}}"'] & \mathsf{DS}_{\mathrm{ct}} \arrow[l, shift left, "\text{Prop. 1}", "/" marking] \arrow[l, bend right=40, "\text{SND, \textbf{Th. 2}}"'] \arrow[r, leftrightarrow, "\textbf{Th. 5}"] & \mathsf{DS}_{\mathrm{tr}}
\end{tikzcd}$$

Figure 1: **Top:** Signatures and verification in the signature schemes given by our transforms, where $\mathsf{ID.CR}$ is the commitment reproducing algorithm of ID. Signing of message M (not shown) is done by letting $(\mathrm{Ct}, \mathrm{Ch}, \mathrm{Rp}) \leftarrow\!\$\ \mathsf{S}^{\mathrm{H}}(1^{\lambda}, pk, sk, M)$ and returning the shown σ. **Bottom:** Relations between uf-security of the signature schemes.

we seek *relations between the variants.* This means that for each pair $\mathsf{DS}_{\mathrm{x}}, \mathsf{DS}_{\mathrm{y}}$ of variant signature schemes derived from a given identification scheme ID, we want to determine an assumption or condition $\mathsf{A}_{\mathrm{x,y}}$ on ID under which the security of DS_{x} implies the security of DS_{y}. Then, if we know from prior work that DS_{x} is secure, and can establish that ID satisfies $\mathsf{A}_{\mathrm{x,y}}$, we can conclude that DS_{y} is secure too. This would leverage existing proofs in a modular way. We seek assumptions $\mathsf{A}_{\mathrm{x,y}}$ as weak as possible, both to maximize potential applicability and to understand, theoretically, what are the minimal requirements for a relation to hold.

The literature does contain claims about such relations [2, 20, 17], but (as we will discuss in more detail below) they are mostly informal, specific to particular schemes, or make assumptions we will show to be unnecessarily strong.

Our framework. We capture the variants via transforms that we call general to reflect a broader parameterization than in prior work. A *general Fiat-Shamir transform* **gFS** determines a signature scheme $\mathsf{DS} = \mathbf{gFS}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ based on input parameters an identification scheme ID, a hash function F (allowed access to the random oracle H) and (most novel) a *signing algorithm* S (also allowed access to H). The signing algorithm takes $1^{\lambda}, pk, sk, M$ and returns either $\perp$ or an honest, accepting transcript $(\mathrm{Ct}, \mathrm{Ch}, \mathrm{Rp})$ satisfying $\mathrm{Ch} = \mathsf{F}^{\mathrm{H}}(1^{\lambda}, pk, (\mathrm{Ct}, M))$. But, beyond requiring this condition, we *do not prescribe how the signing algorithm operates.* To sign message M, run $T \leftarrow\!\$\ \mathsf{S}^{\mathrm{H}}(1^{\lambda}, pk, sk, M)$, and return $\perp$ as signature if $T = \perp$. Otherwise, parse T as $(\mathrm{Ct}, \mathrm{Ch}, \mathrm{Rp})$. Exactly what is returned as the signature σ, and how that signature is verified, depends on the transform. This is summarized for each of our three transforms $\mathbf{gFS}_{\mathrm{tr}}, \mathbf{gFS}_{\mathrm{ct}}, \mathbf{gFS}_{\mathrm{ch}}$ in Figure 1, reflecting the three variants discussed above. The schemes are shown in full in Figure 4. As we will see, the broad parameterization enhances applicability because our relations will hold for all choices of F, S.

Relations between security of signature schemes. The security attribute we consider for the signature schemes, hereafter called uf-security, is the usual unforgeability under chosen message attack [18] extended, due to growing recognition of its importance, to the multi-user setting [4, 20]. Now, given $\mathsf{ID}, \mathsf{F}, \mathsf{S}$, consider the three signature schemes $\mathsf{DS}_{\mathrm{x}} = \mathbf{gFS}_{\mathrm{x}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ for $\mathrm{x} \in \{\mathrm{tr}, \mathrm{ct}, \mathrm{ch}\}$. We seek relations between their uf-security, as discussed above. This means that for each (distinct) pair $\mathrm{x}, \mathrm{y} \in \{\mathrm{tr}, \mathrm{ct}, \mathrm{ch}\}$ we ask under what assumption $\mathsf{A}_{\mathrm{x,y}}$ the uf-security of DS_{x} implies the uf-security of DS_{y}.

Our results are summarized by the picture at the bottom of Figure 1. That $\mathsf{DS}_{\mathrm{tr}}$ and $\mathsf{DS}_{\mathrm{ct}}$ have equivalent uf-security is trivial. The interesting question is, does uf-security of one of $\mathsf{DS}_{\mathrm{ct}}, \mathsf{DS}_{\mathrm{ch}}$ imply uf-security of the other? The straight, barred arrows say that in general (that is, without any condition beyond basic completeness on the commitment reproducing algorithm) the answer is no. The curved, un-barred arrows say the answer is yes, under conditions on the commitment reproducing algorithm (formally, on the overlying identification scheme ID that includes this algorithm) that we give. Specifically, Theorem 2 says that if ID has a property we define and call soundness (SND) then, if $\mathsf{DS}_{\mathrm{ct}}$ is uf-secure, so is $\mathsf{DS}_{\mathrm{ch}}$. Theorem 4 says that if ID has a property we define and call consistency (CNS) then, if $\mathsf{DS}_{\mathrm{ch}}$ is uf-secure, so is $\mathsf{DS}_{\mathrm{ct}}$. SND-security asks that it be computationally hard to find a challenge and response such that the commitment reproducing algorithm succeeds in returning a commitment but the resulting transcript is not accepting. CNS-security asks that it be computationally hard to create an accepting transcript in which the commitment is different from the one given by the commitment reproducing algorithm. The reductions underlying all our positive results are tight.

Breadth of applicability. The positive relations (un-barred arrows in Figure 1) hold for *all choices of hash function* F *and signing algorithm* S. This broadens applicability. With regard to hashing, it means we can transfer security in both the random oracle and the standard models: For $\mathrm{x}, \mathrm{y} \in \{\mathrm{tr}, \mathrm{ct}, \mathrm{ch}\}$, if DS_{x} provides uf-security with a random-oracle hash function then (assuming of course, as necessary, properties of ID as above) so does DS_{y}, but if DS_{x} provides uf-security with hash function SHA256, then so does DS_{y}. With regard to signing, this means that our framework captures both canonical and more modern variants of the Fiat-Shamir paradigm. For example, in the literature Fiat-Shamir with aborts [22, 2, 15, 13] is viewed as an extension of the canonical Fiat-Shamir paradigm. In our framework, the canonical and with-abort variants correspond simply to different choices of signing algorithm S (cf. Figure 4), so our results apply automatically to both.

We elaborate on the second point. We said above how the Fiat-Shamir paradigm prescribes signing a message M, which we now call the canonical way: generate $\mathrm{C}{\scriptstyle\mathrm{T}}$ as would the honest prover, set $\mathrm{C}{\scriptstyle\mathrm{H}} \leftarrow \mathsf{F}^{\mathrm{H}}(1^{\lambda}, pk, (\mathrm{C}{\scriptstyle\mathrm{T}}, M))$, generate $\mathrm{R}{\scriptstyle\mathrm{P}}$ as would the prover, then return σ computed from $\mathrm{C}{\scriptstyle\mathrm{T}}, \mathrm{C}{\scriptstyle\mathrm{H}}, \mathrm{R}{\scriptstyle\mathrm{P}}$ according to the variant (challenge, commit or transcript) of interest. This is captured for us by setting S to the canonical algorithm on the bottom left of Figure 4. This works (yields a correct signature) if the identification scheme has perfect correctness. However, in the identification schemes from lattices [22, 2, 15, 13], the response can be $\perp$ with constant probability. So the process is modified to repeat picking $\mathrm{C}{\scriptstyle\mathrm{T}}, \mathrm{C}{\scriptstyle\mathrm{H}}, \mathrm{R}{\scriptstyle\mathrm{P}}$ as above until the conversation is accepting or some time bound is exceeded, which is called signing with aborts. (In this case, the signature schemes have imperfect correctness, returning $\perp$ with negligible probability.) The challenge, commit and transcript variants for the signature schemes exist here too, so the question of how their security relates arises again. We do not need to address this separately. It is captured for us, and addressed by the results noted above, simply by setting S to the algorithm on the bottom right of Figure 4. Choices of S beyond these two are possible as well, for potential further applications.

Perfect uniqueness. We have introduced the SND and CNS conditions on commitment reproducible identification schemes, showing that they suffice for transfer of uf-security between the signature variants. (SND allows the uf-security of $\mathsf{DS}_{\mathrm{ct}}$ to imply that of $\mathsf{DS}_{\mathrm{ch}}$, and CNS the converse.) We also define a third condition called perfect uniqueness (P-UNIQ). It asks that a transcript $\textsc{Ct}, \textsc{Ch}, \textsc{Rp}$ be accepting if and only if the commitment reproducing algorithm $\mathsf{ID.CR}$ returns exactly $\textsc{Ct}$ on inputs $\textsc{Ch}, \textsc{Rp}$. Figure 8 says that P-UNIQ implies both SND and CNS. Establishing P-UNIQ-ness of a commitment reproducible identification scheme ID is thus a simple path (and one we will often be able to use) to showing that all the signature variants derived from ID have equivalent uf-cma security. However, Figure 8 also says that P-UNIQ is a strictly stronger condition than SND or CNS. So for some commitment reproducible identification schemes, P-UNIQ may fail to be true, yet we might be able to directly establish SND and CNS to show equivalence of uf-security of the signature variants.

Linear identification schemes. We'd like to know whether identification schemes in the literature meet our conditions (P-UNIQ, or SND, CNS as necessary). However, there are many schemes, and new ones keep appearing, and testing them individually is tedious. Instead, in Section 5, we formalize *linear* identification schemes. Proposition 6 says that any linear identification scheme is (unconditionally) P-UNIQ. Our results thus say that the three variant signature schemes emanating from any linear identification scheme have equivalent uf-security.

We then show that classical identification schemes like FS [16], Sch [28], GQ [19] and Ok [26] are linear. We also show that the Ly lattice based identification scheme of [22] is linear. Since proofs of uf-security exist for at least one signature variant for all these identification schemes, we can conclude that all three variants are uf-secure.

Lyubashevsky [23] directly gives a lattice-based signature scheme that he does not derive via the FS paradigm. (Indeed the paper presents no identification scheme.) We show how to capture it in our framework as $\mathbf{gFS}_{\mathrm{ch}}[\mathsf{ID}, \mathsf{F}, \mathsf{SA}_{\mathsf{ID},\mathsf{F},t}]$ where SA_t is the abort-based signing algorithm on the bottom right of Figure 4 and ID is an identification scheme that we define and show is linear. This means we can define the other variant signature schemes and transfer the proofs of [23] to them.

As the above indicates, the concept of linear identification schemes serves also to unify the literature, showing that what look like different schemes are in fact instances of one underlying scheme. We see this as something that was understood but not, until now, formalized.

Which variant should one use? Our work is about relating the security of the different signature variants. The question of which variant to prefer in usage is orthogonal, and the answer differs from case to case. We discuss the choices briefly. The challenge variant $\mathbf{gFS}_{\mathrm{ch}}$ usually yields the shortest signatures (examples where this is true are FS [16], GQ [19], Sch in the group of integers modulo a prime [28] and Ly [22]) but requires that ID be commitment reproducible (meaning, there exists a commitment reproducing algorithm $\mathsf{ID.CR}$) which is not always true. When ID is not commitment reproducible, one can use $\mathbf{gFS}_{\mathrm{ct}}$. Here, in some cases (like Sch over elliptic curve groups) the signature size stays as small as with $\mathbf{gFS}_{\mathrm{ch}}$, but in other cases, it might grow. The transcript variant $\mathbf{gFS}_{\mathrm{tr}}$ is also an option for usage when commitment reproducibility is lacking, but there seems no practical reason for this, because signatures are always shorter with $\mathbf{gFS}_{\mathrm{ct}}$. We consider $\mathbf{gFS}_{\mathrm{tr}}$ in this paper because it was the variant for which the seminal work of Pointcheval and Stern [27] gave proofs.

Of course performance (including signature size) is just one criterion with regard to a choice for usage. Another is security proofs. The general results in the literature give proofs for $\mathbf{gFS}_{\mathrm{ct}}$ [1] and $\mathbf{gFS}_{\mathrm{tr}}$ [27], not $\mathbf{gFS}_{\mathrm{ch}}$. Our framework and results can be used to transfer them to the (usually more efficient) $\mathbf{gFS}_{\mathrm{ch}}$.

Related work. Kiltz, Masny, and Pan [20] briefly note that $\mathsf{DS}_{\mathrm{ch}}, \mathsf{DS}_{\mathrm{ct}}$ are equivalent in terms of uf-security assuming the verification algorithm has a certain property. This seems to be equivalent to the identification scheme being P-UNIQ. Figure 8 shows that the SND and CNS properties that allow us to establish the same equivalence are implied by, and strictly weaker than, P-UNIQ, making our results stronger. Also their results are for the canonical signing algorithm, while ours are for an arbitrary one. Abdalla, Fouque, Lyubashevsky, and Tibouchi [2] give results for commitment-style signatures with aborts, saying that these transfer to the challenge style for their schemes because "the commitment is uniquely determined by the challenge and response." The phrase in quotes is not too precise but the intent is likely P-UNIQ. Galbraith, Petit, and Silva [17] show that, for their particular scheme, under weak conditions on commitment reproducibility, security of the commit version implies security of a version that is like the challenge one except that signature verification additionally checks that the verifier accepts the transcript. This is added verification cost compared to the classical Fiat-Shamir style challenge variant, which is the version we consider and which does not have such a check.

We view our work as unifying, systematizing and formalizing long-standing understanding, scattered observations and folklore. Nothing in this paper is very novel or technically difficult. Our hope is that it fills some gaps and can be a point of reference for variants of Fiat-Shamir signatures.

Extensions. Beyond basic (uf-cma) signature schemes, identification schemes have been used to build identity-based signatures [7], blind signatures [14, 27] leakage-resilient signatures [3, 5], double authentication preventing signatures [9] and beyond. Returning to the basic setting, variants of the Fiat-Shamir paradigm offering better concrete security have been considered [8]. In all these places and settings, the commit, challenge and transcript variants arise. One can ask how their security relates, and extend our framework and results to answer this question.

2 Basic definitions

Notation. We let ε denote the empty string. If Z is a string then $|Z|$ denotes its length. If X is a finite set, we let $x \leftarrow\$ X$ denote picking an element of X uniformly at random and assigning it to x, and we let $|X|$ denote the size of X. We use $\perp$ (bot) as a special symbol to denote rejection, and it is assumed to not be in $\{0,1\}^*$. Both inputs and outputs to algorithms can be $\perp$. We adopt the convention that if any input to an algorithm is $\perp$, then its output is $\perp$ as well. By $\lambda \in \mathbb{N}$ we denote the security parameter, and by 1^λ its unary representation. Recall that a function $\nu\colon \mathbb{N} \to \mathbb{R}$ is negligible if for every positive polynomial $p\colon \mathbb{N} \to \mathbb{R}$ there is a $\lambda_p \in \mathbb{N}$ such that $\nu(\lambda) \leq 1/p(\lambda)$ for all $\lambda \geq \lambda_p$.

Algorithms may be randomized unless otherwise indicated. Running time is worst case. "PT" stands for "polynomial time," whether for a randomized algorithm or a deterministic one. If A is an algorithm, we let $y \leftarrow A^{\mathrm{O}_1,\cdots}(x_1,\ldots;\omega)$ denote running A on inputs $x_1,\ldots$ and coins ω, with oracle access to $\mathrm{O}_1,\ldots$, and assigning the output to y. By $y \leftarrow\$ A^{\mathrm{O}_1,\cdots}(x_1,\ldots)$ we denote picking ω at random and letting $y \leftarrow A^{\mathrm{O}_1,\cdots}(x_1,\ldots;\omega)$. We let $[A^{\mathrm{O}_1,\cdots}(x_1,\ldots)]$ denote the set of all possible outputs of A when run on inputs $x_1,\ldots$ and with oracle access to $\mathrm{O}_1,\ldots$. An adversary is an algorithm.

We use the code-based game-playing framework of [11]. (See Figure 5 for an example.) By $\Pr[\mathbf{G}]$ we denote the probability that the execution of game $\mathbf{G}$ results in the game returning true. We adopt the convention that the running time of an adversary executing with some game refers to the worst case execution time of the game with the adversary, meaning the time taken for oracles to compute replies to queries is included. The random oracle (RO) model [10] is captured by inclusion in the game of a procedure H that implements a variable output length RO. See for example Figure 3.

Identification schemes. An identification scheme ID (called a canonical identification scheme in [1]) specifies several algorithms and associated quantities, as follows. In an initialization phase, via $(pk, sk) \leftarrow\$ \mathsf{ID.Kg}(1^\lambda)$, the prover runs the key-generation algorithm $\mathsf{ID.Kg}$ on input the unary

Prover		**Verifier**
Input: pk, sk		Input: pk
$(\textsc{Ct}, \textsc{St}) \leftarrow\!\!\$\ \mathsf{ID.Ct}(1^\lambda, pk)$	$\xrightarrow{\textsc{Ct}}$	
	$\xleftarrow{\textsc{Ch}}$	$\textsc{Ch} \leftarrow\!\!\$\ \mathsf{ID.ChS}(\lambda)$
$\textsc{Rp} \leftarrow \mathsf{ID.Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St})$	$\xrightarrow{\textsc{Rp}}$	$d \leftarrow \mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp})$

Figure 2: Operation of an identification scheme ID.

representation 1^λ of the security parameter λ to obtain a public key pk and a private key sk, both of which she stores. It is assumed that the verifier is in possession of pk. (In practice this is likely done via certificates, but that is not in the scope of the identification scheme.) Identification then operates as depicted in Figure 2. Via $(\textsc{Ct}, \textsc{St}) \leftarrow\!\!\$\ \mathsf{ID.Ct}(1^\lambda, pk)$, the prover generates a *commitment* $\textsc{Ct}$ and corresponding private state $\textsc{St}$. The verifier sends a challenge $\textsc{Ch} \leftarrow\!\!\$\ \mathsf{ID.ChS}(\lambda)$ drawn at random from the *challenge space* $\mathsf{ID.ChS}(\lambda) = \{0,1\}^{\mathsf{ID.ChL}(\lambda)}$ where $\mathsf{ID.ChL}\colon \mathbb{N} \to \mathbb{N}$ is the *challenge length* function associated to ID. The prover's *response* $\textsc{Rp} \leftarrow \mathsf{ID.Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St})$ is computed via a deterministic algorithm $\mathsf{ID.Rp}$. The verifier's *decision* $d \leftarrow \mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp})$, which is either $\mathsf{true}, \mathsf{false}$ or $\perp$, is also computed deterministically. Algorithms $\mathsf{ID.Kg}$, $\mathsf{ID.Ct}$, $\mathsf{ID.Rp}$, $\mathsf{ID.V}$ are required to be PT.

The *honest-transcript generating function* $\mathbf{HTR}_{\mathsf{ID},\lambda}$ associated to ID and $\lambda \in \mathbb{N}$ takes input $(pk, sk) \in [\mathsf{ID.Kg}(1^\lambda)]$, and returns a transcript of a conversation between the honest prover and the verifier, as follows:

```
$\underline{\mathbf{HTR}_{\mathsf{ID},\lambda}(pk, sk)}$
$(\textsc{Ct}, \textsc{St}) \leftarrow\!\!\$\ \mathsf{ID.Ct}(1^\lambda, pk)$ ; $\textsc{Ch} \leftarrow\!\!\$\ \mathsf{ID.ChS}(\lambda)$ ; $\textsc{Rp} \leftarrow \mathsf{ID.Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St})$
Return $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp})$
```

For $\lambda \in \mathbb{N}$ and $(pk, sk) \in [\mathsf{ID.Kg}(1^\lambda)]$, we define the *set of accepting transcripts*

$$\mathbf{ACC}_{\mathsf{ID},\lambda}(pk) = \{\ (\textsc{Ct}, \textsc{Ch}, \textsc{Rp})\ :\ \mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp}) = \mathsf{true}\ \}\ .$$

Correctness, for most schemes, is simple, saying that honest transcripts are always accepting: formally, for all $\lambda \in \mathbb{N}$ and all $(pk, sk) \in [\mathsf{ID.Kg}(1^\lambda)]$ we have $[\mathbf{HTR}_{\mathsf{ID},\lambda}(pk, sk)] \subseteq \mathbf{ACC}_{\mathsf{ID},\lambda}(pk)$. We call this perfect correctness. However we will need to also consider a relaxation where there is a correctness error, and this has to be carefully formulated. We say that ID has correctness error $\nu\colon \mathbb{N} \to \mathbb{R}$ if for all $\lambda \in \mathbb{N}$ and all $(pk, sk) \in [\mathsf{ID.Kg}(1^\lambda)]$ we have $\Pr[(\textsc{Ct}, \textsc{Ch}, \textsc{Rp}) \notin \mathbf{ACC}_{\mathsf{ID},\lambda}(pk)] \leq \nu(\lambda)$, where the probability is over $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp}) \leftarrow\!\!\$\ \mathbf{HTR}_{\mathsf{ID},\lambda}(pk, sk)$. This captures the requirement that the verifier accepts with probability at least $1 - \nu(\lambda)$ in an interaction with the honest prover. Some commonly occurring choices for ν are a constant, like $\nu(\cdot) = 1/2$, or a negligible function, and in the latter case we say that ID has negligible correctness error.

Signature Schemes. A (digital) signature scheme DS specifies several algorithms and associated quantities, as follows. In an initialization phase, via $(pk, sk) \leftarrow\!\!\$\ \mathsf{DS.Kg}(1^\lambda)$, the signer runs the PT key-generation algorithm $\mathsf{DS.Kg}$ on input 1^λ to obtain a public key pk and a private key sk, both of which she stores. It is assumed that the verifier is in possession of pk. (As with identification, how this happens is not in the scope of the signature scheme.) Via $\sigma \leftarrow\!\!\$\ \mathsf{DS.Sign}^{\mathrm{H}}(1^\lambda, pk, sk, M)$, the signer generates a signature σ of a message $M \in \{0,1\}^*$. Via $d \leftarrow \mathsf{DS.V}^{\mathrm{H}}(1^\lambda, pk, M, \sigma)$, a verifier can deterministically obtain a decision regarding whether σ is a valid signature of M under pk. The signing and verifying algorithms have oracle access to the random oracle H and are required to be PT. We say that DS has correctness error $\nu\colon \mathbb{N} \to \mathbb{R}$ if, for all $\lambda \in \mathbb{N}$, all $(pk, sk) \in [\mathsf{DS.Kg}(1^\lambda)]$ and all $M \in \{0,1\}^*$ we have $\Pr[\mathsf{DS.V}^{\mathrm{H}}(1^\lambda, pk, M, \mathsf{DS.Sign}^{\mathrm{H}}(1^\lambda, pk, sk, M)) \neq \mathsf{true}] \leq \nu(\lambda)$, where

```
Game $\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS},\mathcal{A}}(\lambda)$
$n \leftarrow 0$ ; $S \leftarrow \emptyset$
$(M, \sigma, i) \leftarrow_{\$} \mathcal{A}^{\mathrm{New},\mathrm{Sign},\mathrm{H}}(1^\lambda)$
$d \leftarrow \mathsf{DS.V}^{\mathrm{H}}(1^\lambda, pk_i, M, \sigma)$
Return $(d = \mathsf{true}) \wedge ((i, M) \notin S)$

$\mathrm{H}(W, \ell)$
If $\mathrm{HT}[W, \ell] = \perp$ then $\mathrm{HT}[W, \ell] \leftarrow_{\$} \{0,1\}^\ell$
Return $\mathrm{HT}[W, \ell]$

$\mathrm{Sign}(i, M)$
$\sigma \leftarrow_{\$} \mathsf{DS.Sign}^{\mathrm{H}}(1^\lambda, pk_i, sk_i, M)$
$S \leftarrow S \cup \{(i, M)\}$
Return $\sigma$

New()
$n \leftarrow n + 1$
$(pk_n, sk_n) \leftarrow_{\$} \mathsf{DS.Kg}(1^\lambda)$
Return $pk_n$
```

Figure 3: Game for UF-CMA security of digital signature schemes in the multi-user setting.

the probability is over the random choices of H and the coins of $\mathsf{DS.Sign}$. We say correctness is perfect if $\nu(\cdot) = 0$, the usual requirement, but imperfect correctness will be important in some of our applications.

Our security metric for signatures, called uf-security, is the usual unforgeability under chosen-message attack [18], but in the multi-user setting, due to increasing recognition of the importance of the latter [4, 20]. For the formalization, consider game $\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS},\mathcal{A}}(\lambda)$ in Figure 3 associated to signature scheme DS and adversary $\mathcal{A}$. By calling the New oracle, the adversary can initialize a new user (signer), obtaining her public key. The number of users n, being the number of queries to New, is thus under the adversary's control. Via the Sign oracle, the adversary can mount its chosen-message attack, obtaining a signature on a message of its choice under a user of its choice. The adversary eventually outputs a pointer $i \in \{1, \ldots, n\}$ to a user, a message M, and a claimed signature of M under pk_i, winning if the signature is valid and non-trivial. We let $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS},\mathcal{A}}(\lambda) = \Pr[\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS},\mathcal{A}}(\lambda)]$ be the probability that the game returns true. We say that DS is uf-secure if the function $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS},\mathcal{A}}(\cdot)$ is negligible for all PT adversaries $\mathcal{A}$.

3 Transforms and signature relations

The FS transforms are usually viewed as turning an identification scheme into a signature scheme in the random oracle model. Our general transforms take not only an identification scheme, but a hash function F, so that both standard model and random oracle model hash functions are covered. More novel, they take a description S of a signing process, to cover the fact that FS has been used in settings with and without abort. We begin with commitment reproducibility, needed for some of the transforms, then discuss the other parameters, and then specify the transforms. We then define the SND and CNS security notions for commitment reproducible identification schemes that allow us to relate the security of the schemes emanating from the different general transforms. In Section 4 we study relations between different security notions for commitment reproducible identification schemes.

Commitment reproducibility. A *commitment reproducing algorithm* for identification scheme ID is a deterministic, PT algorithm $\mathsf{ID.CR}$ that returns an output in $\{0,1\}^* \cup \{\perp\}$. We require the following completeness condition: for all $\lambda \in \mathbb{N}$, all $(pk, sk) \in [\mathsf{ID.Kg}(1^\lambda)]$ and all $(\mathrm{Ct}, \mathrm{Ch}, \mathrm{Rp}) \in [\mathbf{HTR}_{\mathsf{ID},\lambda}(pk, sk)] \cap \mathbf{ACC}_{\mathsf{ID},\lambda}(pk)$ we have $\mathrm{Ct} = \mathsf{ID.CR}(1^\lambda, pk, \mathrm{Ch}, \mathrm{Rp})$. Completeness says that the commitment in an accepting transcript of an interaction between the honest prover and the verifier is uniquely determined by the challenge and response, and moreover can be computed from them in PT by the commitment reproducing algorithm. An identification scheme ID is *commitment reproducible* if it specifies (in addition to the quantities it already specifies as per Section 2) a

```
$\mathsf{DS}_{\mathrm{tr}}.\mathsf{Sign}^{\mathrm{H}}(1^\lambda, pk, sk, M)$
  $T \leftarrow\$ \mathsf{S}^{\mathrm{H}}(1^\lambda, pk, sk, M)$
  If $(T = \bot)$ then return $\bot$
  $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp}) \leftarrow T$
  $\sigma \leftarrow (\textsc{Ct}, \textsc{Ch}, \textsc{Rp})$ ; Return $\sigma$

$\mathsf{DS}_{\mathrm{tr}}.\mathsf{V}^{\mathrm{H}}(1^\lambda, pk, M, \sigma)$
  $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp}) \leftarrow \sigma$
  $d_0 \leftarrow \mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp})$
  $d_1 \leftarrow (\textsc{Ch} = \mathsf{F}^{\mathrm{H}}(1^\lambda, pk, (\textsc{Ct}, M)))$
  Return $(d_0 \wedge d_1)$
```

```
$\mathsf{DS}_{\mathrm{ct}}.\mathsf{Sign}^{\mathrm{H}}(1^\lambda, pk, sk, M)$
  $T \leftarrow\$ \mathsf{S}^{\mathrm{H}}(1^\lambda, pk, sk, M)$
  If $(T = \bot)$ then return $\bot$
  $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp}) \leftarrow T$
  $\sigma \leftarrow (\textsc{Ct}, \textsc{Rp})$ ; Return $\sigma$

$\mathsf{DS}_{\mathrm{ct}}.\mathsf{V}^{\mathrm{H}}(1^\lambda, pk, M, \sigma)$
  $(\textsc{Ct}, \textsc{Rp}) \leftarrow \sigma$
  $\textsc{Ch} \leftarrow \mathsf{F}^{\mathrm{H}}(1^\lambda, pk, (\textsc{Ct}, M))$
  Return $\mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp})$
```

```
$\mathsf{DS}_{\mathrm{ch}}.\mathsf{Sign}^{\mathrm{H}}(1^\lambda, pk, sk, M)$
  $T \leftarrow\$ \mathsf{S}^{\mathrm{H}}(1^\lambda, pk, sk, M)$
  If $(T = \bot)$ then return $\bot$
  $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp}) \leftarrow T$
  $\sigma \leftarrow (\textsc{Ch}, \textsc{Rp})$ ; Return $\sigma$

$\mathsf{DS}_{\mathrm{ch}}.\mathsf{V}^{\mathrm{H}}(1^\lambda, pk, M, \sigma)$
  $(\textsc{Ch}, \textsc{Rp}) \leftarrow \sigma$
  $\textsc{Ct} \leftarrow \mathsf{ID.CR}(1^\lambda, pk, \textsc{Ch}, \textsc{Rp})$
  If $(\textsc{Ct} = \bot)$ then return false
  Return $(\textsc{Ch} = \mathsf{F}^{\mathrm{H}}(1^\lambda, pk, (\textsc{Ct}, M)))$
```

```
Algorithm $\mathsf{SC}^{\mathrm{H}}_{\mathsf{ID},\mathsf{F}}(1^\lambda, pk, sk, M)$
  $(\textsc{Ct}, \textsc{St}) \leftarrow\$ \mathsf{ID.Ct}(1^\lambda, pk)$
  $\textsc{Ch} \leftarrow \mathsf{F}^{\mathrm{H}}(1^\lambda, pk, (\textsc{Ct}, M))$
  $\textsc{Rp} \leftarrow \mathsf{ID.Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St})$
  Return $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp})$

Algorithm $\mathsf{SA}^{\mathrm{H}}_{\mathsf{ID},\mathsf{F},t}(1^\lambda, pk, sk, M)$
  $d \leftarrow$ false ; $i \leftarrow 0$
  While $(d =$ false and $i < t(\lambda))$ do:
    $i \leftarrow i + 1$
    $(\textsc{Ct}, \textsc{St}) \leftarrow\$ \mathsf{ID.Ct}(1^\lambda, pk)$
    $\textsc{Ch} \leftarrow \mathsf{F}^{\mathrm{H}}(1^\lambda, pk, (\textsc{Ct}, M))$
    $\textsc{Rp} \leftarrow \mathsf{ID.Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St})$
    $d \leftarrow \mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp})$
  If $(d =$ true$)$ then return $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp})$
  Else return $\bot$
```

Figure 4: Top three panels show signing and verifying algorithms of the signature schemes $\mathsf{DS}_{\mathrm{tr}}$, $\mathsf{DS}_{\mathrm{ct}}$ and $\mathsf{DS}_{\mathrm{ch}}$ obtained by applying the $\mathbf{gFS}_{\mathrm{tr}}, \mathbf{gFS}_{\mathrm{ct}}$ and $\mathbf{gFS}_{\mathrm{ch}}$ transforms, respectively, to identification scheme ID, hash function F and signing algorithm S. Bottom panel shows examples of signing algorithms.

commitment reproducing algorithm $\mathsf{ID.CR}$ that satisfies the completeness condition.

Commitment reproducibility is enough to define the $\mathbf{gFS}_{\mathrm{ch}}$ transform. But note that the condition we have put so far on $\mathsf{ID.CR}$ (completeness) says nothing about dishonest transcripts, meaning ones created in interactions between a cheating prover and the verifier. To establish relations between the uf-security of the signature schemes, we will require that ID has further attributes (SND, CNS, to be defined) related to such dishonest transcripts.

Hashing. The **gFS** transforms use a hash function. Most of our results hold regardless of the choice of the hash function, in particular both when it is a standard-model function and when it is a random oracle. To capture this formally, we define a hash function as a deterministic algorithm F that may have access to a random oracle H. It is *compatible* with identification scheme ID if $\mathsf{F}^{\mathrm{H}}(1^\lambda, pk, x) \in \mathsf{ID.ChS}(\lambda)$ for all $\lambda \in \mathbb{N}$, all $(pk, sk) \in [\mathsf{ID.Kg}(1^\lambda)]$, all x and all H. In our usage, $x = (\textsc{Ct}, M)$ will consist of a commitment and message. By setting $\mathsf{F}^{\mathrm{H}}(1^\lambda, pk, x) = \mathrm{H}((1^\lambda, pk, x), \ell(\lambda))$ for some $\ell\colon \mathbb{N} \to \mathbb{N}$ we can cover the case where the hash function is a random oracle, but we can also, for example, set $\mathsf{F}^{\mathrm{H}}(1^\lambda, pk, x) = \mathrm{SHA256}((1^\lambda, pk, x))$ to cover schemes where the hash function has been instantiated via SHA256.

Signing. Let ID be an identification scheme, and F a hash function compatible with it. A *signing algorithm* compatible with ID and F is a PT algorithm S that operates as $T \leftarrow_\$ \mathsf{S}^{\mathrm{H}}(1^\lambda, pk, sk, M)$. We require that if $T \neq \bot$ then it parses as $(\mathrm{Ct}, \mathrm{Ch}, \mathrm{Rp}) \leftarrow T$ satisfying $\mathrm{Ch} = \mathsf{F}^{\mathrm{H}}(1^\lambda, pk, (\mathrm{Ct}, M))$ and $(\mathrm{Ct}, \mathrm{Ch}, \mathrm{Rp}) \in [\mathbf{HTR}_{\mathsf{ID},\lambda}(pk, sk)] \cap \mathbf{ACC}_{\mathsf{ID},\lambda}(pk)$. That is, a non-$\bot$ signature is an honest, accepting transcript in which the challenge is the hash of the commitment and message. We say that S has signing error $\nu\colon \mathbb{N} \to \mathbb{R}$ if $\Pr[\mathsf{S}^{\mathrm{H}}(1^\lambda, pk, sk, M) = \bot] \leq \nu(\lambda)$ for all $\lambda \in \mathbb{N}$, all $(pk, sk) \in [\mathsf{ID.Kg}(1^\lambda)]$ and all $M \in \{0,1\}^*$, where the probability is over the coins of S and H.

The bottom panel of Figure 4 shows some choices of signing algorithms. On the left is the canonical signing algorithm $\mathsf{SC}_{\mathsf{ID},\mathsf{F}}$. This is the classical choice, representing the usual, prescribed way to generate FS signatures. When ID has perfect correctness, $\mathsf{SC}_{\mathsf{ID},\mathsf{F}}$ has zero signing error. On the right is a signing with aborts algorithm $\mathsf{SA}_{\mathsf{ID},\mathsf{F},t}$ as per [22], where $t\colon \mathbb{N} \to \mathbb{N}$ is a polynomial. This may be used when ID has imperfect correctness. It tries to generate an honest, accepting transcript, returning $\bot$ if it fails after $t(\cdot)$ attempts. If ID has correctness error a (non-zero) constant $\nu(\cdot) = \varepsilon < 1$, then setting $t(\lambda)$, to, say, $\lceil \log^2(\lambda) \cdot \log(1/\varepsilon) \rceil$ will result in $\mathsf{SA}_{\mathsf{ID},\mathsf{F},t}$ having negligible signing error in the case that F is a random oracle. For other choices of F, the correctness error of $\mathsf{SA}_{\mathsf{ID},\mathsf{F},t}$ would have to be evaluated directly (this seems to be somewhat glossed over in prior work) but for practical choices of F we expect it to still be about ν by the random oracle paradigm [10]. Our transforms will not pin down a particular way of generating signatures, but rather allow that to be specified by a signing algorithm S that they take as input. This allows our results to cover many different types of signing.

The **gFS** transforms. Let ID be an identification scheme, F a hash function compatible with it, and S a signing algorithm compatible with both. The $\mathbf{gFS}_{\mathrm{tr}}$ transform associates to $\mathsf{ID}, \mathsf{F}, \mathsf{S}$ the signature scheme $\mathsf{DS}_{\mathrm{tr}} = \mathbf{gFS}_{\mathrm{tr}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ whose algorithms are specified in the first panel in Figure 4. The $\mathbf{gFS}_{\mathrm{ct}}$ transform associates to $\mathsf{ID}, \mathsf{F}, \mathsf{S}$ the signature scheme $\mathsf{DS}_{\mathrm{ct}} = \mathbf{gFS}_{\mathrm{ct}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ whose algorithms are specified in the second panel in Figure 4. Assuming additionally that ID is commitment reproducible, and letting $\mathsf{ID.CR}$ be its commitment reproducing algorithm, the $\mathbf{gFS}_{\mathrm{ch}}$ transform associates to $\mathsf{ID}, \mathsf{F}, \mathsf{S}$ the signature scheme $\mathsf{DS}_{\mathrm{ch}} = \mathbf{gFS}_{\mathrm{ch}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ whose algorithms are specified in the third panel of Figure 4. Although this is not explicitly indicated in the code, note that in all cases, as per our general conventions, the signature verification algorithm returns $\bot$ if its input signature σ is $\bot$. The correctness error of a signature scheme $\mathsf{DS} = \mathbf{gFS}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ given by one of our transforms is just the signing error of the signing algorithm S. So, for example, if ID has perfect correctness and $\mathsf{S} = \mathsf{SC}_{\mathsf{ID},\mathsf{F}}$, then DS has perfect correctness.

Attributes of the commitment reproducing algorithm. Security of the different variants of the FS transform will rely on different properties of commitment reproducible identification schemes that we now introduce. Figure 8 shows the relations between the notions we define here. In the following let ID be a commitment reproducible identification scheme.

The strongest attribute is what we call *Perfect Uniqueness* (P-UNIQ). It asks that for all $\lambda \in \mathbb{N}$, all $(pk, sk) \in [\mathsf{ID.Kg}(1^\lambda)]$, all $\mathrm{Ch} \in \mathsf{ID.ChS}(\lambda)$ and all Ct, Rp that are not $\bot$ we have: $\mathsf{ID.V}(1^\lambda, pk, \mathrm{Ct}, \mathrm{Ch}, \mathrm{Rp}) = \mathsf{true}$ if and only if $\mathrm{Ct} = \mathsf{ID.CR}(1^\lambda, pk, \mathrm{Ch}, \mathrm{Rp})$. Figure 8 says the SND, CNS attributes we define next are implied by P-UNIQ, but strictly weaker than it.

We now introduce *soundness*. To understand what it means, we start with *Perfect Soundness* (P-SND). This asks that for all $\lambda \in \mathbb{N}$, all $(pk, sk) \in [\mathsf{ID.Kg}(1^\lambda)]$, all $\mathrm{Ch} \in \mathsf{ID.ChS}(\lambda)$ and all Rp we have: If $\mathrm{Ct} \leftarrow \mathsf{ID.CR}(1^\lambda, pk, \mathrm{Ch}, \mathrm{Rp})$ is not $\bot$ then $\mathsf{ID.V}(1^\lambda, pk, \mathrm{Ct}, \mathrm{Ch}, \mathrm{Rp}) = \mathsf{true}$. SND-security is a computational relaxation of this, asking that it be computationally hard to create a challenge and response where commitment reproducibility succeeds but the transcript is rejecting. This is formalized in game $\mathbf{G}^{\mathrm{snd}}_{\mathsf{ID},\mathcal{A}}(\lambda)$ in Figure 5. Via oracle New, the adversary can initialize a user (we are in the multi-user setting) and obtain not only its public key but also its secret key. It outputs

```
Game $\mathbf{G}^{\mathrm{snd}}_{\mathsf{ID},\mathcal{A}}(\lambda)$
$n \leftarrow 0$
$(\textsc{Ch}, \textsc{Rp}, i) \leftarrow\!\!\$\; \mathcal{A}^{\mathrm{New}}(1^\lambda)$
$\textsc{Ct} \leftarrow \mathsf{ID.CR}(1^\lambda, pk_i, \textsc{Ch}, \textsc{Rp})$
$d \leftarrow \mathsf{ID.V}(1^\lambda, pk_i, \textsc{Ct}, \textsc{Ch}, \textsc{Rp})$
Return $(d = \mathsf{false}) \wedge (\textsc{Ct} \neq \bot)$

New()
$n \leftarrow n + 1$ ; $(pk_n, sk_n) \leftarrow\!\!\$\; \mathsf{ID.Kg}(1^\lambda)$
Return $(pk_n, sk_n)$
```

```
Game $\mathbf{G}^{\mathrm{cns}}_{\mathsf{ID},\mathcal{A}}(\lambda)$
$n \leftarrow 0$
$(\textsc{Ct}_1, \textsc{Ch}, \textsc{Rp}, i) \leftarrow\!\!\$\; \mathcal{A}^{\mathrm{New}}(1^\lambda)$
$\textsc{Ct}_0 \leftarrow \mathsf{ID.CR}(1^\lambda, pk_i, \textsc{Ch}, \textsc{Rp})$
$d_1 \leftarrow \mathsf{ID.V}(1^\lambda, pk_i, \textsc{Ct}_1, \textsc{Ch}, \textsc{Rp})$
Return $(d_1 = \mathsf{true}) \wedge (\textsc{Ct}_0 \neq \textsc{Ct}_1)$

New()
$n \leftarrow n + 1$ ; $(pk_n, sk_n) \leftarrow\!\!\$\; \mathsf{ID.Kg}(1^\lambda)$
Return $(pk_n, sk_n)$
```

Figure 5: Games defining soundness (SND-security) and consistency (CNS-security) of a commitment reproducible identification scheme ID.

a challenge $\textsc{Ch} \in \mathsf{ID.ChS}(\lambda)$ and response $\textsc{Rp}$, as well as a pointer to some user $i \in \{1, \ldots, n\}$. It wins if the commitment reproducing algorithm, given $pk_i, \textsc{Ch}, \textsc{Rp}$, returns a non-$\bot$ value but the corresponding transcript is rejected by the verifier. Let $\mathbf{Adv}^{\mathrm{snd}}_{\mathsf{ID},\mathcal{A}}(\lambda) = \Pr[\mathbf{G}^{\mathrm{snd}}_{\mathsf{ID},\mathcal{A}}(\lambda)]$. We say that ID is SND-secure if the function $\mathbf{Adv}^{\mathrm{snd}}_{\mathsf{ID},\mathcal{A}}(\cdot)$ is negligible for every PT adversary $\mathcal{A}$.

We turn to *consistency*. Again, to understand it we start with *Perfect Consistency* (P-CNS). This asks that for all $\lambda \in \mathbb{N}$, all $(pk, sk) \in [\mathsf{ID.Kg}(1^\lambda)]$, all $\textsc{Ch} \in \mathsf{ID.ChS}(\lambda)$ and all $\textsc{Ct}, \textsc{Rp}$ we have: If $\textsc{Ct} \neq \mathsf{ID.CR}(1^\lambda, pk, \textsc{Ch}, \textsc{Rp})$ then $\mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp}) \neq \mathsf{true}$. CNS-security is a computational relaxation of this, asking that it be computationally hard to create an accepting transcript in which the commitment is different from the one given by the commitment reproducing algorithm. This is formalized using game $\mathbf{G}^{\mathrm{cns}}_{\mathsf{ID},\mathcal{A}}(\lambda)$ in Figure 5. Via oracle New, the adversary can initialize a user and obtain both its keys. It outputs $\textsc{Ct}, \textsc{Ch}, \textsc{Rp}$ with $\textsc{Ch} \in \mathsf{ID.ChS}(\lambda)$ and a pointer to some user $i \in \{1, \ldots, n\}$. It wins if the transcript is accepting but the commitment reproducing algorithm returns a commitment different from the one in the transcript. Let $\mathbf{Adv}^{\mathrm{cns}}_{\mathsf{ID},\mathcal{A}}(\lambda) = \Pr[\mathbf{G}^{\mathrm{cns}}_{\mathsf{ID},\mathcal{A}}(\lambda)]$. We say that ID is CNS-secure if the function $\mathbf{Adv}^{\mathrm{cns}}_{\mathsf{ID},\mathcal{A}}(\cdot)$ is negligible for every PT adversary $\mathcal{A}$.

For convenience of our reductions, the definitions of soundness and consistency are in the multi-user setting. A standard hybrid argument shows that single user security (captured as security relative to adversaries making only one call to New) implies multi-user security. This reduction is not tight, the advantage degrading linearly in the number of queries to New. When we say that the results in our paper are underlain by tight reductions we mean that the reductions in Theorems 2 and 4 are tight to the assumptions made in these theorems, which are the multi-user versions of SND and CNS, respectively.

It is obvious that P-SND implies SND and P-CNS implies CNS. However soundness and consistency are distinct notions and turn out to be incomparable. The full picture of the relationship between the notions is given in Figure 8.

Signature scheme relations. We give the formal result statements and proofs underlying the picture at the bottom of Figure 1. We start with whether uf-security of $\mathsf{DS}_{\mathrm{ct}}$ implies that of $\mathsf{DS}_{\mathrm{ch}}$. The following Proposition says that in general (meaning, with no conditions on the commitment reproducing algorithm other than completeness) the answer is "no." Theorem 2 will show that SND-security of ID suffices to make the answer "yes." For simplicity the Proposition sets the signing algorithm to the canonical one, but the Theorem holds for *all* signing algorithms.

Proposition 1. *Let* ID^* *be a commitment reproducible identification scheme and* F *a hash function compatible with* ID^*. *Assume signature scheme* $\mathsf{DS}^*_{\mathrm{ct}} = \mathbf{gFS}_{\mathrm{ct}}[\mathsf{ID}^*, \mathsf{F}, \mathsf{SC}_{\mathsf{ID}^*,\mathsf{F}}]$ *is uf-secure. Then* *there is a commitment reproducible identification scheme* ID *such that* F *is compatible with* ID *and (1)* $\mathsf{DS}_{\rm ct} = \mathbf{gFS}_{\rm ct}[\mathsf{ID}, \mathsf{F}, \mathsf{SC}_{\mathsf{ID},\mathsf{F}}]$ *is uf-secure but (2)* $\mathsf{DS}_{\rm ch} = \mathbf{gFS}_{\rm ch}[\mathsf{ID}, \mathsf{F}, \mathsf{SC}_{\mathsf{ID},\mathsf{F}}]$ *is not uf secure.*

Proof. Let ID have the same key generation algorithm, commitment algorithm and challenge space as ID^*. The other algorithms of ID are as follows:

```
$\underline{\mathsf{ID.Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St})}$
Return $(\mathsf{ID}^*.\mathsf{Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St}), 0)$
$\underline{\mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, (\textsc{Rp}^*, b))}$
Return $((b = 0) \wedge \mathsf{ID}^*.\mathsf{V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp}^*))$
```

This suffices to ensure claim (1), namely $\mathsf{DS}_{\rm ct}$ is uf-secure. Formally, this claim is proved by a reduction showing how to translate an adversary violating uf-security of $\mathsf{DS}_{\rm ct}$ into one violating uf-security of $\mathsf{DS}^*_{\rm ct}$. The key element is that signatures can be translated between the schemes by adding a 0 to, or removing a 0 from, the response. We omit the details. Note also that ID preserves the correctness error of ID^*. Now modify the commitment reproducing algorithm as follows:

```
$\underline{\mathsf{ID.CR}(1^\lambda, pk, \textsc{Ch}, (\textsc{Rp}^*, b))}$
If $(b = 0)$ then return $\mathsf{ID}^*.\mathsf{CR}(1^\lambda, pk, \textsc{Ch}, \textsc{Rp}^*)$
Else return 0
```

Note that $\mathsf{ID.CR}$ continues to satisfy completeness. Now claim (2) is justified by the following attack:

```
$\underline{\text{Adversary } \mathcal{A}_{\rm ch}^{\rm New,Sign,H}(1^\lambda)}$
$pk \leftarrow\$ \text{New}()$ ; $M \leftarrow 0$ ; $\textsc{Ct} \leftarrow 0$ ; $\textsc{Ch} \leftarrow \mathsf{F}(1^\lambda, pk, (\textsc{Ct}, M))$ ; $\textsc{Rp} \leftarrow (0, 1)$
Return $(M, (\textsc{Ch}, \textsc{Rp}), 1)$
```

Since $\mathsf{ID.CR}(1^\lambda, pk, \textsc{Ch}, (0, 1))$ returns $\textsc{Ct}$ we have $\mathbf{Adv}^{\rm uf}_{\mathsf{DS}_{\rm ch},\mathcal{A}_{\rm ch}}(\lambda) = 1$. □

If ID has the stronger property of being SND-secure, then uf-security of $\mathsf{DS}_{\rm ct}$ does transfer to $\mathsf{DS}_{\rm ch}$. Note that ID as constructed in the proof of Proposition 1 is *not* SND-secure, so there is no contradiction. Hence the Proposition can also be viewed as showing that the SND-security assumption is necessary for the following Theorem. For conciseness, the theorem statement is asymptotic, but it is underlain by a tight reduction explicitly stated and proved in the proof.

Theorem 2. *Let* ID *be a commitment reproducible identification scheme,* F *a hash function compatible with* ID *and* S *a signing algorithm compatible with* ID, F*. Let* $\mathsf{DS}_{\rm ct} = \mathbf{gFS}_{\rm ct}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ *and* $\mathsf{DS}_{\rm ch} = \mathbf{gFS}_{\rm ch}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$*. Assume* ID *is SND-secure and* $\mathsf{DS}_{\rm ct}$ *is uf-secure. Then* $\mathsf{DS}_{\rm ch}$ *is uf-secure.*

This result holds regardless of F, S, meaning no (extra) conditions are put on these, which means we cover both canonical and with-abort signing via the choices of S shown in Figure 4. As a clarification, note that whether or not $\mathsf{DS}_{\rm ct}$ is uf-secure depends, of course, on the choices of $\mathsf{ID}, \mathsf{F}, \mathsf{S}$, but that is orthogonal to our results. Establishing uf-security of $\mathsf{DS}_{\rm ct}$ is the responsibility of the user of the theorem. For example it might be done using an existing, general proof like that of [1].

Proof of Theorem 2. Given signing adversary $\mathcal{A}_{\rm ch}$ making $q_{\rm New}$ queries to New we construct signing adversary $\mathcal{A}_{\rm ct}$ and soundness adversary $\mathcal{A}$ such that for all $\lambda \in \mathbb{N}$

$$\mathbf{Adv}^{\rm uf}_{\mathsf{DS}_{\rm ch},\mathcal{A}_{\rm ch}}(\lambda) \leq \mathbf{Adv}^{\rm uf}_{\mathsf{DS}_{\rm ct},\mathcal{A}_{\rm ct}}(\lambda) + \mathbf{Adv}^{\rm snd}_{\mathsf{ID},\mathcal{A}}(\lambda) \ . \tag{1}$$

```
Games G_0, G_1
$n \leftarrow 0$; $S \leftarrow \emptyset$
$(M, \sigma, i) \leftarrow\$ \mathcal{A}_{\mathrm{ch}}^{\mathrm{New},\mathrm{Sign},\mathrm{H}}(1^\lambda)$
$(\mathrm{CH}, \mathrm{RP}) \leftarrow \sigma$
$\mathrm{CT} \leftarrow \mathsf{ID.CR}(1^\lambda, pk_i, \mathrm{CH}, \mathrm{RP})$
If $((i, M) \in S) \vee (\mathrm{CT} = \bot)$ then
    return false
$\mathrm{CH}' \leftarrow \mathsf{F}^{\mathrm{H}}(1^\lambda, pk_i, (\mathrm{CT}, M))$
$d \leftarrow \mathsf{ID.V}(1^\lambda, pk_i, \mathrm{CT}, \mathrm{CH}, \mathrm{RP})$
Return $(\mathrm{CH} = \mathrm{CH}') \wedge (d = \mathsf{true})$ // Game $\mathbf{G}_0$
Return $(\mathrm{CH} = \mathrm{CH}') \wedge (d = \mathsf{false})$ // Game $\mathbf{G}_1$

New()
$n \leftarrow n + 1$
$(pk_n, sk_n) \leftarrow\$ \mathsf{DS.Kg}(1^\lambda)$
Return $pk_n$

Sign(i, M)
$\sigma \leftarrow\$ \mathsf{DS}_{\mathrm{ch}}.\mathsf{Sign}^{\mathrm{H}}(1^\lambda, pk_i, sk_i, M)$
$S \leftarrow S \cup \{(i, M)\}$ ; Return $\sigma$

H(W, ℓ)
If $\mathrm{HT}[W, \ell] = \bot$ then $\mathrm{HT}[W, \ell] \leftarrow\$ \{0,1\}^\ell$
Return $\mathrm{HT}[W, \ell]$
```

```
Adversary $\mathcal{A}_{\mathrm{ct}}^{\mathrm{New},\mathrm{Sign},\mathrm{H}}(1^\lambda)$
$n \leftarrow 0$
$(M, \sigma, i) \leftarrow\$ \mathcal{A}_{\mathrm{ch}}^{\mathrm{New}^*,\mathrm{Sign}^*,\mathrm{H}}(1^\lambda)$
$(\mathrm{CH}, \mathrm{RP}) \leftarrow \sigma$
$\mathrm{CT} \leftarrow \mathsf{ID.CR}(1^\lambda, pk_i, \mathrm{CH}, \mathrm{RP})$
Return $(M, (\mathrm{CT}, \mathrm{RP}), i)$

New*()
$n \leftarrow n + 1$ ; $pk_n \leftarrow\$ \mathrm{New}()$
Return $pk_n$

Sign*(i, M)
$\sigma \leftarrow\$ \mathrm{Sign}(i, M)$
If $(\sigma = \bot)$ then return $\bot$
$(\mathrm{CT}, \mathrm{RP}) \leftarrow \sigma$
$\mathrm{CH} \leftarrow \mathsf{F}^{\mathrm{H}}(1^\lambda, pk_i, (\mathrm{CT}, M))$
Return $(\mathrm{CH}, \mathrm{RP})$

Adversary $\mathcal{A}^{\mathrm{New}}(1^\lambda)$
$n \leftarrow 0$ ; $(M, \sigma, i) \leftarrow\$ \mathcal{A}_{\mathrm{ch}}^{\mathrm{New}^*,\mathrm{Sign}^*,\mathrm{H}^*}(1^\lambda)$
$(\mathrm{CH}, \mathrm{RP}) \leftarrow \sigma$
Return $(\mathrm{CH}, \mathrm{RP}, i)$

New*()
$n \leftarrow n + 1$ ; $(pk_n, sk_n) \leftarrow\$ \mathrm{New}()$
Return $pk_n$

Sign*(i, M)
Return $\mathsf{DS}_{\mathrm{ch}}.\mathsf{Sign}^{\mathrm{H}^*}(1^\lambda, pk_i, sk_i, M)$

H*(W, ℓ)
If $\mathrm{HT}[W, \ell] = \bot$ then $\mathrm{HT}[W, \ell] \leftarrow\$ \{0,1\}^\ell$
Return $\mathrm{HT}[W, \ell]$
```

Figure 6: Games and adversaries for proof of Theorem 2. For the adversaries, a star superscript to a procedure indicates that it is a subroutine in the code of the corresponding adversary, constructed by it to simulate an oracle expected by $\mathcal{A}_{\mathrm{ch}}$.

Adversaries $\mathcal{A}_{\mathrm{ct}}$ and $\mathcal{A}$ preserve the running time of $\mathcal{A}_{\mathrm{ch}}$ and number of queries to the New oracle. Adversary $\mathcal{A}_{\mathrm{ct}}$ additionally preserves the number of Sign and H queries of $\mathcal{A}_{\mathrm{ch}}$. The theorem follows. We now prove Equation (1). Towards this fix $\lambda \in \mathbb{N}$ and consider games G_0 and G_1 defined in Figure 6. The games are the same except for the final return statement, which differs for the two games as shown. The games run $\mathcal{A}_{\mathrm{ch}}$ as per game $\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ch}},\mathcal{A}_{\mathrm{ch}}}(\lambda)$, so that $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ch}},\mathcal{A}_{\mathrm{ch}}}(\lambda)$ is the probability that $\mathrm{CH} = \mathrm{CH}'$. This is partitioned into the events $(\mathrm{CH} = \mathrm{CH}') \wedge (d = \mathsf{true})$ and $(\mathrm{CH} = \mathrm{CH}') \wedge (d = \mathsf{false})$, which implies that $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ch}},\mathcal{A}_{\mathrm{ch}}}(\lambda) = \Pr[\mathrm{G}_0] + \Pr[\mathrm{G}_1]$. Intuitively, $d = \mathsf{true}$ means we can violate uf-security of $\mathsf{DS}_{\mathrm{ct}}$ while $d = \mathsf{false}$ means we violate SND-security of ID. To capture this formally, consider the adversaries defined in Figure 6. We claim that

$$\Pr[\mathrm{G}_0] \le \mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ct}},\mathcal{A}_{\mathrm{ct}}}(\lambda) \quad \text{and} \quad \Pr[\mathrm{G}_1] \le \mathbf{Adv}^{\mathrm{snd}}_{\mathsf{ID},\mathcal{A}}(\lambda)\ , \tag{2}$$

which establishes Equation (1). Adversary $\mathcal{A}_{\mathrm{ct}}$ runs $\mathcal{A}_{\mathrm{ch}}$, responding to the latter's H queries via its own H oracle, and simulating $\mathcal{A}_{\mathrm{ch}}$'s New and Sign oracles via the shown procedures New* (that invokes $\mathcal{A}_{\mathrm{ct}}$'s New oracle) and Sign* (that invokes $\mathcal{A}_{\mathrm{ct}}$'s Sign oracle). Finally it transforms the signature. Adversary $\mathcal{A}_{\mathrm{ct}}$ wins game $\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ct}},\mathcal{A}_{\mathrm{ct}}}(\lambda)$ if its output $(M, (\mathrm{CT}, \mathrm{RP}), i)$ satisfies

$\mathsf{ID.V}(1^\lambda, pk_i, \textsc{Ct}, \textsc{Rp}, \textsc{Ch}) = \mathsf{true}$, which happens when game G_0 returns true, justifying the first equation in (2). Adversary $\mathcal{A}$ runs $\mathcal{A}_{\mathrm{ch}}$, simulating all the latter's oracles as shown. An important point here is that $\mathcal{A}$'s New oracle returns not only the public key but also the secret key, which is used by Sign* to simulate $\mathcal{A}_{\mathrm{ch}}$'s Sign oracle. Adversary $\mathcal{A}$ wins game $\mathbf{G}^{\mathrm{snd}}_{\mathsf{ID},\mathcal{A}}(\lambda)$ if its output $(\textsc{Ch}, \textsc{Rp}, i)$ satisfies $\textsc{Ct} \leftarrow \mathsf{ID.CR}(1^\lambda, pk_i, \textsc{Ch}, \textsc{Rp})$ is not $\perp$ yet $\mathsf{ID.V}(1^\lambda, pk_i, \textsc{Ct}, \textsc{Ch}, \textsc{Rp}) = \mathsf{true}$, which happens when game G_1 returns true, justifying the second equation in (2). $\square$

We turn to the converse, asking whether uf-security of $\mathsf{DS}_{\mathrm{ch}}$ implies that of $\mathsf{DS}_{\mathrm{ct}}$. The results are analogous to those above. Proposition 3 says that in general the answer is "no," and Theorem 4 says that it becomes "yes" assuming ID is CNS-secure.

Proposition 3. *Let* ID^* *be a commitment reproducible identification scheme and* F *a hash function compatible with* ID^*. *Assume signature scheme* $\mathsf{DS}^*_{\mathrm{ch}} = \mathbf{gFS}_{\mathrm{ch}}[\mathsf{ID}^*, \mathsf{F}, \mathsf{SC}_{\mathsf{ID}^*,\mathsf{F}}]$ *is uf-secure. Then there is a commitment reproducible identification scheme* ID *such that* F *is compatible with* ID *and (1)* $\mathsf{DS}_{\mathrm{ch}} = \mathbf{gFS}_{\mathrm{ch}}[\mathsf{ID}, \mathsf{F}, \mathsf{SC}_{\mathsf{ID},\mathsf{F}}]$ *is uf-secure but (2)* $\mathsf{DS}_{\mathrm{ct}} = \mathbf{gFS}_{\mathrm{ct}}[\mathsf{ID}, \mathsf{F}, \mathsf{SC}_{\mathsf{ID},\mathsf{F}}]$ *is not uf secure.*

Proof. Let ID have the same key generation algorithm, commitment algorithm and challenge space as ID^*. The other algorithms of ID are as follows:

```
$\underline{\mathsf{ID.Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St})}$
Return $(\mathsf{ID}^*.\mathsf{Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St}), 0)$
$\underline{\mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, (\textsc{Rp}^*, b))}$
Return $((b = 1) \vee \mathsf{ID}^*.\mathsf{V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp}^*))$

$\underline{\mathsf{ID.CR}(1^\lambda, pk, \textsc{Ch}, (\textsc{Rp}^*, b))}$
If $(b = 0)$ then return $\mathsf{ID}^*.\mathsf{CR}(1^\lambda, pk, \textsc{Ch}, \textsc{Rp}^*)$
Else return $\perp$
```

This suffices to ensure claim (1), namely $\mathsf{DS}_{\mathrm{ch}}$ is uf-secure. Formally, this claim is proved by a reduction showing how to translate an adversary violating uf-security of $\mathsf{DS}_{\mathrm{ch}}$ into one violating uf-security of $\mathsf{DS}^*_{\mathrm{ch}}$. The key element is that signatures can be translated between the schemes by adding a 0 to, or removing a 0 from, the response, and that if $b = 0$, then the commitments given by the commitment reproducing algorithms will be the same for both schemes. We omit the details. Note that ID preserves the correctness error of ID^* and $\mathsf{ID.CR}$ continues to satisfy completeness. Now claim (2) is justified by the following attack:

```
$\underline{\text{Adversary } \mathcal{A}_{\mathrm{ct}}^{\mathrm{New,Sign,H}}(1^\lambda)}$
$pk \leftarrow\$ \mathrm{New}()$ ; $M \leftarrow 0$ ; $\textsc{Ct} \leftarrow 0$ ; $\textsc{Ch} \leftarrow \mathsf{F}(1^\lambda, pk, (\textsc{Ct}, M))$ ; $\textsc{Rp} \leftarrow (0, 1)$
Return $(M, (\textsc{Ct}, \textsc{Rp}), 1)$
```

By definition $\mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \mathsf{F}(1^\lambda, pk, (\textsc{Ct}, M)), \textsc{Rp}) = \mathsf{true}$ since the second element of $\textsc{Rp}$ is 1, so $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ct}},\mathcal{A}_{\mathrm{ct}}}(\lambda) = 1$. $\square$

Theorem 4. *Let* ID *be a commitment reproducible identification scheme,* F *a hash function compatible with* ID *and* S *a signing algorithm compatible with* ID, F. *Let* $\mathsf{DS}_{\mathrm{ct}} = \mathbf{gFS}_{\mathrm{ct}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ *and* $\mathsf{DS}_{\mathrm{ch}} = \mathbf{gFS}_{\mathrm{ch}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$. *Assume* ID *is CNS-secure and* $\mathsf{DS}_{\mathrm{ch}}$ *is uf-secure. Then* $\mathsf{DS}_{\mathrm{ct}}$ *is uf-secure.*

```
Game $\mathbf{G}_0$, $\mathbf{G}_1$
    $n \leftarrow 0$; $S \leftarrow \emptyset$
    $(M, \sigma, i) \leftarrow\$ \mathcal{A}_{\mathrm{ct}}^{\mathrm{New,Sign,H}}(1^\lambda)$
    If $((i, M) \in S)$ then return $\mathsf{false}$
    $(\textsc{Ct}, \textsc{Rp}) \leftarrow \sigma$
    $\textsc{Ch} \leftarrow \mathsf{F}^{\mathrm{H}}(1^\lambda, pk_i, (\textsc{Ct}, M))$
    $d \leftarrow \mathsf{ID.V}(1^\lambda, pk_i, \textsc{Ct}, \textsc{Ch}, \textsc{Rp})$
    $\textsc{Ct}' \leftarrow \mathsf{ID.CR}(1^\lambda, pk_i, \textsc{Ch}, \textsc{Rp})$
    $\textsc{Ch}' \leftarrow \mathsf{F}^{\mathrm{H}}(1^\lambda, pk_i, (\textsc{Ct}', M))$
    Return $(d = \mathsf{true}) \wedge (\textsc{Ch} = \textsc{Ch}')$ // Game $\mathbf{G}_0$
    Return $(d = \mathsf{true}) \wedge (\textsc{Ch} \neq \textsc{Ch}')$ // Game $\mathbf{G}_1$

New()
    $n \leftarrow n + 1$
    $(pk_n, sk_n) \leftarrow\$ \mathsf{DS.Kg}(1^\lambda)$
    Return $pk_n$

Sign$(i, M)$
    $\sigma \leftarrow\$ \mathsf{DS}_{\mathrm{ct}}.\mathsf{Sign}^{\mathrm{H}}(1^\lambda, pk_i, sk_i, M)$
    $S \leftarrow S \cup \{(i, M)\}$
    Return $\sigma$

H$(W, \ell)$
    If $\mathrm{HT}[W, \ell] = \perp$ then $\mathrm{HT}[W, \ell] \leftarrow\$ \{0,1\}^\ell$
    Return $\mathrm{HT}[W, \ell]$
```

```
Adversary $\mathcal{A}_{\mathrm{ch}}^{\mathrm{New,Sign,H}}(1^\lambda)$
    $n \leftarrow 0$
    $(M, \sigma, i) \leftarrow\$ \mathcal{A}_{\mathrm{ct}}^{\mathrm{New}^*,\mathrm{Sign}^*,\mathrm{H}}(1^\lambda)$
    $(\textsc{Ct}, \textsc{Rp}) \leftarrow \sigma$
    $\textsc{Ch} \leftarrow \mathsf{F}^{\mathrm{H}}(1^\lambda, pk_i, (\textsc{Ct}, M))$
    Return $(M, (\textsc{Ch}, \textsc{Rp}), i)$

New*()
    $n \leftarrow n + 1$
    $pk_n \leftarrow\$ \mathrm{New}()$
    Return $pk_n$

Sign*$(i, M)$
    $\sigma \leftarrow\$ \mathrm{Sign}(i, M)$
    If $(\sigma = \perp)$ then return $\perp$
    $(\textsc{Ch}, \textsc{Rp}) \leftarrow \sigma$
    $\textsc{Ct} \leftarrow \mathsf{ID.CR}(1^\lambda, pk_i, \textsc{Ch}, \textsc{Rp})$
    Return $(\textsc{Ct}, \textsc{Rp})$
```

```
Adversary $\mathcal{A}^{\mathrm{New}}(1^\lambda)$
    $n \leftarrow 0$
    $(M, \sigma, i) \leftarrow\$ \mathcal{A}_{\mathrm{ct}}^{\mathrm{New}^*,\mathrm{Sign}^*,\mathrm{H}}(1^\lambda)$
    $(\textsc{Ct}, \textsc{Rp}) \leftarrow \sigma$; $\textsc{Ch} \leftarrow \mathsf{F}^{\mathrm{H}}(1^\lambda, pk_i, (\textsc{Ct}, M))$
    Return $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp}, i)$

New*()
    $n \leftarrow n + 1$
    $(pk_n, sk_n) \leftarrow\$ \mathrm{New}()$
    Return $pk_n$

Sign*$(i, M)$
    Return $\mathsf{DS}_{\mathrm{ct}}.\mathsf{Sign}^{\mathrm{H}}(1^\lambda, pk_i, sk_i, M)$

H$(W, \ell)$
    If $\mathrm{HT}[W, \ell] = \perp$ then $\mathrm{HT}[W, \ell] \leftarrow\$ \{0,1\}^\ell$
    Return $\mathrm{HT}[W, \ell]$
```

Figure 7: Games and adversaries for proof of Theorem 4. For the adversaries, a star superscript to a procedure indicates that it is a subroutine in the code of the corresponding adversary, constructed by it to simulate an oracle expected by $\mathcal{A}_{\mathrm{ct}}$.

Proof of Theorem 4. Given signing adversary $\mathcal{A}_{\mathrm{ct}}$ making q_{New} queries to New we construct signing adversary $\mathcal{A}_{\mathrm{ch}}$ and consistency adversary $\mathcal{A}$ such that for all $\lambda \in \mathbb{N}$

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ct}},\mathcal{A}_{\mathrm{ct}}}(\lambda) \leq \mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ch}},\mathcal{A}_{\mathrm{ch}}}(\lambda) + \mathbf{Adv}^{\mathrm{cns}}_{\mathsf{ID},\mathcal{A}}(\lambda) \; . \tag{3}$$

$\mathcal{A}_{\mathrm{ch}}$ and $\mathcal{A}$ preserve the running time of $\mathcal{A}_{\mathrm{ct}}$ and number of queries to the New oracle. $\mathcal{A}_{\mathrm{ch}}$ additionally preserves the number of Sign and H queries of $\mathcal{A}_{\mathrm{ct}}$. The theorem follows. We now prove Equation (3). Towards this fix $\lambda \in \mathbb{N}$ and consider games G_0 and G_1 defined in Figure 7. The games are the except for the final return statement, which differs for the two games as shown. The games run $\mathcal{A}_{\mathrm{ct}}$ as per $\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ct}},\mathcal{A}}(\lambda)$, so that $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ct}},\mathcal{A}_{\mathrm{ct}}}(\lambda)$ is the probability that $d = \mathsf{true}$. This is partitioned into the events $(d = \mathsf{true}) \wedge (\textsc{Ch} = \textsc{Ch}')$ and $(d = \mathsf{true}) \wedge (\textsc{Ch} \neq \textsc{Ch}')$, which implies that

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ct}},\mathcal{A}_{\mathrm{ct}}}(\lambda) = \Pr[\mathbf{G}_0] + \Pr[\mathbf{G}_1] \; .$$

$$\begin{tikzcd}
& \text{P-SND} \wedge \text{P-CNS} \arrow[dl] \arrow[dr] \arrow[d, leftrightarrow] & \\
\text{P-SND} \arrow[d, shift left] \arrow[drr, "|" marking] & \text{P-UNIQ} & \text{P-CNS} \arrow[d, shift right] \arrow[dll, "|" marking] \\
\text{SND} \arrow[u, shift left, "|" marking] & & \text{CNS} \arrow[u, shift right, "|" marking]
\end{tikzcd}$$

Figure 8: Relations between security notions for commitment reproducible identification scheme. Arrows denote implications and barred arrows denote separations.

Intuitively, $(\textsc{Ch} = \textsc{Ch}')$ means that we can violate uf-security of $\mathsf{DS}_{\mathrm{ch}}$, whereas $(\textsc{Ch} \neq \textsc{Ch}')$ means that ID is not CNS-secure. To capture this formally, consider the adversaries defined in Figure 7. We claim that

$$\Pr[\mathbf{G}_0] \leq \mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ch}},\mathcal{A}_{\mathrm{ch}}}(\lambda) \quad \text{and} \quad \Pr[\mathbf{G}_1] \leq \mathbf{Adv}^{\mathrm{cns}}_{\mathsf{ID},\mathcal{A}}(\lambda)\ , \tag{4}$$

which establishes Equation (4). Adversary $\mathcal{A}_{\mathrm{ch}}$ runs $\mathcal{A}_{\mathrm{ct}}$, responding to the latter's H queries via its own H oracle, and simulating $\mathcal{A}_{\mathrm{ct}}$'s New and Sign oracles via the shown procedures New* (that invokes $\mathcal{A}_{\mathrm{ch}}$'s New oracle) and Sign* (that invokes $\mathcal{A}_{\mathrm{ch}}$'s Sign oracle). Finally it transforms the signature. Adversary $\mathcal{A}_{\mathrm{ch}}$ wins game $\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS}_{\mathrm{ch}},\mathcal{A}_{\mathrm{ch}}}(\lambda)$ if its output $(M, (\textsc{Ch}, \textsc{Rp}), i)$ satisfies $\textsc{Ch} = \mathsf{F}^{\mathrm{H}}(1^\lambda, pk_i, (\textsc{Ct}, M))$ with $\textsc{Ct} \leftarrow \mathsf{ID.CR}(1^\lambda, pk_i, \textsc{Ch}, \textsc{Rp})$, which happens when game G_0 returns true, justifying the first equation in (4). Adversary $\mathcal{A}$ runs $\mathcal{A}_{\mathrm{ct}}$, simulating all the latter's oracles as shown. An important point here is that $\mathcal{A}$'s New oracle returns not only the public key but also the secret key, which is used by Sign* to simulate $\mathcal{A}_{\mathrm{ct}}$'s Sign oracle. Adversary $\mathcal{A}$ wins game $\mathbf{G}^{\mathrm{cns}}_{\mathsf{ID},\mathcal{A}}(\lambda)$ if its output $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp}, i)$ satisfies $\textsc{Ct}' \neq \textsc{Ct}$, where $\textsc{Ct}' \leftarrow \mathsf{ID.CR}(1^\lambda, pk_i, \textsc{Ch}, \textsc{Rp})$, and $\mathsf{ID.V}(1^\lambda, pk_i, \textsc{Ct}, \textsc{Ch}, \textsc{Rp}, \textsc{Ch}) = \mathsf{true}$. But the second winning condition of game $\mathbf{G}_1$, namely $\textsc{Ch} \neq \textsc{Ch}'$, implies that $\textsc{Ct} \neq \textsc{Ct}'$, justifying the second equation in (4). $\square$

Recall that the interest of $\mathbf{gFS}_{\mathrm{tr}}$ is that the first proofs were for this variant [27]. However the following says it is equivalent in uf-security to $\mathbf{gFS}_{\mathrm{ct}}$.

Theorem 5. *Let* ID *be an identification scheme,* F *a hash function compatible with* ID *and* S *a signing algorithm compatible with* ID, F. *Let* $\mathsf{DS}_{\mathrm{ct}} = \mathbf{gFS}_{\mathrm{ct}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ *and* $\mathsf{DS}_{\mathrm{tr}} = \mathbf{gFS}_{\mathrm{tr}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$. *Then* $\mathsf{DS}_{\mathrm{ct}}$ *is uf-secure if and only if* $\mathsf{DS}_{\mathrm{tr}}$ *is uf-secure.*

The formal proof, as usual, is by reduction, and the reductions in both directions are tight. The idea is simple, namely that the challenge $\textsc{Ch}$ in a transcript signature $(\textsc{Ct}, \textsc{Ch}, \textsc{Rp})$ must be $\textsc{Ch} = \mathsf{F}^{\mathrm{H}}(1^\lambda, pk, (\textsc{Ct}, \textsc{Rp}))$ and is thus effectively redundant given $\textsc{Ct}, \textsc{Rp}$. If not present, it can be added, and, if present, it can be removed. This enables all the necessary simulations and transforms for the proofs. We omit the details.

4 Identification relations

We have defined several attributes of commitment reproducing identification schemes: P-UNIQ, P-SND, SND, P-CNS, CNS. Figure 8 determines the relations between the five notions, in the style introduced by [6]. An arrow XX $\rightarrow$ YY is an implication: *every* commitment reproducible identification scheme that has property XX also has property YY. A barred arrow XX $\not\rightarrow$ YY is a separation: *there exists* a commitment reproducible identification scheme having property XX but not having property YY.

The picture shows a minimal set of implications and separations but determines the relation between *any* two nodes. For example, does P-CNS imply P-SND? No, because if it did we would get a path from P-CNS to SND, contradicting that shown separation.

What emanates from the relations? Recall we have seen that if $\mathsf{DS}_{\rm ct} = \mathbf{gFS}_{\rm ct}[\mathsf{ID},\mathsf{F},\mathsf{S}]$ and $\mathsf{DS}_{\rm ch} = \mathbf{gFS}_{\rm ch}[\mathsf{ID},\mathsf{F},\mathsf{S}]$ then SND suffices for uf-security of $\mathsf{DS}_{\rm ct}$ to imply that of $\mathsf{DS}_{\rm ch}$, and CNS suffices for the converse. Figure 8 says that P-UNIQ would also suffice for (both) these conclusions, but that SND, CNS are strictly weaker assumptions. It also says that SND, CNS are distinct; neither implies the other. In fact even P-SND does not imply CNS, and P-CNS does not imply SND. So the conditions required for uf-security to transfer across $\mathsf{DS}_{\rm ch}$ and $\mathsf{DS}_{\rm ct}$ are not symmetric.

We move on to justify the relations in Figure 8. It is clear that P-UNIQ is equivalent to P-SND $\wedge$ P-CNS, since, by our general conventions, if any of $\text{CT}, \text{CH}, \text{RP} = \bot$ then $\mathsf{ID.V}(pk, \text{CT}, \text{CH}, \text{RP}) = \bot$, and P-SND and P-CNS each correspond to one direction in the iff statement of P-UNIQ. It is also easy to see that SND, being a computational relaxation of P-SND, is implied by it, and correspondingly CNS is implied by P-CNS. We move on to the separations.

SND $\nrightarrow$ P-SND. We provide a counterexample as follows. Start with any commitment reproducible identification scheme ID^* that is SND. We will modify it to a commitment reproducible identification scheme ID that has the same correctness error as ID^* and continues to be SND, but is not P-SND. The idea is to create a weakness on an exponentially-small fraction of the key space. Proceeding, let ID have the same challenge space as ID^*. The other algorithms of ID are as follows:

```
$\underline{\mathsf{ID.Kg}(1^\lambda)}$
$t \leftarrow\$ \{0,1\}^\lambda$ ; $(pk^*, sk^*) \leftarrow\$ \mathsf{ID}^*.\mathsf{Kg}(1^\lambda)$
If $(t = 1^\lambda)$ then $pk \leftarrow (pk^*, 1)$ else $pk \leftarrow (pk^*, 0)$
Return $(pk, sk^*)$

$\underline{\mathsf{ID.Ct}(1^\lambda, (pk^*, p))}$   // $p \in \{0,1\}$
$(\text{CT}, \text{ST}) \leftarrow\$ \mathsf{ID}^*.\mathsf{Ct}(1^\lambda, pk^*)$ ; Return $(\text{CT}, \text{ST})$

$\underline{\mathsf{ID.Rp}(1^\lambda, (pk^*, p), sk^*, \text{CH}, \text{ST})}$   // $p \in \{0,1\}$
Return $(\mathsf{ID}^*.\mathsf{Rp}(1^\lambda, pk^*, sk^*, \text{CH}, \text{ST}), 0)$

$\underline{\mathsf{ID.V}(1^\lambda, (pk^*, p), \text{CT}, \text{CH}, (\text{RP}^*, r))}$   // $p \in \{0,1\}$
If $(r = 0)$ then return $\mathsf{ID}^*.\mathsf{V}(1^\lambda, pk^*, \text{CT}, \text{CH}, \text{RP}^*)$
Return false

$\underline{\mathsf{ID.CR}(1^\lambda, (pk^*, p), \text{CH}, (\text{RP}^*, r))}$   // $p \in \{0,1\}$
If $(p = 1 \wedge r = 1)$ then return 0
If $(p = 0 \wedge r = 1)$ then return $\bot$
Return $\mathsf{ID}^*.\mathsf{CR}(1^\lambda, pk^*, \text{CH}, \text{RP}^*)$
```

Note that ID preserves the correctness error of ID^* and $\mathsf{ID.CR}$ continues to satisfy completeness. The SND-security of ID can be reduced to the assumed SND-security of ID^*. Briefly, in the execution of game $\mathbf{G}^{\rm snd}_{\mathsf{ID},\mathcal{A}}(\lambda)$ with a PT adversary $\mathcal{A}$, let BAD denote the event that some call to New() returns $((pk^*, p), sk^*)$ with $p = 1$. On the one hand, if BAD does not happen, then ID is functioning just like ID^*. On the other hand, the probability that BAD happens is negligible. We omit the details. Now we show that ID is not P-SND. Let $\lambda \in \mathbb{N}$. Then there exists $(pk^*, sk^*) \in [\mathsf{ID}^*.\mathsf{Kg}(1^\lambda)]$ such that $((pk^*, 1), sk^*) \in [\mathsf{ID.Kg}(1^\lambda)]$. Let CH be any challenge in $\mathsf{ID.ChS}(\lambda)$ and let $\text{RP} = (0, 1)$. Then $\mathsf{ID.CR}(1^\lambda, (pk^*, 1), \text{CH}, (0, 1))$ returns the non-$\bot$ value 0 but $\mathsf{ID.V}(1^\lambda, (pk^*, 1), 0, \text{CH}, (0, 1))$ returns false.

CNS $\not\Rightarrow$ P-CNS. This is similar to the above and omitted.

P-SND $\not\Rightarrow$ CNS. We provide a counterexample as follows. Start with any commitment reproducible identification scheme ID^* that has perfect correctness and is P-SND. We will modify it to a commitment reproducible identification scheme ID that has perfect correctness and continues to be P-SND, but is not CNS. Let ID have the same key-generation algorithm, commitment algorithm and challenge space as ID^*. The other algorithms of ID are as follows:

```
$\underline{\mathsf{ID.Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St})}$
Return $(\mathsf{ID}^*.\mathsf{Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St}), 0)$

$\underline{\mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, (\textsc{Rp}^*, r))}$
Return $\mathsf{ID}^*.\mathsf{V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp}^*)$

$\underline{\mathsf{ID.CR}(1^\lambda, pk, \textsc{Ch}, (\textsc{Rp}^*, r))}$
If $(r = 0)$ then return $\mathsf{ID}^*.\mathsf{CR}(1^\lambda, pk, \textsc{Ch}, \textsc{Rp}^*)$
Return $\perp$
```

Note that ID preserves the correctness error of ID^* and $\mathsf{ID.CR}$ continues to satisfy completeness. The P-SND-security of ID follows from the assumed P-SND-security of ID^*. Briefly, if $r = 0$ then ID behaves like ID^*, while if $r \neq 0$ then $\mathsf{ID.CR}$ returns $\perp$, which cannot help violate P-SND. The claim that ID is not CNS-secure is justified by the following attack:

```
$\underline{\text{Adversary } \mathcal{A}^{\text{New}}(1^\lambda)}$
$(pk, sk) \leftarrow\$ \text{New}()$ ; $(\textsc{Ct}, \textsc{St}) \leftarrow\$ \mathsf{ID}^*.\mathsf{Ct}(1^\lambda, pk)$ ; $\textsc{Ch} \leftarrow\$ \mathsf{ID.ChS}(\lambda)$
$\textsc{Rp}^* \leftarrow \mathsf{ID}^*.\mathsf{Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St})$
Return $(\textsc{Ct}, \textsc{Ch}, (\textsc{Rp}^*, 1), 1)$
```

Note $\textsc{Ct} \neq \perp$ because ID^* has perfect correctness. Now $\mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, (\textsc{Rp}^*, 1)) = \mathsf{true}$ but $\mathsf{ID.CR}(1^\lambda, pk, \textsc{Ch}, (\textsc{Rp}^*, 1))$ returns $\perp$. So $\mathbf{Adv}^{\mathrm{cns}}_{\mathsf{ID},\mathcal{A}}(\lambda) = 1$.

P-CNS $\not\Rightarrow$ SND. We provide a counterexample as follows. Start with any commitment reproducible identification scheme ID^* that has perfect correctness and is P-CNS. We will modify it to a commitment reproducible identification scheme ID that has perfect correctness and continues to be P-CNS, but is not SND. Let ID have the same key-generation algorithm, commitment algorithm and challenge space as ID^*. The other algorithms of ID are as follows:

```
$\underline{\mathsf{ID.Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St})}$
Return $(\mathsf{ID}^*.\mathsf{Rp}(1^\lambda, pk, sk, \textsc{Ch}, \textsc{St}), 0)$

$\underline{\mathsf{ID.V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, (\textsc{Rp}^*, r))}$
If $(r = 0)$ then return $\mathsf{ID}^*.\mathsf{V}(1^\lambda, pk, \textsc{Ct}, \textsc{Ch}, \textsc{Rp}^*)$
Else return $\mathsf{false}$

$\underline{\mathsf{ID.CR}(1^\lambda, pk, \textsc{Ch}, (\textsc{Rp}^*, r))}$
Return $\mathsf{ID}^*.\mathsf{CR}(1^\lambda, pk, \textsc{Ch}, \textsc{Rp}^*)$
```

Note that ID preserves the correctness error of ID^* and $\mathsf{ID.CR}$ continues to satisfy completeness. The P-CNS-security of ID follows from the assumed P-CNS-security of ID^*. Briefly, if $r = 0$ then ID behaves like ID^*, while if $r \neq 0$ then $\mathsf{ID.V}$ rejects, which cannot help violate P-SND. The claim that ID is not SND-secure is justified by the following attack:

```
Adversary $\mathcal{A}^{\mathrm{New}}(1^\lambda)$
$(pk, sk) \leftarrow\$ \mathrm{New}()$ ; $(\mathrm{Ct}, \mathrm{St}) \leftarrow\$ \mathsf{ID}^*.\mathsf{Ct}(1^\lambda, pk)$ ; $\mathrm{Ch} \leftarrow\$ \mathsf{ID}.\mathsf{ChS}(\lambda)$
$\mathrm{Rp}^* \leftarrow \mathsf{ID}^*.\mathsf{Rp}(1^\lambda, pk, sk, \mathrm{Ch}, \mathrm{St})$
Return $(\mathrm{Ct}, \mathrm{Ch}, (\mathrm{Rp}^*, 1), 1)$
```

Note that $\mathsf{ID}.\mathsf{CR}(1^\lambda, pk, \mathrm{Ch}, (\mathrm{Rp}^*, 1))$ will return Ct because ID^* has perfect correctness and $\mathsf{ID}^*.\mathsf{CR}$ satisfies completeness. But $\mathsf{ID}.\mathsf{V}(1^\lambda, pk, \mathrm{Ct}, \mathrm{Ch}, (\mathrm{Rp}^*, 1)) = \mathsf{false}$. So $\mathbf{Adv}^{\mathrm{snd}}_{\mathsf{ID},\mathcal{A}}(\lambda) = 1$.

5 Linear identification schemes

We define a class of identification schemes we call linear, and show how many known schemes, both classical and modern, fall in this class. We show that all schemes in the class satisfy P-UNIQ. This means we can conclude P-UNIQ-ness of many existing schemes, and provide a way to check it for future schemes as well. Beyond that the definition serves to unify and better understand the literature by showing that what look like different schemes are actually the same.

Introduction. Let us first describe linear identification schemes informally. In their simplest form, the public key contains a function h that is homomorphic (linear) in the sense that $h(c \cdot x + y) = c \cdot h(x) + h(y)$. The public key also contains $X = h(x)$ where x is the secret key. The prover picks some y and sends $Y = h(y)$ as the commitment. The verifier responds with a challenge c. The prover returns response $z = c \cdot x + y$. The verifier accepts iff $h(z) = c \cdot X + Y$. This reflects many classical schemes. Lattice-based schemes [22] add some twists, with $\perp$ sometimes being returned in place of z, and with regard to the spaces from which quantities are drawn.

This paradigm is understood in the literature. We aim to formalize it. For this we must address many issues, particularly in order to cover the lattice-based schemes. Our framework is asymptotic, so all quantities will be functions of the security parameter λ. The scheme is parameterized by a family of functions LF. We distinguish between functions and their representations: a key L will specify what we called h above as the function $\mathsf{LF}.\mathsf{H}(1^\lambda, L, \cdot)$. The informal description above referred to operations "$\cdot$" and "$+$" about which we have to be careful. We will ask that the domain $\mathsf{LF}.\mathsf{D}(L)$ and range $\mathsf{LF}.\mathsf{R}(L)$ of the function are modules over a ring of scalars, which determines the operations. For the FS paradigm, challenges must be strings, not elements in an abstract space, so we require an embedding function Emb mapping challenge strings Ch to scalars c. In some schemes in the literature, the secret key x and state y are chosen from subsets of the domain, so we parameterize the scheme by algorithms $\mathrm{SKG}, \mathrm{StG}$ to make these choices. Further parameters are an algorithm Flt to filter responses and an algorithm VF for refining the verifier's decision.

Algebra. Suppose S is a ring with multiplicative identity element 1_S. A S-module, or a module over S, is an additive Abelian group M together with an operation $\cdot : S \times M \to M$ such that for all $r, s \in S$ and $x, y \in M$ we have (1) $r \cdot (x + y) = r \cdot x + r \cdot y$ (2) $(r + s) \cdot x = r \cdot x + s \cdot x$ (3) $(rs) \cdot x = r \cdot (s \cdot x)$ (4) $1_S \cdot x = x$. We refer to elements of S as scalars, and to S as the set of scalars. A module is a generalization of a vector space, which is the special case of S being a field.

As an example, let $\mathbb{G}$ be an Abelian group of order m. Then $\mathbb{G}$ is a $\mathbb{Z}_m$-module where for $s \in \mathbb{Z}_m$ and $x \in \mathbb{G}$ we define $s \cdot x = x + x + \cdots + x$ where there are s copies of x in the sum. Here we are writing the group operation additively, so $s \cdot x$ is the exponentiation operation, raising x to the power s.

Linear function families. A *linear function family* LF specifies a PT key-generation algorithm $\mathsf{LF}.\mathsf{K}$ that, via $L \leftarrow\$ \mathsf{LF}.\mathsf{K}(1^\lambda)$, returns a key. Next, it specifies a deterministic PT evaluation algorithm $\mathsf{LF}.\mathsf{H}$ which, for all $\lambda \in \mathbb{N}$ and keys $L \in [\mathsf{LF}.\mathsf{K}(\lambda)]$ defines a function $\mathsf{LF}.\mathsf{H}(1^\lambda, L, \cdot)$:

```
Algorithm $\mathsf{ID.Kg}(1^\lambda)$
    $L \leftarrow\!\!\$\ \mathsf{LF.K}(\lambda)$ ; $x \leftarrow\!\!\$\ \mathrm{SKG}(1^\lambda, L)$
    $X \leftarrow \mathsf{LF.H}(1^\lambda, L, x)$
    Return $((L, X), x)$

Algorithm $\mathsf{ID.Ct}(1^\lambda, (L, X))$
    $y \leftarrow\!\!\$\ \mathrm{StG}(1^\lambda, L)$ ; $Y \leftarrow \mathsf{LF.H}(1^\lambda, L, y)$
    Return $Y$

Algorithm $\mathsf{ID.Rp}(1^\lambda, (L, X), x, \mathrm{CH}, y)$
    $c \leftarrow \mathrm{Emb}(1^\lambda, L, \mathrm{CH})$ ; $z \leftarrow c \cdot x + y$
    $\mathrm{RP} \leftarrow\!\!\$\ \mathrm{Flt}(1^\lambda, L, x, c, z)$
    Return $\mathrm{RP}$

Algorithm $\mathsf{ID.V}(1^\lambda, (L, X), Y, \mathrm{CH}, \mathrm{RP})$
    If $(\mathrm{RP} = \bot)$ then return $\bot$
    $z \leftarrow \mathrm{RP}$
    If $(Y \notin \mathsf{LF.R}(L)$ or $z \notin \mathsf{LF.D}(L))$ then return $\mathsf{false}$
    If $(\mathrm{VF}(1^\lambda, L, Y, \mathrm{CH}, z) \neq \mathsf{true})$ then return $\mathsf{false}$
    $Z \leftarrow \mathsf{LF.H}(1^\lambda, L, z)$ ; $c \leftarrow \mathrm{Emb}(1^\lambda, L, \mathrm{CH})$
    Return $(Z = c \cdot X + Y)$

Algorithm $\mathsf{ID.CR}(1^\lambda, (L, X), \mathrm{CH}, \mathrm{RP})$
    If $(\mathrm{RP} = \bot)$ then return $\bot$
    $z \leftarrow \mathrm{RP}$
    If $(z \notin \mathsf{LF.D}(L))$ then return $\bot$
    $Z \leftarrow \mathsf{LF.H}(1^\lambda, L, z)$ ; $c \leftarrow \mathrm{Emb}(1^\lambda, L, \mathrm{CH})$
    $Y \leftarrow Z - c \cdot X$
    If $(\mathrm{VF}(1^\lambda, L, Y, \mathrm{CH}, z) \neq \mathsf{true})$ then return $\bot$
    Return $Y$
```

```
Algorithm $\mathrm{SKG}(1^\lambda, L)$
    $x \leftarrow\!\!\$\ \mathsf{LF.D}(L)$ ; Return $x$

Algorithm $\mathrm{StG}(1^\lambda, L)$
    $y \leftarrow\!\!\$\ \mathsf{LF.D}(L)$ ; Return $y$

Algorithm $\mathrm{Flt}(1^\lambda, L, x, c, z)$
    Return $z$

Algorithm $\mathrm{VF}(1^\lambda, L, Y, \mathrm{CH}, z)$
    Return $\mathsf{true}$
```

Figure 9: **Top:** Algorithms of the linear identification scheme $\mathsf{ID} = \mathbf{LinID}[\mathsf{LF}, \mathrm{CL}, \mathrm{Emb}, \mathrm{SKG}, \mathrm{StG}, \mathrm{Flt}, \mathrm{VF}]$. **Bottom:** Parameter settings for simple linear schemes.

$\mathsf{LF.D}(L) \to \mathsf{LF.R}(L)$. Here $\mathsf{LF.D}(L)$, $\mathsf{LF.R}(L)$ are the domain and range sets associated to key L, and they are required to be modules over a ring of scalars $\mathsf{LF.S}(L)$. For each $\lambda \in \mathbb{N}$ and key $L \in [\mathsf{LF.K}(\lambda)]$, the function $\mathsf{LF.H}(1^\lambda, L, \cdot)$ is required to be a module homomorphism, meaning

$$\mathsf{LF.H}(1^\lambda, L, c \cdot x + y) = c \cdot \mathsf{LF.H}(1^\lambda, L, x) + \mathsf{LF.H}(1^\lambda, L, y)$$

for all $c \in \mathsf{LF.S}(L)$ and all $x, y \in \mathsf{LF.D}(L)$. As clarifications, on the left above, the $c \cdot x$ is multiplication of domain module element x by scalar c and the addition $c \cdot x + y$ is in the domain module. On the right, the operations are in the range module, which does not have to be the same as the domain module, so in particular the "$\cdot$" and "$+$" could mean different things on the two sides of the equation.

Linear identification schemes. Let LF be a linear function family as above. We define a transform **LinID** that determines a commitment reproducible identification scheme $\mathsf{ID} = \mathbf{LinID}[\mathsf{LF}, \mathrm{CL}, \mathrm{Emb}, \mathrm{SKG}, \mathrm{StG}, \mathrm{Flt}, \mathrm{VF}]$ from LF and a number of other parameters that we will describe below. The algorithms of the identification scheme are shown in Figure 9, and the challenge length is $\mathsf{ID.cl} = \mathrm{CL}$. We say that an identification scheme ID is linear if there exist $\mathsf{LF}, \mathrm{CL}, \mathrm{Emb}, \mathrm{SKG}, \mathrm{StG}, \mathrm{Flt}, \mathrm{VF}$ such that $\mathsf{ID} = \mathbf{LinID}[\mathsf{LF}, \mathrm{CL}, \mathrm{Emb}, \mathrm{SKG}, \mathrm{StG}, \mathrm{Flt}, \mathrm{VF}]$.

We now say what are the parameters. CL: $\mathbb{N} \to \mathbb{N}$ is a function that, as above, prescribes the challenge length. Emb is a deterministic PT *embedding* algorithm that via $c \leftarrow \mathrm{Emb}(1^\lambda, L, \mathrm{CH})$ returns a point in $\mathsf{LF.S}(L)$. (Think of it as embedding a challenge into the scalar space.) SKG is a PT *secret generation* algorithm that via $x \leftarrow\!\!\$\ \mathrm{SKG}(1^\lambda, L)$ generates a point x in $\mathsf{LF.D}(L)$. StG is a PT *state generation* algorithm that via $y \leftarrow\!\!\$\ \mathrm{StG}(1^\lambda, L)$ generates a point y in $\mathsf{LF.D}(L)$. Flt is a PT *filtering* algorithm that via $\mathrm{RP} \leftarrow\!\!\$\ \mathrm{Flt}(1^\lambda, L, x, c, z)$ generates a response required to be in the set $\{z, \bot\}$. (Think of it as filtering an optimistic choice of response z, returning z

under some conditions, otherwise returning $\perp$.) VF is a deterministic PT algorithm that via $d \leftarrow \mathrm{VF}(1^\lambda, L, Y, \mathrm{Ch}, z)$ returns a boolean decision. This completes the list of ingredients used in the algorithms of the identification scheme in Figure 9.

In many linear schemes, the bulk of the parameters take the trivial, default values shown at the bottom of Figure 9. We call such schemes *simple* linear identification schemes, and may omit the trivial parameters, writing $\mathsf{ID} = \mathbf{LinID}[\mathsf{LF}, \mathrm{CL}, \mathrm{Emb}]$. Simple linear identification schemes have perfect correctness.

Linear schemes are P-UNIQ. We show that any linear identification scheme ID satisfies P-UNIQ. By Figure 8, this means it satisfies SND and CNS. So by Theorems 2 and 4, if one of $\mathsf{DS}_{\mathrm{ct}} = \mathbf{gFS}_{\mathrm{ct}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ or $\mathsf{DS}_{\mathrm{ch}} = \mathbf{gFS}_{\mathrm{ch}}[\mathsf{ID}, \mathsf{F}, \mathsf{S}]$ is uf-secure, then, automatically, so is the other. A strength of this result is that it holds across all choices of hash function F and signing algorithms S, in particular covering both classical signing and signing with aborts.

Proposition 6. *Let* $\mathsf{ID} = \mathbf{LinID}[\mathsf{LF}, \mathrm{SKG}, \mathrm{StG}, \mathrm{CL}, \mathrm{Emb}, \mathrm{Flt}, \mathrm{VF}]$ *be a linear identification scheme as above. Then* ID *is P-UNIQ.*

Proof. Let $\lambda \in \mathbb{N}$ and $((L, X), x) \in [\mathsf{ID.Kg}(1^\lambda)]$. Let $\mathrm{Ch} \in \{0,1\}^{\mathrm{CL}(\lambda)}$ and let Y, Rp be non-$\perp$. Let $z \leftarrow \mathrm{Rp}$ and let $c \leftarrow \mathrm{Emb}(1^\lambda, L, \mathrm{Ch})$. (1) Assume $\mathsf{ID.V}(1^\lambda, pk, Y, \mathrm{Ch}, \mathrm{Rp}) = \mathsf{true}$. By definition of ID.V it must be that $\mathrm{VF}(1^\lambda, L, Y, \mathrm{Ch}, z) = \mathsf{true}$ and $Y \in \mathsf{LF.R}(L)$ and $z \in \mathsf{LF.D}(L)$. By definition of ID.V it must be that $Z = c \cdot X + Y$. By the algebraic properties of modules it must be that $Y = Z - c \cdot X$. So $\mathsf{ID.CR}(1^\lambda, (L, X), \mathrm{Ch}, \mathrm{Rp})$ will return Y. (2) Assume $\mathsf{ID.CR}(1^\lambda, (L, X), \mathrm{Ch}, \mathrm{Rp})$ returns Y. By definition of ID.CR this means $\mathrm{VF}(1^\lambda, L, Y, \mathrm{Ch}, z) = \mathsf{true}$, so $Y \in \mathsf{LF.R}(L)$ and $z \in \mathsf{LF.D}(L)$. By the algebraic properties of modules it must be that $Z = c \cdot X + Y$. So $\mathsf{ID.V}(1^\lambda, pk, Y, \mathrm{Ch}, \mathrm{Rp}) = \mathsf{true}$. $\square$

Examples of linear identification schemes. Many identification schemes in the literature can be shown to be linear. Examples include classical schemes like FS [16], Sch [28], GQ [19] and Ok [26]. They also include lattice based schemes like Ly [22]. We will illustrate by showing linearity for a few of these schemes.

Sch is linear. We show that the Sch scheme [28] is a simple linear identification scheme. The intuition, in the informal language of our discussion above, is simple, namely we set $h(x) = g^x$ where g is a generator of a group $\mathbb{G}$. The homomorphic property comes down to $g^{c \cdot x + y} = (g^x)^c \cdot g^y$. Below, we prove the claim formally by exhibiting $\mathsf{LF}, \mathrm{CL}, \mathrm{Emb}$ such that $\mathsf{Sch} = \mathbf{LinID}[\mathsf{LF}, \mathrm{CL}, \mathrm{Emb}]$. Our convention being that the module operation is additive means that g^x becomes $x \cdot g$, which may look natural to those used to elliptic curves and less so to others.

Figure 10 shows the settings. Key generation algorithm $\mathsf{LF.K}(1^\lambda)$ returns key $L = (\langle \mathbb{G} \rangle, p, g)$ where $\langle \mathbb{G} \rangle$ is a description of a cyclic group $\mathbb{G}$ of prime order $p \in \{2^{\lambda-1}+1, \ldots, 2^\lambda - 1\}$, and $g \in \mathbb{G}$ is a generator of $\mathbb{G}$. (There are many ways to make these choices, and we do not need to pin one down.) The set of scalars $\mathsf{LF.S}((\langle \mathbb{G} \rangle, p, g)) = \mathbb{Z}_p$ is a ring (in fact a field) as required. The domain $\mathsf{LF.D}((\langle \mathbb{G} \rangle, p, g)) = \mathbb{Z}_p$ is an additive Abelian group that is a $\mathbb{Z}_p$-module with the $\cdot$ operation being multiplication in $\mathbb{Z}_p$. The range is $\mathsf{LF.R}((\langle \mathbb{G} \rangle, p, g)) = \mathbb{G}$. Writing the group operation additively, this is an Abelian (because cyclic) group that is a $\mathbb{Z}_p$-module when we define $c \cdot X$ to be the sum of $c \in \mathbb{Z}_p$ copies of $X \in \mathbb{G}$. (This is X raised to the power c.) We define $\mathsf{LF.H}(1^\lambda, (\langle \mathbb{G} \rangle, p, g), x) = x \cdot g$. (The sum of $x \in \mathbb{Z}_p$ copies of the generator $g \in \mathbb{G}$, this is g raised to the power x.) The function $\mathsf{LF.H}(1^\lambda, (\langle \mathbb{G} \rangle, p, g), \cdot)$ is a module homomorphism, as required. We would pick the challenge length CL so that $\mathrm{CL}(\lambda) \leq \lambda - 1$, and then define $\mathrm{Emb}(1^\lambda, (\langle \mathbb{G} \rangle, p, g), \mathrm{Ch})$ to be the integer $\mathrm{StToInt}(\mathrm{Ch})$ whose string representation is Ch. This integer is in the ring $\mathbb{Z}_p$ of scalars, as required, because

L	$(\langle\mathbb{G}\rangle, p, g)$
$\mathsf{LF.S}(L)$	$\mathbb{Z}_p$
$\mathsf{LF.D}(L)$	$\mathbb{Z}_p$
$\mathsf{LF.R}(L)$	$\mathbb{G}$
$\mathsf{LF.H}(1^\lambda, L, x)$	$x \cdot g$
$\mathrm{Emb}(1^\lambda, L, \mathrm{C{\scriptsize H}})$	$\mathrm{StToInt}(\mathrm{C{\scriptsize H}})$

L	$(q, n, m, d, \sigma, \kappa, a)$
$\mathsf{LF.S}(L)$	$R_{q,n}$
$\mathsf{LF.D}(L)$	$R_{q,n}^m$
$\mathsf{LF.R}(L)$	$R_{q,n}$
$\mathsf{LF.H}(1^\lambda, L, x)$	$\mathrm{IP}(a, x)$
$\mathrm{Emb}(1^\lambda, L, \mathrm{C{\scriptsize H}})$	Return $c \in \{\, c \in R_{q,n} \,:\, \|c\|_1 \le \kappa \,\}$
$\mathrm{SKG}(1^\lambda, L)$	Return $x \leftarrow\!\!\$\, D_{q,n}(\sigma)^m$
$\mathrm{StG}(1^\lambda, L)$	Return $y \leftarrow\!\!\$\, D_{q,n}(mn\sigma\kappa)^m$
$\mathrm{Flt}(1^\lambda, L, x, c, z)$	If $z \in D_{q,n}(\sigma\kappa(mn-1))^m$ then return z else return $\perp$
$\mathrm{VF}(1^\lambda, L, Y, \mathrm{C{\scriptsize H}}, z)$	Return $(z \in D_{q,n}(\sigma\kappa(mn-1))^m)$

Figure 10: **Top:** Settings to capture Sch as a simple linear identification scheme. **Bottom:** Settings to capture Ly as a linear identification scheme.

$p > 2^{\lambda-1}$. Looking at the algorithms of Figure 9, we see that $\mathbf{LinID}[\mathsf{LF}, \mathrm{CL}, \mathrm{Emb}]$ recovers the Schnorr identification scheme [28].

Ly IS LINEAR. We show that the Ly scheme [22] is a linear identification scheme. The linear function family is from [22, 24]. To prove the claim we exhibit $\mathsf{LF}, \mathrm{CL}, \mathrm{Emb}, \mathrm{SKG}, \mathrm{StG}, \mathrm{Flt}, \mathrm{VF}$ such that $\mathsf{Ly} = \mathbf{LinID}[\mathsf{LF}, \mathrm{CL}, \mathrm{Emb}, \mathrm{SKG}, \mathrm{StG}, \mathrm{Flt}, \mathrm{VF}]$.

First, some notation. For positive integers q, n we let $R_{q,n}$ be the ring $\mathbb{Z}_q/\langle X^n + 1\rangle$, so ring elements are polynomials of degree at most n with coefficients in $\mathbb{Z}_q$. We can accordingly represent an element of $R_{q,n}$ as an n-vector over $\mathbb{Z}_q$, and, for such a vector x, talk of its L_p-norm, denoted $\|a\|_p$, for $p \in \{1, \ldots, \infty\}$. For a real number w, we let $D_{q,n}(w) = \{\, s \in R_{q,n} \,:\, \|s\|_\infty \le w \,\}$ be the set of all ring elements whose L_∞-norm is at most w. By $R_{q,n}^m$ we denote the set of m-vectors over $R_{q,n}$. With addition being component-wise, this is a module over $R_{q,n}$, with $\cdot$ simply being multiplication in $R_{q,n}$. Note that elements of $R_{q,n}^m$ are vectors whose components are polynomials, themselves represented as vectors. We define the inner product of two such vectors $a, x \in R_{q,n}^m$ by

$$\mathrm{IP}(a, x) = a[1] \cdot x[1] + \cdots + a[m] \cdot x[m] \in R_{q,n} \ ,$$

the additions and multiplications above being over $R_{q,n}$.

Figure 10 shows the settings. Key generation algorithm $\mathsf{LF.K}(1^\lambda)$ returns a key $L = (q, n, m, d, \sigma, \kappa, a)$ where $q, n, m, d, \sigma, \kappa$ are positive integers such that q is odd, n is 2 raised to an odd power, and $a \leftarrow\!\!\$\, R_{q,n}^m$. (There are many ways to make the non-prescribed choices, and we do not need to pin one down.) The set of scalars is the ring, $\mathsf{LF.S}((q, n, m, d, \sigma, \kappa, a)) = R_{q,n}$. We set the domain to $\mathsf{LF.D}((q, n, m, d, \sigma, \kappa, a)) = R_{q,n}^m$, which is a $R_{q,n}$-module, and the range to the ring, $\mathsf{LF.R}((q, n, m, d, \sigma, \kappa, a)) = R_{q,n}$, which is also a $R_{q,n}$-module. We define $\mathsf{LF.H}(1^\lambda, (q, n, m, d, \sigma, \kappa, a), x) = \mathrm{IP}(a, x)$ to be the inner product of the vectors a, x. The function $\mathsf{LF.H}(1^\lambda, (q, n, m, d, \sigma, \kappa, a), \cdot)$

is a module homomorphism, as required. The embedding function $\mathrm{Emb}(1^\lambda, (q, n, m, d, \sigma, \kappa, a), \mathrm{C{\small H}})$ returns a scalar in the subset $\{ c \in R_{q,n} : \|c\|_1 \leq \kappa \}$ of the set of all scalars $R_{q,n}$. (How exactly it does this we have not been able to determine. The scheme described in [22] picks the challenge directly at random from this set, but, as we have explained, challenges must be strings of a known length for the FS transforms, so some embedding is needed.) The secret x is drawn at random from the set $D_{q,n}(\sigma)^m \subseteq \mathsf{LF.D}((q, n, m, d, \sigma, \kappa, a))$, meaning x is an m-vector each of whose components (a polynomial represented as an n-vector) has L_∞-norm at most σ. The state y is drawn from $D_{q,n}(mn\sigma\kappa)^m$. Filtering algorithm $\mathrm{Flt}(1^\lambda, (q, n, m, d, \sigma, \kappa, a), x, c, z)$ returns z if $z \in D_{q,n}(\sigma\kappa(mn-1))^m$ and $\perp$ otherwise. Verification algorithm $\mathrm{VF}(1^\lambda, (q, n, m, d, \sigma, \kappa, a), Y, \mathrm{C{\small H}}, z)$ returns true if $z \in D_{q,n}(\sigma\kappa(mn-1))^m$ and false otherwise.

Acknowledgments

The second author thanks Tom Ristenpart for asking about the security of the different variants of Fiat-Shamir signatures. We thank the NordSec 2018 reviewers for their comments.

References

[1] M. Abdalla, J. H. An, M. Bellare, and C. Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433. Springer, Heidelberg, Apr. / May 2002.

[2] M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly-secure signatures from lossy identification schemes. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 572–590. Springer, Heidelberg, Apr. 2012.

[3] J. Alwen, Y. Dodis, and D. Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 36–54. Springer, Heidelberg, Aug. 2009.

[4] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.

[5] M. Bellare and W. Dai. Defending against key exfiltration: Efficiency improvements for big-key cryptography via large-alphabet subkey prediction. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 17*, pages 923–940. ACM Press, Oct. / Nov. 2017.

[6] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 26–45. Springer, Heidelberg, Aug. 1998.

[7] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. *Journal of Cryptology*, 22(1):1–61, Jan. 2009.

[8] M. Bellare, B. Poettering, and D. Stebila. From identification to signatures, tightly: A framework and generic transforms. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 435–464. Springer, Heidelberg, Dec. 2016.

[9] M. Bellare, B. Poettering, and D. Stebila. Deterring certificate subversion: Efficient double-authentication-preventing signatures. In S. Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 121–151. Springer, Heidelberg, Mar. 2017.

[10] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993.

[11] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006.

[12] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang. High-speed high-security signatures. In B. Preneel and T. Takagi, editors, *CHES 2011*, volume 6917 of *LNCS*, pages 124–142. Springer, Heidelberg, Sept. / Oct. 2011.

[13] N. Bindel, S. Akleylek, E. Alkim, P. S. L. M. Barreto, J. Buchmann, E. Eaton, G. Gutoski, J. Kramer, P. Longa, H. Polat, J. E. Ricardini, and G. Zanon. qTESLA. Technical report, National Institute of Standards and Technology, 2017. Available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

[14] S. Brands. Untraceable off-line cash in wallets with observers (extended abstract). In D. R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 302–318. Springer, Heidelberg, Aug. 1994.

[15] L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehle. CRYSTALS – dilithium: Digital signatures from module lattices. Cryptology ePrint Archive, Report 2017/633, 2017. `http://eprint.iacr.org/2017/633`.

[16] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, Aug. 1987.

[17] S. D. Galbraith, C. Petit, and J. Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 3–33. Springer, Heidelberg, Dec. 2017.

[18] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, Apr. 1988.

[19] L. C. Guillou and J.-J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 123–128. Springer, Heidelberg, May 1988.

[20] E. Kiltz, D. Masny, and J. Pan. Optimal security proofs for signatures from identification schemes. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 33–61. Springer, Heidelberg, Aug. 2016.

[21] LANIX. Things that use Ed25519, Aug. 2018. `https://ianix.com/pub/curve25519-deployment.html`.

[22] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Heidelberg, Dec. 2009.

[23] V. Lyubashevsky. Lattice signatures without trapdoors. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Heidelberg, Apr. 2012.

[24] V. Lyubashevsky and D. Micciancio. Generalized compact Knapsacks are collision resistant. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 144–155. Springer, Heidelberg, July 2006.

[25] K. Ohta and T. Okamoto. On concrete security treatment of signatures derived from identification. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 354–369. Springer, Heidelberg, Aug. 1998.

[26] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In E. F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 31–53. Springer, Heidelberg, Aug. 1993.

[27] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

[28] C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.