From Identification to Signatures, Tightly: A Framework and Generic Transforms

Mihir Bellare[1] Bertram Poettering[2] Douglas Stebila[3]

June 2016

Abstract

This paper provides a framework to treat the problem of building signature schemes from identification schemes in a unified and systematic way. The outcomes are (1) Alternatives to the Fiat-Shamir transform that, applied to trapdoor identification schemes, yield signature schemes with tight security reductions to standard assumptions (2) An understanding and characterization of existing transforms in the literature. One of our transforms has the added advantage of producing signatures shorter than produced by the Fiat-Shamir transform. Reduction tightness is important because it allows the implemented scheme to use small parameters (thereby being as efficient as possible) while retaining provable security.

[1] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: mihir@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grants CNS-1228890 and CNS-1526801, a gift from Microsoft corporation and ERC Project ERCC (FP7/615074).

[2] Ruhr University Bochum, Universitätsstr. 150, 44801 Bochum, Germany. Email: bertram.poettering@rub.de. URL: http://www.crypto.rub.de/. Supported by ERC Project ERCC (FP7/615074).

[3] Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada. Email: stebilad@mcmaster.ca. URL: https://www.cas.mcmaster.ca/~stebilad/. Supported in part by Australian Research Council (ARC) Discovery Project grant DP130104304.

Contents

1 Introduction

This paper provides a framework to treat the problem of building signature schemes from identification schemes in a unified and systematic way. We are able to explain and characterize existing transforms as well as give new ones whose security proofs give *tight* reductions to *standard* assumptions. This is important so that the implemented scheme can use small parameters, thereby being efficient while retaining provable security. Let us begin by identifying the different elements involved.

ID-TO-SIG TRANSFORMS. Recall that in a three-move identification scheme ID the prover sends a *commitment* Y computed using private randomness y, the verifier sends a random *challenge* c, the prover returns a *response* z computed using y and its secret key *isk*, and the verifier computes a boolean decision from the conversation transcript $Y\|c\|z$ and public key *ivk* (see Fig. 3). We are interested in transforms $\boxed{\textbf{Id2Sig}}$ that take ID and return a signature scheme DS. The transform must be generic, meaning DS is proven to meet some signature security goal $\boxed{\mathrm{P}_{\mathrm{sig}}}$ assuming only that ID meets some identification security goal $\boxed{\mathrm{P}_{\mathrm{id}}}$. This proof is underlain by a reduction $\boxed{\mathrm{P}_{\mathrm{sig}}{\rightarrow}\mathrm{P}_{\mathrm{id}}}$ that may be tight or loose. Boxing an item here highlights elements of interest and choice in the id-to-sig process.

CANONICAL EXAMPLE. In the most canonical example we have, **Id2Sig** = **FS** is the Fiat-Shamir transform [17] ; P_{id} = IMP-PA is security against impersonation under passive attack [15, 1] ; $\mathrm{P}_{\mathrm{sig}}$ = UF is unforgeability under chosen-message attack [20] ; and the reduction $\mathrm{P}_{\mathrm{sig}}{\rightarrow}\mathrm{P}_{\mathrm{id}}$ is that of AABN [1], which is loose.

We are going to revisit this to give other choices of the different elements, but first let us recall some more details of the above. In the Fiat-Shamir transform **FS** [17], a signature of a message m is a pair (Y, z) such that the transcript $Y\|c\|z$ is accepting for $c = \mathrm{H}(Y\|m)$, where H is a random oracle. IMP-PA requires that an adversary given transcripts of honest protocol executions still fails to make the honest verifier accept in an interaction where it plays the role of the prover, itself picking Y any way it likes, receiving a random c, and then producing z. The loss in the $\mathrm{P}_{\mathrm{sig}}{\rightarrow}\mathrm{P}_{\mathrm{id}}$ reduction of AABN [1] is a factor of the number q of adversary queries to the random oracle H: If $\epsilon_{\mathrm{id}}, \epsilon_{\mathrm{sig}}$ denote, respectively, the advantages in breaking the IMP-PA security of ID and the UF security of DS, then $\epsilon_{\mathrm{sig}} \approx q\,\epsilon_{\mathrm{id}}$.

ALGEBRAIC ASSUMPTION TO ID. Suppose a cryptographer wants to build a signature scheme meeting the definition $\mathrm{P}_{\mathrm{sig}}$. The cryptographer would like to base security on some algebraic assumption $\boxed{\mathrm{P}_{\mathrm{alg}}}$. This could be factoring, RSA inversion, bilinear Diffie-Hellman, some lattice assumption, or many others. Given an id-to-sig transform as above, the task amounts to designing an identification scheme ID achieving P_{id} under $\mathrm{P}_{\mathrm{alg}}$. (Then one can just apply the transform to ID.) This proof is underlain by another reduction $\boxed{\mathrm{P}_{\mathrm{id}}{\rightarrow}\mathrm{P}_{\mathrm{alg}}}$ that again may be tight or loose. The tightness of the overall reduction $\mathrm{P}_{\mathrm{sig}}{\rightarrow}\mathrm{P}_{\mathrm{alg}}$ thus depends on the tightness of both $\mathrm{P}_{\mathrm{sig}}{\rightarrow}\mathrm{P}_{\mathrm{id}}$ and $\mathrm{P}_{\mathrm{id}}{\rightarrow}\mathrm{P}_{\mathrm{alg}}$.

CANONICAL EXAMPLE. Continuing with the FS+AABN-based example from above, we would need to build an identification scheme meeting P_{id} = IMP-PA under $\mathrm{P}_{\mathrm{alg}}$. The good news is that a wide swathe of such identification schemes are available, for many choices of $\mathrm{P}_{\mathrm{alg}}$ (GQ [22] under RSA, FS [17] under Factoring, Schnorr [35] under Discrete Log, ...). However the reduction $\mathrm{P}_{\mathrm{id}}{\rightarrow}\mathrm{P}_{\mathrm{alg}}$ is (very) loose.

Again, we are going to revisit this to give other choices of the different elements, but first let us recall some more details of the above. The practical identification schemes here are typically Sigma protocols (this means they satisfy honest-verifier zero-knowledge and special soundness, the latter

meaning that from two accepting conversation transcripts with the same commitment but different challenges, one can extract the secret key) and $\mathrm{P}_{\mathrm{alg}} = \mathrm{KR}$ is the problem of computing the secret key given only the public key. To solve this problem, we have to run a given IMP-PA adversary twice and hope for two successes. The analysis exploits the Reset Lemma of [6]. If $\epsilon_{\mathrm{alg}}, \epsilon_{\mathrm{id}}$ denote, respectively, the advantages in breaking the algebraic problem and the IMP-PA security of ID, then it results in $\epsilon_{\mathrm{id}} \approx \sqrt{\epsilon_{\mathrm{alg}}}$. If ϵ_{sig} is the advantage in breaking UF security of DS, combined with the above, we have $\epsilon_{\mathrm{sig}} \approx q\sqrt{\epsilon_{\mathrm{alg}}}$.

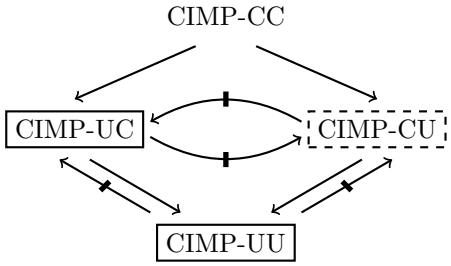
Approach. We see from the above that a tight overall reduction $\mathrm{P}_{\mathrm{sig}} \rightarrow \mathrm{P}_{\mathrm{alg}}$ requires that the $\mathrm{P}_{\mathrm{sig}} \rightarrow \mathrm{P}_{\mathrm{id}}$ and $\mathrm{P}_{\mathrm{id}} \rightarrow \mathrm{P}_{\mathrm{alg}}$ reductions *both* be tight. What we observe is that we have a degree of freedom in achieving this, namely *the choice of the security goal* P_{id} *for the identification scheme.* Our hope is to pick P_{id} such that (1) We can give (new) transforms **Id2Sig** for which $\mathrm{P}_{\mathrm{sig}} \rightarrow \mathrm{P}_{\mathrm{id}}$ is tight, and simultaneously (2) We can give identification schemes such that $\mathrm{P}_{\mathrm{id}} \rightarrow \mathrm{P}_{\mathrm{alg}}$ is tight. We view these as two pillars of an edifice and are able to provide both via our definitions of security of identification under constrained impersonation coupled with some new id-to-sig transforms. We first pause to discuss some prior work, but a peek at Fig. 1 gives an outline of the results we will expand on later.

Prior work. The first proofs of security for **FS**-based signatures [34] reduced UF security of the **FS**-derived signature scheme directly to the hardness of the algebraic problem $\mathrm{P}_{\mathrm{alg}}$, assuming H is a random oracle [7]. These proofs exploit forking lemmas [34, 5, 4]. Modular proofs of the form discussed above, that use identification as an intermediate step, begin with [32, 1]. The modular approach has many advantages. One is that since the id-to-sig transforms are generic, we have only to design and analyze identification schemes. Another is the better understanding and isolation of the role of random oracles: they are used by **Id2Sig** but not in the identification scheme. We have accordingly adopted this approach. Note that both the direct (forking lemma based) and the AABN-based indirect (modular) approach result in reductions of the same looseness we discussed above. Our (alternative but still modular) approaches will remove this loss.

Consideration of reduction tightness for signatures begins with BR [9], whose PSS scheme has a tight reduction to the RSA problem. KW [23] give another signature scheme with a tight reduction to RSA, and they and GJ [18] give signature schemes with tight reductions to the Diffie-Hellman problem. The lack of tightness of the overall reduction for **FS**-based signatures is well recognized as an important problem and drawback. It was first addressed by Micali and Reyzin [29], who give a particular signature scheme with a tight reduction to factoring. It is obtained from a particular identification scheme via a method they call "swap". ABP [2] say that the method generalizes to other factoring-based schemes. However, "swap" has never been stated as a general transform of an identification scheme into a signature scheme; it appears rather as an ad hoc technique to go directly and tightly from the algebraic problem to the signature. This lack of abstraction is perhaps due in part to a lack of definitions, and the ones we provide allow us to fill the gap. In Section 6.5 we elevate the swap method to a general **Swap** transform, characterize the identification schemes to which it applies, and prove that, when it applies, it gives a tight $\mathrm{P}_{\mathrm{sig}} \rightarrow \mathrm{P}_{\mathrm{id}}$ reduction.

ABP [2] show a tight reduction of **FS**-derived GQ signatures to the Φ-hiding assumption of [12]. In contrast, our methods will yield GQ signatures with a tight reduction to the standard one-wayness of RSA. AFLT [3] use a slight variant of the Fiat-Shamir transform to turn lossy identification schemes into signature schemes with security based tightly on key indistinguishability, resulting in signature schemes with tight reductions to the decisional short discrete logarithm problem, the shortest vector problem in ideal lattices and subset sum.

Constrained impersonation. Recall our goal is to define a notion of identification security P_{id}



$\mathrm{P_{id}}$	**Id2Sig** Transform	Trapdoor?	$\mathrm{P_{sig}}$-secure **Signature**	**Reductions** $\mathrm{P_{sig}}\to\mathrm{P_{id}}$/$\mathrm{P_{id}}\to\mathrm{P_{alg}}$
CIMP-CU	**FS**	No	$(Y,z)\ :\ c=\mathrm{H}(Y\|\|m)$	Tight/Loose
CIMP-UC	**MdCmt**	Yes	$(c,z)\ :\ Y=\mathrm{H}(m)\ ;\ c\leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$	Tight/Tight
CIMP-UU	**MdCmtCh**	Yes	$(s,z)\ :\ s\leftarrow\!\!\$\ \{0,1\}\ ;\ Y=\mathrm{H}_1(m)$ $c=\mathrm{H}_2(m\|\|s)$	Tight/Tight
CIMP-CC	**MdCh**	No	$(Y,s,z)\ :\ s\leftarrow\!\!\$\ \{0,1\}^{\mathsf{sl}}$ $c=\mathrm{H}(m\|\|s)$	Tight/Unknown
CIMP-UC	**Swap**	Yes	$(c,z)\ :\ c\leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$ $Y=\mathrm{H}(m\|\|c)$	Tight/Tight

Figure 1: **Top:** Relations between notions $\mathrm{P_{id}}$ of security for an identification scheme ID under constrained impersonation. Solid arrows denote implications, barred arrows denote separations. A solid box around a notion means a tight $\mathrm{P_{id}}\to\mathrm{P_{alg}}$ reduction for Sigma protocols; dotted means a loose one; no box means no known reduction. **Bottom:** Transforms of identification schemes into UUF (row 2,3) or UF (rows 1,4,5) signature schemes. The first column is the assumption $\mathrm{P_{id}}$ on the identification scheme. The third column indicates whether or not the identification scheme is assumed to be trapdoor. $\mathsf{ID.cl}$ is the challenge length and sl is a seed length. In rows 1,4 the commitment Y is chosen at random. The third transform has the shortest signatures, consisting of a response plus a single bit.

---

such that (1) We can give transforms **Id2Sig** for which $\mathrm{P_{sig}}\to\mathrm{P_{id}}$ is tight, and (2) We can give identification schemes such that $\mathrm{P_{id}}\to\mathrm{P_{alg}}$ is tight. In fact our definitional goal is broader, namely to give a framework that allows us to understand and encompass both old and new transforms, the former including **FS** and **Swap**. We do all this with a definitional framework that we refer to as *constrained impersonation*. It yields four particular definitions denoted CIMP-XY for $\mathrm{XY}\in\{\mathrm{CU},\mathrm{UC},\mathrm{UU},\mathrm{CC}\}$. Each, in the role of $\mathrm{P_{id}}$, will be the basis for an id-to-sig transform such that $\mathrm{P_{sig}}\to\mathrm{P_{id}}$ is tight, and two will allow $\mathrm{P_{id}}\to\mathrm{P_{alg}}$ to be tight.

In constrained impersonation we continue, as with IMP-PA, to allow a passive attack in which the adversary $\mathcal{A}$ against the identification scheme ID can obtain transcripts $Y_1\|c_1\|z_1, Y_2\|c_2\|z_2\ldots$ of interactions between the honest prover and verifier. Then $\mathcal{A}$ tries to impersonate, meaning get the honest verifier to accept. If X=C then the *commitment* in this impersonation interaction is adversary-*chosen*, while if X=U (*unchosen*) it must be pegged to a commitment from one of the transcripts. If Y=C, the *challenge* is adversary-chosen, while if Y=U it is as usual picked at random by the verifier. In all cases, multiple impersonation attempts are allowed. The formal definitions are in Section 3. CIMP-CU is a multi-impersonation version of IMP-PA, but the rest are novel.

What do any of these notions have to do with identification if one understands the latter as

the practical goal of proving one's identity to a verifier? Beyond CIMP-CU, very little. In practice it is unclear how one can constrain a prover to only use, in impersonation, a commitment from a prior transcript. It is even more bizarre to allow a prover to pick the challenge. Our definitions however are not trying to capture any practical usage of identification. They view the latter as an analytical tool, an intermediate land allowing a smooth transition from an algebraic problem to signatures. The constrained impersonation notions work well in this regard, as we will see, both to explain and understand existing work and to obtain new signature schemes with tight reductions.

Relations between the four notions of constrained impersonation are depicted in Fig. 1. An arrow A $\rightarrow$ B is an implication: *Every* identification scheme that is A-secure is also B-secure. A barred arrow A $\not\rightarrow$ B is a separation: *There exists* an identification scheme that is A-secure but *not* B-secure. (For now ignore the boxes around notions.) In particular we see that CIMP-UU is weaker than, and CIMP-UC incomparable to, the more standard CIMP-CU. See Proposition 1 for more precise renditions of the implications.

Auxiliary definitions and tools. Before we see how to leverage the constrained impersonation framework, we need a few auxiliary definitions and results that, although simple, are, we believe, of independent interest and utility.

We define a signature scheme to be UUF (Unique Unforgeable) if it is UF with the restriction that a message can be signed at most once. (You are not allowed to twice ask the signing oracle to sign a particular m.) It turns out that some of our id-to-sig transforms naturally achieve UUF, not UF. However there are simple, generic transforms of UUF signature scheme into a UF ones — succinctly, UF$\rightarrow$UUF— that do not introduce much overhead and have tight reductions. One is to remove randomness, and the other is to add it. In more detail, a well-known method to derandomize a signature scheme is to specify the coins by hashing the secret key along with the message. This has been proved to work in some instances [31, 25] but not in general. We observe that this method has the additional benefit of turning a UUF scheme into a UF one. We call the transform **DR**. Theorem 4 shows that it works. (In particular it shows UF-security of the derandomized scheme in a more general setting than was known before.) The second transform, **AR**, appends a random salt to the message before signing and includes the salt in the signature. Theorem 5 shows that it works. The first transform is attractive because it does not increase signature size. The second does, but is standard-model. We stress that the reductions are tight in both cases, so this step does not impact overall tightness. Now we can take (the somewhat easier to achieve) UUF as our goal.

Recall that in an identification scheme, the prover uses private randomness y to generate its commitment Y. We call the scheme *trapdoor* if the prover can pick the commitment Y directly at random from the space of commitments and then compute the associated private randomness y using its secret key via a prescribed algorithm. The concept is implicit in [29] but does not seem to have been formalized before, so we give a formal definition in Section 3. Many existing identification schemes will meet our definition of being trapdoor modulo possibly some changes to the key structure. Thus the GQ scheme of [22] is trapdoor if we add the decryption exponent d to the secret key. With similar changes to the keys, the Fiat-Shamir [17] and Ong-Schnorr [33] identification schemes are trapdoor. The factoring-based identification scheme of [29] is also trapdoor. But not all identification schemes are trapdoor. One that is not is Schnorr's (discrete-log based) scheme [35].

Summary of results. For each notion $\mathrm{P_{id}} \in \{\text{CIMP-CU}, \text{CIMP-UC}, \text{CIMP-UU}, \text{CIMP-CC}\}$ we give an id-to-sig transform that turns any given $\mathrm{P_{id}}$-secure identification scheme ID into a $\mathrm{P_{sig}}$ = UUF signature scheme DS; the transform from CIMP-CC security achieves a UF signature scheme. *The reduction* $\mathrm{P_{sig}}\rightarrow\mathrm{P_{id}}$ *is tight in all four cases.* (To further make the signature scheme UF secure, we can apply the above-mentioned UF$\rightarrow$UUF transforms while preserving tightness.) The table

in Fig. 1 summarizes the results and the transforms. They are discussed in more detail below and then fully in Section 6.

This is one pillar of the edifice, and not useful by itself. The other pillar is the $\mathrm{P}_{\mathrm{id}} \rightarrow \mathrm{P}_{\mathrm{alg}}$ reduction. In the picture at the top of Fig. 1, a solid-line box around P_{id} means that the reduction $\mathrm{P}_{\mathrm{id}} \rightarrow \mathrm{P}_{\mathrm{alg}}$ is tight, a dotted-line box indicates a reduction is possible but is not tight, and no box means no known reduction. These results assume the identification scheme is a Sigma protocol, as most are, and are discussed in Section 4. We see that two points of our framework can be *tightly* obtained from the algebraic problem, so that in these cases the overall $\mathrm{P}_{\mathrm{sig}} \rightarrow \mathrm{P}_{\mathrm{alg}}$ reduction is tight, which was the ultimate goal.

More details on results. The transform from CIMP-CU is the classical **FS** one. The reduction is now tight, even though it was not from IMP-PA [1], simply because CIMP-CU is IMP-PA extended to allow multiple impersonation attempts. The result, which we state as Theorem 9, is implicit in [1], but we give a proof to illustrate how simple the proof now is. In this case our framework serves to better understand, articulate and simplify something implicit in the literature, rather than deliver anything particularly new.

For CIMP-UC, we give a transform called **MdCmt**, for "Message-Derived Commitment", where, to sign m, the signer computes the commitment Y as a hash of the message, picks a challenge at random, uses its trapdoor to compute the coins y corresponding to Y, uses y and the secret key to compute a response z, and returns the challenge and response as the signature. See Section 6.1.

For CIMP-UU, the weakest of the four notions, our transform **MdCmtCh**, for "Message-Derived Commitment and Challenge", has the signer compute the commitment Y as a hash of the message. It then picks a single random bit s and computes the challenge as a hash of the message and seed, returning as signature the seed and response, the latter computed as before. Beyond a tight reduction, this transform has the added feature of *short signatures*, the signature being a response plus a single bit. (In all other transforms, whether prior or ours, the signature is at least a response plus a challenge, often more.) See Section 6.2.

Since CIMP-CC implies CIMP-UC and CIMP-UU (Fig. 1, top), the **MdCmt** and **MdCmtCh** transforms would both work. However, these require the identification scheme to be trapdoor and achieve UUF rather than UF. (The above-mentioned UUF $\rightarrow$ UF transforms are applied on top to get UF.) We give an alternative transform called **MdCh** ("Message-Derived Challenge") from CIMP-CC that directly achieves UF and works (gives a tight reduction) even if the identification scheme is not trapdoor. It has the signer pick a random commitment, produce the challenge as in **MdCmtCh**, namely as a randomized hash of the message, compute the response, and return the conversation transcript as signature. See Section 6.3.

Again the salient fact is that the reductions underlying all four transforms are tight.

To leverage the above we now have to consider achieving CIMP-XY. We do this in Section 4. We give reductions $\mathrm{P}_{\mathrm{id}} \rightarrow \mathrm{P}_{\mathrm{alg}}$ of the P_{id} = CIMP-XY security of identification schemes that are Sigma protocols to their key-recovery (KR) security, the latter being the problem of recovering the secret key given only the public key, which is typically the algebraic problem $\mathrm{P}_{\mathrm{alg}}$ whose hardness is assumed. For CIMP-UC and CIMP-UU the $\mathrm{P}_{\mathrm{id}} \rightarrow \mathrm{P}_{\mathrm{alg}}$ reduction is tight, as per Theorem 2, making these the most attractive starting points. For CIMP-CU we must use the Reset Lemma [6] so the reduction (cf. Theorem 3) is loose. CIMP-CC is a very strong notion and, as we discuss at the end of Section 4, not achieved by Sigma protocols but achievable by other means.

Swap. As indicated above, our framework allows us to generalize the swap method of [29] into an id-to-sig transform **Swap** and understand and characterize what it does. In Section 6.5 we present **Swap** as a generic transform of a trapdoor identification scheme ID to a signature scheme that

is just like **MdCmt** (cf. row 2 of the table of Fig. 1) except that the challenge c is included in the input to the hash function (cf. row 5 of the table of Fig. 1). Recall that **MdCmt** turns a CIMP-UC identification scheme into a UUF signature scheme. We can thence get a UF signature scheme by applying the **AR** transform of Section 5.2. **Swap** is a shortcut, or optimization, of this two step process: it directly turns a CIMP-UC identification scheme into a UF signature scheme by effectively re-using the randomness of **MdCmt** in **AR**. We note that the composition of our **DR** with our **MdCmtCh** yields a UF signature scheme with shorter signatures than **Swap** while also having a tight reduction to the weaker CIMP-UU assumption, and would thus be superior. However we think **Swap** is of historical interest and accordingly present it. See Section 6.5 for details.

Discussion. Schnorr signatures [35] over elliptic curves have been considered superior in performance to factoring or RSA-based signatures based on the argument that one can use smaller groups for the same security. This argument ignores reduction tightness. Schnorr signatures are obtained from the Schnorr identification scheme [35] via the (loose) **FS** transform, while our reductions allow us to obtain factoring and RSA based signatures via tight reductions. If group sizes are chosen based on tightness of reductions to standard assumptions, the performance gap between Schnorr and factoring or RSA based signatures narrows.

An intriguing application area to explore for our transforms is in lattice-based cryptography. Here signatures have been obtained via the **FS** transform [27, 28]. The underlying lattice-based identification schemes do not appear to be trapdoor, so our transforms would not apply. However, via the techniques of MP [30], one can build lattice-based trapdoor identification schemes to which our transforms apply. Whether there is a performance benefit will depend on the trade-off between the added cost from having the trapdoor and the smaller parameters permitted by the improved security reduction.

We measure reduction tightness stringently, in a model where running time, queries and success probability are separate parameters. The picture changes if one considers the expected success ratio, namely the ratio of running time to success probability. Reduction tightness under this metric is considered in PS [34] and the concurrent and independent work of KMP [24].

We establish the classical notion of standard unforgeability (UF) [20]. Our transforms also establish strong unforgeability if the identification scheme has the extra property of unique responses. (For any public key, commitment and challenge, there exists at most one response that the verifier accepts.)

GK [19] give an example of a 3-move protocol where **FS** yields a secure signature scheme in the ROM, but the RO is not instantiable. Their protocol however is not a Sigma protocol, as is assumed for the ones we start with and is true for practical identification schemes. Currently, secure instantiation of the RO, both for **FS** and our transforms, is not ruled out for such identification schemes.

2 Notation and basic definitions

Notation. We let ε denote the empty string. 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. We use $a_1 \| a_2 \| \cdots \| a_n$ as shorthand for $(a_1, a_2, \ldots, a_n)$. Algorithms may be randomized unless otherwise indicated. Running time is worst case. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with random coins r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \leftarrow\!\!\$\, A(x_1, \ldots)$ be the result of picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$. We let $[A(x_1, \ldots)]$ denote the set of all possible outputs of A when invoked with inputs $x_1, \ldots$. We use the code based game playing framework of [10].

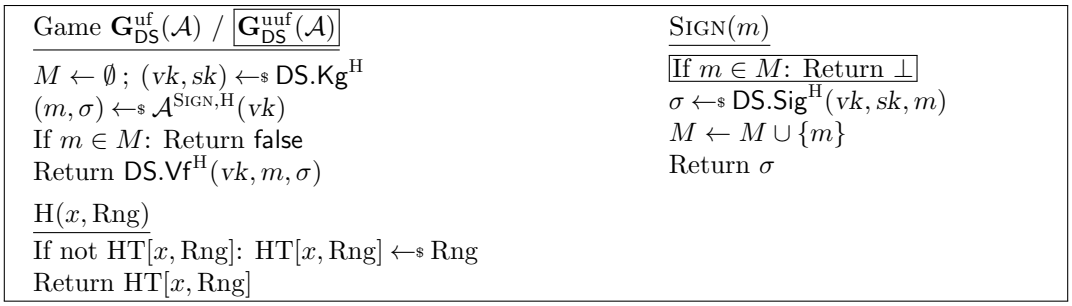


Figure 2: Games defining unforgeability and unique unforgeability of signature scheme DS. Game $\mathbf{G}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A})$ includes the boxed code and game $\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A})$ does not.

(See Fig. 2 for an example.) By $\Pr[\mathrm{G}]$ we denote the event that the execution of game G results in the game returning true. Boolean flags (like bad) in games are assumed initialized to false. By $a_1 \| a_2 \| \cdots a_n \leftarrow x$ we mean that x is parsed into its constituents. We adopt the convention that the running time of an adversary refers to the worst case execution time of the game with the adversary. This means that the time taken for oracles to compute replies to queries is included.

Our treatment of random oracles is more general than usual. In our constructions, we will need random oracles with different ranges. For example we may want one random oracle returning points in a group $\mathbb{Z}_N^*$ and another returning strings of some length l. To provide a single unified definition, we have the procedure H in the games take not just the input x but a description Rng of the set from which outputs are to be drawn at random. Thus $y \leftarrow\!\!\$\ \mathrm{H}(x, \mathbb{Z}_N^*)$ will return a random element of $\mathbb{Z}_N^*$, while $c \leftarrow\!\!\$\ \mathrm{H}(x, \{0,1\}^l)$ will return a random l-bit string, and so on. Sometimes if the range set is understood, it is dropped as an argument.

Signatures. In a signature scheme DS, the signer generates signing key sk and verifying key vk via $(vk, sk) \leftarrow\!\!\$\ \mathsf{DS.Kg}^{\mathrm{H}}$ where H is the random oracle, the latter with syntax as discussed above. Now it can compute a signature $\sigma \leftarrow\!\!\$\ \mathsf{DS.Sig}^{\mathrm{H}}(vk, sk, m)$ on any message $m \in \{0,1\}^*$. A verifier can deterministically compute a boolean $v \leftarrow \mathsf{DS.Vf}^{\mathrm{H}}(vk, m, \sigma)$ indicating whether or not σ is a valid signature of m relative to vk. Correctness as usual requires that $\mathsf{DS.Vf}^{\mathrm{H}}(vk, m, \mathsf{DS.Sig}^{\mathrm{H}}(vk, sk, m)) = \mathsf{true}$ with probability one. Game $\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A})$ associated to DS and adversary $\mathcal{A}$ as per Fig. 2 captures the classical unforgeability notion of [20] lifted to the ROM as per [7], and we let $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A})]$ be the UF-advantage of $\mathcal{A}$. The same figure also defines game $\mathbf{G}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A})$ to capture *unique* unforgeability. The difference is the inclusion of the boxed code, which disallows $\mathcal{A}$ from getting more than one signature on the same message. We let $\mathbf{Adv}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A})]$ be the UUF-advantage of $\mathcal{A}$.

Of course, UF implies UUF, meaning any signature scheme that is UF secure is also UUF secure. The converse is not true, meaning there exist UUF signature schemes that are not UF secure (we will see natural examples in this paper). In Section 5 we give simple, generic and tight ways to turn any given UUF signature scheme into a UF one.

We note that unique unforgeability (UUF) should not be confused with unique signatures as defined in [21, 26]. In a unique signature scheme, there is, for any message, at most one signature the verifier will accept. If a unique signature scheme is UUF then it is also UF. But there are UUF (and UF) schemes that are not unique.

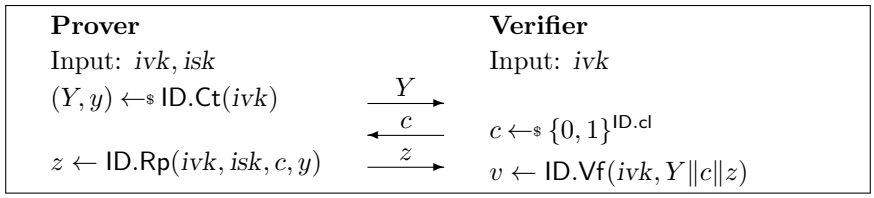


Figure 3: Functioning of an identification scheme ID.

3 Constrained impersonation framework

We introduce a framework of definitions of identification schemes secure against constrained impersonation.

Identification. An identification (ID) scheme ID operates as depicted in Fig. 3. First, via $(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$, the prover generates a public *verification key* ivk, private *identification key* isk, and *trapdoor* itk. Via $(Y, y) \leftarrow\!\!\$\ \mathsf{ID.Ct}(ivk)$ it generates *commitment* Y and corresponding private state y. The verifier sends a random challenge of length $\mathsf{ID.cl}$. The prover's *response* z and the verifier's boolean *decision* v are deterministically computed per $z \leftarrow \mathsf{ID.Rp}(ivk, isk, c, y)$ and $v \leftarrow \mathsf{ID.Vf}(ivk, Y\|c\|z)$, respectively. We assume throughout that identification schemes have perfect correctness. An example ID scheme is the GQ; see Section 7. For basic ID schemes, the trapdoor plays no role; its use arises in trapdoor identification.

Trapdoor identification. We now explain what it means for an ID scheme to be *trapdoor*. Namely there is an algorithm $\mathsf{ID.Ct}^{-1}$ that produces y from Y with the aid of the trapdoor itk. Formally, the outputs of the following two processes must be identically distributed. Both processes generate $(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$. The first process then lets $(Y, y) \leftarrow\!\!\$\ \mathsf{ID.Ct}(ivk)$. The second process picks $Y \leftarrow\!\!\$\ \mathsf{ID.CS}(ivk)$ and lets $y \leftarrow\!\!\$\ \mathsf{ID.Ct}^{-1}(ivk, itk, Y)$. (Here $\mathsf{ID.CS}(ivk)$ is the space of commitments associated to ID.) Both processes return (ivk, isk, itk, Y, y). In a trapdoor scheme no one can distinguish which process was used to generate the output.

Security against impersonation. Classically, the security goal for an identification scheme ID has been impersonation [16, 1]. The framework has two stages. First, the adversary, given ivk but not isk, attacks the honest, isk-using prover. Second, using the information it gathers in the first stage, it engages in an interaction with the verifier, attempting to impersonate the real prover by successfully identifying itself. In the second stage, the adversary, in the role of malicious prover, submits a commitment Y of its choice, receives an honest verifier challenge c, submits a response z of its choice, and wins if $\mathsf{ID.Vf}(ivk, Y\|c\|z) = \mathsf{true}$. A hierarchy of possible first-phase attacks is defined in [6]. In the context of conversion to signatures, the relevant one is the weakest, namely passive attacks, where the adversary is just an eavesdropper and gets honestly-generated protocol transcripts. This is the IMP-PA notion. (Active and even concurrent attacks are relevant in other contexts [6].) We note that in the second stage, the adversary is allowed only one interaction with the honest verifier.

Security against constrained impersonation. We introduce a new framework of goals for identification that we call *constrained impersonation*. There are two dimensions, the commitment dimension X and the challenge dimension Y, for each of which there are two choices, $\mathrm{X} \in \{\mathrm{C}, \mathrm{U}\}$ and $\mathrm{Y} \in \{\mathrm{C}, \mathrm{U}\}$, where C stands for *chosen* and U for *unchosen*. This results in four notions, CIMP-UU, CIMP-UC, CIMP-CU, CIMP-CC. It works as follows. The adversary is allowed a passive attack, namely the ability to obtain transcripts of interactions between the honest prover and the verifier. The choices pertain to the impersonation, when the adversary interacts with the

Game $\mathbf{G}^{\text{cimp-xy}}_{\mathsf{ID}}(\mathcal{P})$

$\mathrm{S} \leftarrow \emptyset$; $i \leftarrow 0$; $j \leftarrow 0$
$(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$
$(k, z) \leftarrow\!\!\$\ \mathcal{P}^{\textsc{Tr},\textsc{Ch}}(ivk)$
If not $(1 \le k \le j)$: Return false
$T \leftarrow \mathrm{CT}[k] \| z$
Return $\mathsf{ID.Vf}(ivk, T)$

$\textsc{Tr}()$
$i \leftarrow i + 1$
$(Y_i, y_i) \leftarrow\!\!\$\ \mathsf{ID.Ct}(ivk)$
$c_i \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$
$z_i \leftarrow \mathsf{ID.Rp}(ivk, isk, c_i, y_i)$
$T \leftarrow Y_i \| c_i \| z_i$; $\mathrm{S} \leftarrow \mathrm{S} \cup \{(Y_i, c_i)\}$
Return T

$\textsc{Ch}(l)$ // xy=uu
If not $(1 \le l \le i)$: Return $\perp$
$j \leftarrow j + 1$; $c \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$
$\mathrm{CT}[j] \leftarrow Y_l \| c$; Return c

$\textsc{Ch}(l, c)$ // xy=uc
If not $(1 \le l \le i)$: Return $\perp$
If $(c = c_l)$: Return $\perp$
$j \leftarrow j + 1$
$\mathrm{CT}[j] \leftarrow Y_l \| c$; Return c

$\textsc{Ch}(Y)$ // xy=cu
$j \leftarrow j + 1$; $c \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$
$\mathrm{CT}[j] \leftarrow Y \| c$; Return c

$\textsc{Ch}(Y, c)$ // xy=cc
If $(Y, c) \in \mathrm{S}$: Return $\perp$
$j \leftarrow j + 1$
$\mathrm{CT}[j] \leftarrow Y \| c$; Return c

Figure 4: Games defining security of identification scheme ID against constrained impersonation under passive attack.

honest verifier in an attempt to make it accept. When X = C, the adversary can send the verifier a commitment of its choice, as in classical impersonation. But when X = U, it cannot. Rather, it is required (constrained) to use a commitment that is from one of the transcripts it obtained in the first phase and thus in particular honestly generated. Next comes the challenge. If Y = U, this is chosen freshly at random, as in the classical setting, but if Y = C, the adversary actually gets to pick its own challenge. Regardless of choices made in these four configurations, to win the adversary must finally supply a correct response. And, also regardless of these choices, the adversary can mount *multiple* attempts to convince the verifier.

For choices $\mathrm{xy} \in \{\mathrm{uu}, \mathrm{uc}, \mathrm{cu}, \mathrm{cc}\}$ of parameters, the formalization considers game $\mathbf{G}^{\text{cimp-xy}}_{\mathsf{ID}}(\mathcal{P})$ of Fig. 4 associated to identification scheme ID and adversary $\mathcal{P}$. We let

$$\mathbf{Adv}^{\text{cimp-xy}}_{\mathsf{ID}}(\mathcal{P}) = \Pr[\mathbf{G}^{\text{cimp-xy}}_{\mathsf{ID}}(\mathcal{P})] \ .$$

The transcript oracle $\textsc{Tr}$ returns upon each invocation a transcript of an interaction between the honest prover and verifier, allowing $\mathcal{P}$ to mount its passive attack, and is the same for all four games. The impersonation attempts are mounted through calls to the $\textsc{Ch}$ oracle, which creates a partial transcript $\mathrm{CT}[j]$ consisting of a commitment and a challenge, where j is a session id, and it returns the challenge. Multiple impersonation attempts are captured by the adversary being allowed to call $\textsc{Ch}$ as often as it wants. Eventually the adversary outputs a session id k and a response z for session k, and wins if the corresponding transcript is accepting. In the UU case, $\mathcal{P}$ would give $\textsc{Ch}$ only an index l of an existing transcript already returned by $\textsc{Tr}$, and $\mathrm{CT}[j]$ consists of the commitment from the l-th transcript together with a fresh random challenge. In the UC case, $\textsc{Ch}$ takes in addition a challenge c chosen by the adversary. The game requires that it be different from c_l (the challenge in the l-th transcript), and $\mathrm{CT}[j]$ then consists of the commitment from the l-th transcript together with this challenge. In CU, the adversary can specify the commitment but the challenge is honestly chosen. In CC, it can specify both, as long as the pair did not occur in a transcript. The adversary can call the oracles as often as it wants and in whatever order it wants.

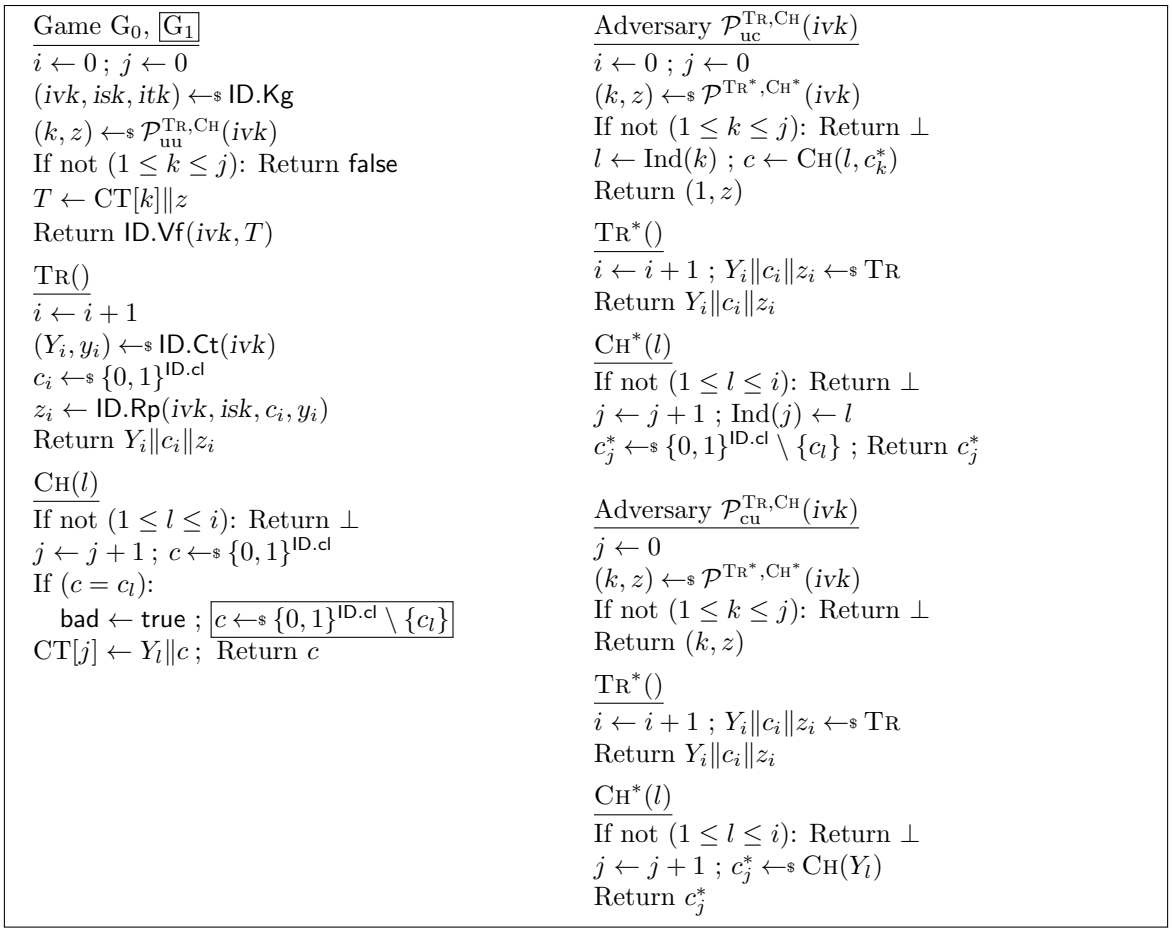


Figure 5: Games and adversaries for proof of Proposition 1 parts **1.** and **2.**

CIMP-CU is a multi-impersonation extension of the classical IMP-PA notion. The other notions are new, and all will be the basis of transforms of identification to signatures admitting *tight* security reductions. CIMP-CU captures a practical identification security goal. As discussed in Section 1, the other notions have no such practical interpretation. However we are not aiming to capture some practical form of identification. We wish to use identification only as an analytical tool in the design of signature schemes. For this purpose, as we will see, our framework and notions are indeed useful, allowing us to characterize past transforms and build new ones.

Implications. Fig. 1 shows the relations between the four CIMP-XY notions. The implications are captured by Proposition 1. (The separations will be discussed below.) The bounds in these claims imply some conditions or assumptions for the implications which we did not emphasize before because they hold for typical identification schemes. Namely, CIMP-UC $\rightarrow$ CIMP-UU assumes the identification scheme has large challenge length. CIMP-CC $\rightarrow$ CIMP-UC assumes it has a large commitment space. CIMP-CC $\rightarrow$ CIMP-CU again assumes it has a large challenge length. We remark that in all but one case, the constructed adversary makes only one Ch query, regardless of how many the starting adversary made.

Proposition 1 Let ID be an identification scheme. Let $\mathsf{ID.CSS} = \min\{\, |\mathsf{ID.CS}(ivk)| \,:\, (ivk, isk, itk) \in [\mathsf{ID.Kg}] \,\}$.

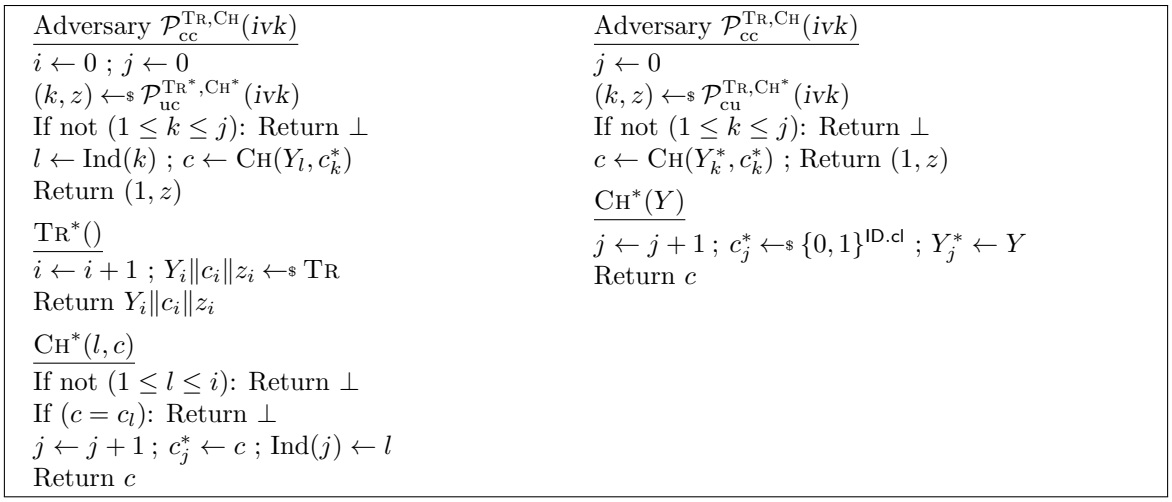


Figure 6: Adversaries for proof of Proposition 1 parts **3.** and **4.**

---

1. [CIMP-UC → CIMP-UU] Given $\mathcal{P}_{\text{uu}}$ making q_c queries to Ch, we construct $\mathcal{P}_{\text{uc}}$ making one Ch query and such that $\mathbf{Adv}_{\mathsf{ID}}^{\text{cimp-uu}}(\mathcal{P}_{\text{uu}}) \le \mathbf{Adv}_{\mathsf{ID}}^{\text{cimp-uc}}(\mathcal{P}_{\text{uc}}) + q_c \cdot 2^{-\mathsf{ID.cl}}$.

2. [CIMP-CU → CIMP-UU] Given $\mathcal{P}_{\text{uu}}$ we construct $\mathcal{P}_{\text{cu}}$ making as many Ch queries as $\mathcal{P}$ and such that $\mathbf{Adv}_{\mathsf{ID}}^{\text{cimp-uu}}(\mathcal{P}_{\text{uu}}) \le \mathbf{Adv}_{\mathsf{ID}}^{\text{cimp-cu}}(\mathcal{P}_{\text{cu}})$.

3. [CIMP-CC → CIMP-UC] Given $\mathcal{P}_{\text{uc}}$ making q_t queries to Tr, we construct $\mathcal{P}_{\text{cc}}$ making one Ch query and such that $\mathbf{Adv}_{\mathsf{ID}}^{\text{cimp-uc}}(\mathcal{P}_{\text{uc}}) \le \mathbf{Adv}_{\mathsf{ID}}^{\text{cimp-cc}}(\mathcal{P}_{\text{cc}}) + q_t(q_t - 1)/2\mathsf{ID.CSS}$.

4. [CIMP-CC → CIMP-CU] Given $\mathcal{P}_{\text{cu}}$ making q_t queries to Tr and q_c queries to Ch, we construct $\mathcal{P}_{\text{cc}}$ making one Ch query and such that $\mathbf{Adv}_{\mathsf{ID}}^{\text{cimp-cu}}(\mathcal{P}_{\text{cu}}) \le \mathbf{Adv}_{\mathsf{ID}}^{\text{cimp-cc}}(\mathcal{P}_{\text{cc}}) + q_t q_c \cdot 2^{-\mathsf{ID.cl}}$.

In all cases, the constructed adversary makes the same number of Tr queries as the starting adversary and has about the same running time.

Proof of Proposition 1: We write Tr^* and Ch^* for subroutines in the code of the constructed adversaries that provide answers to the Tr and Ch queries, respectively, of the given adversary. We now proceed item by item.

For part **1.** consider the games of Fig. 5. Game G_0 omits the boxed code while game G_1 includes it. Game G_0 is game $\mathbf{G}_{\mathsf{ID}}^{\text{cimp-uu}}(\mathcal{P}_{\text{uu}})$ of Fig. 4. Games G_0 and G_1 are identical until bad. By the Fundamental Lemma of Game Playing [10] we have

$$\begin{aligned}\mathbf{Adv}_{\mathsf{ID}}^{\text{cimp-uu}}(\mathcal{P}_{\text{uu}}) &= \Pr[\text{G}_0] = \Pr[\text{G}_1] + (\Pr[\text{G}_0] - \Pr[\text{G}_1]) \le \Pr[\text{G}_1] + \Pr[\text{G}_0 \text{ sets } \mathsf{bad}] \\ &\le \Pr[\text{G}_1] + \frac{q_c}{2^{\mathsf{ID.cl}}} \ .\end{aligned}$$

We construct adversary $\mathcal{P}_{\text{uc}}$ as shown in the top right of Fig. 5. It executes $\mathcal{P}_{\text{uu}}$, responding to Tr and Ch queries of the latter via the shown procedures, which are subroutines in the code of $\mathcal{I}$. It simulates for $\mathcal{P}_{\text{uu}}$ the environment of game G_1. This guarantees that the challenge c_k^* in its Ch query is different from c_l. Thus $\Pr[\text{G}_1] \le \mathbf{Adv}_{\mathsf{ID}}^{\text{cimp-uc}}(\mathcal{P}_{\text{uc}})$, completing the proof.

For part **2.** the adversary $\mathcal{P}_{\text{cu}}$ is on the bottom right in Fig. 5. It answers a $\text{Ch}(l)$ query of $\mathcal{P}_{\text{uu}}$ by the challenge returned by its only Ch oracle on the transcript commitment Y_l. The analysis is straightforward because there are no restrictions on winning in either case.

The remaining two implications are not of importance for our results and given for completeness. We omit the game-based analysis required to make the proofs rigorous. For part **3.** the adversary $\mathcal{P}_{\text{cc}}$ is on the left in Fig. 6. When $\mathcal{P}_{\text{uc}}$ returns (k, z), adversary $\mathcal{P}_{\text{cc}}$ calls its CH oracle on the corresponding commitment and challenge. It will win its game unless the pair is not new, meaning is in the set S, which can happen with at most the probability that some commitment is returned more than once by TR. The probability of the latter is at most $q_t(q_t - 1)/2^{\mathsf{ID.CSS}}$.

For part **4.** the adversary $\mathcal{P}_{\text{cc}}$ is on the right in Fig. 6. Again it will win its game unless the pair is not new, meaning is in the set S. This can happen with at most the probability that a challenge chosen by CH^* is the same as one in a transcript returned by TR. The probability of this is at most $q_t q_c \cdot 2^{-\mathsf{ID.cl}}$. ∎

SEPARATIONS. We now discuss the separations, beginning with CIMP-CU $\not\Longrightarrow$ CIMP-UC. Start with any CIMP-CU scheme. We will modify it so that it remains CIMP-CU-secure but is not CIMP-UC-secure. Distinguish a single challenge $c^* \in \{0,1\}^{\mathsf{ID.cl}}$, e.g., $c^* = 0^{\mathsf{ID.cl}}$. Revise the verifier's algorithm so that it will accept any transcript with challenge c^*. This is still CIMP-CU-secure (as long as $\mathsf{ID.cl}$ is large) since, in the CIMP-CU game, challenges are picked uniformly at random for the adversary, so existence of the magic challenge is unlikely to be useful. This is manifestly not CIMP-UC-secure since there the adversary can use any challenge of its choice. CIMP-UU $\not\Longrightarrow$ CIMP-UC for the same reason.

We turn to CIMP-UC $\not\Longrightarrow$ CIMP-CU . Start with any CIMP-UC scheme. Again we will modify it so that it remains CIMP-UC-secure but is not CIMP-CU-secure. This time, distinguish a single commitment Y^*: one way of doing this is for $\mathsf{ID.Kg}$ to sample $Y^* \leftarrow\!\!\$\ \mathsf{ID.CS}(ivk)$ and include Y^* in the public key ivk; another is to agree for example that $(Y^*, y^*) \leftarrow \mathsf{ID.Ct}(ivk; 0^l)$ where l is the number of random bits required by $\mathsf{ID.Ct}$. Revise the verifier's algorithm so that it will accept any transcript with commitment Y^*. This is still CIMP-UC-secure (assuming $|\mathsf{ID.CS}(ivk)|$ is large) since, in the CIMP-UC game, commitments are generated randomly for the adversary, so existence of a magic commitment is unlikely to be useful. This is manifestly not CIMP-CU-secure since there the adversary can use any commitment of its choice. CIMP-UU $\not\Longrightarrow$ CIMP-CU for the same reason.

Finally, CIMP-UC $\not\Longrightarrow$ CIMP-CC and CIMP-CU $\not\Longrightarrow$ CIMP-CC since otherwise, by transitivity in Fig. 1, we would contradict the separation between CIMP-UC and CIMP-CU.

4 Achieving CIMP-XY security

Here we show how to obtain identification schemes satisfying our CIMP-XY notions of security. We base CIMP-XY security on the problem of recovering the secret key of the identification scheme given nothing but the public key, which plays the role of the algebraic problem P_{alg} in typical identification schemes and corresponds to a standard assumption. (For example for GQ it is one-wayness of RSA.) For CIMP-UC and CIMP-UU, the reductions are tight. For CIMP-CU, the reduction is not tight. CIMP-CC cannot be obtained via these paths, and instead we establish it from signatures. First we need to recall a few standard definitions.

HVZK AND EXTRACTABILITY. We say that an identification scheme ID is *honest verifier zero-knowledge* (HVZK) if there exists an algorithm $\mathsf{ID.Sim}$ (called the simulator) that given the verification key, generates transcripts which have the same distribution as honest ones, even given the verification key. Formally, if $\mathcal{A}$ is an adversary, let $\mathbf{Adv}^{\text{zk}}_{\mathsf{ID}}(\mathcal{A}) = 2\Pr[\mathbf{G}^{\text{zk}}_{\mathsf{ID}}(\mathcal{A})] - 1$ where the game is shown in Fig. 7. Then ID is HVZK if $\mathbf{Adv}^{\text{zk}}_{\mathsf{ID}}(\mathcal{A}) = 0$ for all adversaries $\mathcal{A}$. (Regardless of the running time of $\mathcal{A}$.) We say that an identification scheme ID is *extractable* if there exists an algorithm $\mathsf{ID.Ex}$ (called the extractor) which from any two transcripts that have the same commitment but different

Game $\mathbf{G}^{\mathrm{ex}}_{\mathsf{ID}}(\mathcal{A})$

$(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$
$(Y, c_1, z_1, c_2, z_2) \leftarrow\!\!\$\ \mathcal{A}(ivk, isk, itk)$
$v_1 \leftarrow \mathsf{ID.Vf}(ivk, Y\|c_1\|z_1)$
$v_2 \leftarrow \mathsf{ID.Vf}(ivk, Y\|c_2\|z_2)$
$isk^* \leftarrow\!\!\$\ \mathsf{ID.Ex}(ivk, Y\|c_1\|z_1, Y\|c_2\|z_2)$
Return $(isk^* \neq isk) \wedge (c_1 \neq c_2) \wedge v_1 \wedge v_2$

Game $\mathbf{G}^{\mathrm{zk}}_{\mathsf{ID}}(\mathcal{A})$

$(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}\ ;\ b \leftarrow \{0,1\}$
$(Y_1, y) \leftarrow\!\!\$\ \mathsf{ID.Ct}(ivk)\ ;\ c_1 \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$
$z_1 \leftarrow \mathsf{ID.Rp}(ivk, isk, c_1, y)$
$Y_0\|c_0\|z_0 \leftarrow\!\!\$\ \mathsf{ID.Sim}(ivk)$
$b' \leftarrow\!\!\$\ \mathcal{A}(ivk, Y_b\|c_b\|z_b)$
Return $(b = b')$

Game $\mathbf{G}^{\mathrm{kr}}_{\mathsf{ID}}(\mathcal{I})$

$(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$
$isk^* \leftarrow\!\!\$\ \mathcal{I}(ivk)$
Return $(isk^* = isk)$

Figure 7: Games defining the extractability, HVZK and key-recovery security of an identification scheme ID.

Adversary $\mathcal{I}(ivk)$

$(k, z) \leftarrow\!\!\$\ \mathcal{P}^{\mathrm{Tr},\mathrm{Ch}}(ivk)$
If not $(1 \le k \le j)$: Return $\perp$
$T \leftarrow \mathrm{CT}[k]\|z\ ;\ l \leftarrow \mathrm{Ind}(k)$
$isk^* \leftarrow \mathsf{ID.Ex}(ivk, Y_l\|c_l\|z_l, T)$
Return isk^*

$\mathrm{Tr}()$

$i \leftarrow i + 1$
$Y_i\|c_i\|z_i \leftarrow\!\!\$\ \mathsf{ID.Sim}(ivk)$
Return $Y_i\|c_i\|z_i$

$\mathrm{Ch}(l, c)$

If not $(1 \le l \le i)$: Return $\perp$
If $(c = c_l)$: Return $\perp$
$j \leftarrow j + 1\ ;\ \mathrm{CT}[j] \leftarrow Y_l\|c\ ;\ \mathrm{Ind}(j) \leftarrow l$
Return c

Figure 8: Games and adversary for proof of Theorem 2.

challenges can recover the secret key. Formally, if $\mathcal{A}$ is an adversary, let $\mathbf{Adv}^{\mathrm{ex}}_{\mathsf{ID}}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{ex}}_{\mathsf{ID}}(\mathcal{A})]$ where the game is shown in Fig. 7. Then ID is extractable if $\mathbf{Adv}^{\mathrm{ex}}_{\mathsf{ID}}(\mathcal{A}) = 0$ for all adversaries $\mathcal{A}$. (Regardless of the running time of $\mathcal{A}$.) We say that an identification scheme is a Sigma protocol [14] if it is both HVZK and extractable.

Security against key recovery. An identification scheme ID is resilient to *key recovery* if it is hard to recover the secret key given nothing but the verification key. This was defined by OO [32]. Formally, if $\mathcal{I}$ is an adversary, let $\mathbf{Adv}^{\mathrm{kr}}_{\mathsf{ID}}(\mathcal{I}) = \Pr[\mathbf{G}^{\mathrm{kr}}_{\mathsf{ID}}(\mathcal{I})]$ where the game is shown in Fig. 7. Security against key recovery is precisely the (standard) assumption $\mathrm{P}_{\mathrm{alg}}$ underlying most identification schemes. For example, for the GQ identification scheme, this is the assumption that RSA is one-way. For factoring-based schemes, it is the factoring assumption. Thus reducing the key recovery security is precisely reducing to $\mathrm{P}_{\mathrm{alg}}$ as desired.

Obtaining CIMP-UU and CIMP-UC. Here we show that for Sigma protocols, CIMP-UU and CIMP-UC security reduce tightly to security under key recovery.

Theorem 2 *Let* ID *be an identification scheme that is honest verifier zero-knowledge and extractable. Then for any adversary* $\mathcal{P}$ *against* CIMP-UC *we construct a key recovery adversary* $\mathcal{I}$ *such that*

$$\mathbf{Adv}^{\text{cimp-uc}}_{\mathsf{ID}}(\mathcal{P}) \leq \mathbf{Adv}^{\text{kr}}_{\mathsf{ID}}(\mathcal{I}) \ . \tag{1}$$

Also for any adversary $\mathcal{P}$ *against* CIMP-UU *that makes* q_c *queries to its* Ch *oracle we construct a key recovery adversary* $\mathcal{I}$ *such that*

$$\mathbf{Adv}^{\text{cimp-uu}}_{\mathsf{ID}}(\mathcal{P}) \leq \mathbf{Adv}^{\text{kr}}_{\mathsf{ID}}(\mathcal{I}) + q_c \cdot 2^{-\mathsf{ID.cl}} \ . \tag{2}$$

In both cases, the running time of $\mathcal{I}$ *is about that of* $\mathcal{P}$ *plus the time for one execution of* $\mathsf{ID.Ex}$ *and the time for a number of executions of* $\mathsf{ID.Sim}$ *equal to the number of* Tr *queries of* $\mathcal{P}$*.*

Proof of Theorem 2: Adversary $\mathcal{I}$ for the first claim is shown in Fig. 8. It response to Tr queries of $\mathcal{P}$ via the simulator. In a $\text{Ch}(l, c)$ query, the game ensures that $c \neq c_l$, so $\mathcal{P}$'s output immediately allows the extractor to obtain the secret key. The second claim follows from the first and Part **1.** of Proposition 1. ▌

Obtaining CIMP-CU. CIMP-CU security of Sigma protocols can also be established under their key recovery security, but the reduction is not tight.

Theorem 3 *Let* ID *be an identification scheme that is honest verifier zero-knowledge and extractable. For any adversary* $\mathcal{P}$ *against* CIMP-CU *making* q *queries to its* Ch *oracle, we construct a key recovery adversary* $\mathcal{I}$ *such that*

$$\mathbf{Adv}^{\text{cimp-cu}}_{\mathsf{ID}}(\mathcal{P}) \leq q \left(\sqrt{\mathbf{Adv}^{\text{kr}}_{\mathsf{ID}}(\mathcal{I})} + \frac{1}{2^{\mathsf{ID.cl}}} \right) \ . \tag{3}$$

The running time of $\mathcal{I}$ *is about twice that of* $\mathcal{P}$*.*

To establish Theorem 3, our route will be via standard techniques and known results, and the proof can be found for completeness in Appendix A. Lemma 12 relates CIMP-CU security of ID to IMP-PA, the standard security against impersonation under passive attack as formalized in [1]. (IMP-PA is exactly CIMP-CU restricted to adversaries that make only one Ch query.) Lemma 13 relates IMP-PA security to key recovery using the reset lemma of [6] and the extractability property of ID. Together these yield CIMP-CU security.

Obtaining CIMP-CC. This is our strongest notion, and is quite different from the rest. Sigma protocols will fail to achieve CIMP-CC because an HVZK identification scheme cannot be CIMP-CC-secure. The attack (adversary) $\mathcal{P}$ showing this is as follows. Assuming ID is HVZK, our adversary $\mathcal{P}$, given the verification key ivk, runs the simulator to get a transcript $Y \| c \| z \leftarrow\!\!\$\ \mathsf{ID.Sim}(ivk)$. It makes no Tr queries, so the set S in the game is empty. It then makes query $\text{Ch}(Y, c)$ and returns $(1, z)$ to achieve $\mathbf{Adv}^{\text{cimp-cc}}_{\mathsf{ID}}(\mathcal{P}) = 1$.

This doesn't mean CIMP-CC is unachievable. We now show how to achieve it. Our construction is from a UF digital signature scheme DS. Associate to DS and a challenge length l the identification scheme ID defined as follows. It has challenge length $\mathsf{ID.cl} = l$. Key generation algorithm $\mathsf{ID.Kg}$ lets $(vk, sk) \leftarrow\!\!\$\ \mathsf{DS.Kg}$ and returns (vk, sk, ε). The commitment is empty, meaning $\mathsf{ID.Ct}(ivk)$ returns $(\varepsilon, \varepsilon)$. The response is a signature of the challenge, meaning $\mathsf{ID.Rp}(vk, sk, c, \varepsilon)$ returns $z \leftarrow\!\!\$\ \mathsf{DS.Sig}(vk, sk, c)$. This identification scheme is trivially trapdoor, $\mathsf{ID.Ct}^{-1}(vk, \varepsilon, \varepsilon)$ returning ε. It is CIMP-CC assuming DS is UF and the challenge length l is large enough, with a tight

$\underline{\mathsf{DS}^*.\mathsf{Kg}}$
Return $\mathsf{DS}.\mathsf{Kg}$

$\underline{\mathsf{DS}^*.\mathsf{Vf}(vk, m, \sigma)}$
Return $\mathsf{DS}.\mathsf{Vf}(vk, m, \sigma)$

$\underline{\mathsf{DS}^*.\mathsf{Sig}^{\mathrm{H}}(vk, sk, m)}$
$r \leftarrow \mathrm{H}(sk \| m)$
Return $\mathsf{DS}.\mathsf{Sig}(vk, sk, m; r)$

$\underline{\mathsf{DS}^*.\mathsf{Kg}}$
Return $\mathsf{DS}.\mathsf{Kg}$

$\underline{\mathsf{DS}^*.\mathsf{Vf}(vk, m, \sigma^*)}$
$\sigma \| s \leftarrow \sigma^*$
Return $\mathsf{DS}.\mathsf{Vf}(vk, m \| s, \sigma')$

$\underline{\mathsf{DS}^*.\mathsf{Sig}(vk, sk, m)}$
$s \leftarrow\!\!\$\ \{0,1\}^{\mathsf{sl}}$
$\sigma \leftarrow\!\!\$\ \mathsf{DS}.\mathsf{Sig}(vk, sk, m \| s)$
$\sigma^* \leftarrow \sigma \| s$; Return σ^*

Figure 9: **Left:** Our construction of deterministic signature scheme $\mathsf{DS}^* = \mathbf{DR}[\mathsf{DS}]$ from a signature scheme DS. By $\mathrm{H}(\cdot)$ we denote $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{DS.rl}})$, which has range $\{0,1\}^{\mathsf{DS.rl}}$. **Right:** Our construction of added-randomness signature scheme $\mathsf{DS}^* = \mathbf{AR}[\mathsf{DS}, \mathsf{sl}]$ from a signature scheme DS and a seed length $\mathsf{sl} \in \mathbb{N}$.

reduction. More precisely, given adversary $\mathcal{P}$ making q_t queries to its Tr oracle and q_c queries to its Ch oracle, we can build adversary $\mathcal{A}$ such that $\mathbf{Adv}^{\text{cimp-cc}}_{\mathsf{ID}}(\mathcal{P}) \leq \mathbf{Adv}^{\text{uf}}_{\mathsf{DS}}(\mathcal{A}) + q_t q_c \cdot 2^{-l}$. Adversary $\mathcal{A}$ makes q_t queries to its Sign oracle and has about the same running time as $\mathcal{P}$.

While this shows CIMP-CC is achievable, and even under standard assumptions, it is not of help for us, since we want to obtain signature schemes from identification schemes and if the latter are themselves built from a signature scheme then nothing has been gained. We consider CIMP-CC nonetheless because our framework naturally gives rise to it and we wish to see the full picture, and also because there may be other ways to achieve CIMP-CC.

5 From UUF to UF

Some of our transforms of identification schemes into signature schemes naturally achieve UUF security rather than UF security. To achieve the latter, one can take our UUF schemes and apply the transforms in this Section to turn them into UF schemes. The reductions are tight and the costs are low. First we observe that standard derandomization (removing randomness) has the additional benefit (apparently not noted before) of turning UUF into UF. Second, we show that message randomization (adding randomness) is also a natural solution.

5.1 From UUF to UF by removing randomness

It is standard to derandomize a signing algorithm by obtaining the coins as a hash of the message and a secret key. This has been shown to preserve UF security —meaning, if the starting scheme is UF-secure, so is the derandomized scheme— in some cases. This is true if one uses a PRF as the hash function with the PRF key added to the signing secret key [31], but this changes the signing key, which can be undesirable in practice. Instead one can hash the signing key with the message, modeling the hash function as a random oracle. This has been proven to work for certain particular choices of the starting signature scheme, namely when this scheme is ECDSA [25]. Such de-randomization is used in the Ed25519 signature scheme [11]. However, it has not been proven in the general case. This will follow from our results.

The purpose of the method, above, was exactly to derandomize, namely to ensure that the

signing process is deterministic, and the starting signature scheme was assumed UF-secure. We observe here that the method has an additional benefit which does not seem to have been noted before, namely that it works even if the starting scheme is only UUF secure, meaning it upgrades UUF security to UF security. It is an attractive way to do this because it preserves signature size and verification time, while adding to the signing time only the cost of one hash. We specify a derandomization transform and prove that that it turns UUF schemes into UF ones in general, meaning assuming nothing more than UUF security of the starting scheme. In particular, we justify derandomization in a broader context than previous work.

THE CONSTRUCTION. For a signature scheme DS, let $\mathsf{DS.rl}$ denote the length of the randomness (number of coins) used by the signing algorithm $\mathsf{DS.Sig}$. We write $\sigma \leftarrow \mathsf{DS.Sig}(vk, sk, m; r)$ for the execution of $\mathsf{DS.Sig}$ on inputs vk, skm and coins $r \in \{0,1\}^{\mathsf{DS.rl}}$. Let signature scheme $\mathsf{DS}^* = \mathbf{DR}[\mathsf{DS}]$ be obtained from DS as in Fig. 9. Here, the $\mathrm{H}(\cdot)$ used to compute r in algorithm $\mathsf{DS}^*.\mathsf{Sig}$ is $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{DS.rl}})$, meaning the range is set to $\{0,1\}^{\mathsf{DS.rl}}$.

While algorithms of the starting scheme DS may invoke the random oracle (and, in the schemes we construct in Section 6, they do), it is assumed they do not invoke $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{DS.rl}})$. This can be ensured in a particular case by domain separation. Given this, other calls of the algorithms of the starting scheme to the random oracle can be simulated directly in the proof via the random oracle available to the constructed adversaries. Accordingly in the scheme description of Fig. 9, and proof below, for simplicity, we do not give the algorithms of the starting signature scheme access to the random oracle. That is, think of the starting scheme as being a standard-model one.

UNFORGEABILITY. The following says that the constructed scheme DS^* is UF-secure assuming the starting scheme DS was UUF secure, with a tight reduction. The reason a deterministic scheme that is UUF is also UF is clear, namely there is nothing to gain by calling the signing oracle more than once on a particular message, because one just gets back the same thing each time. What the proof needs to ensure is that the method of making the scheme deterministic does not create any weaknesses. The danger is that including the secret key as an input to the hash increases the exposure of the key. The proof says that it might a little, but the advantage does not go up by more than a factor of two.

Theorem 4 *Let signature scheme* $\mathsf{DS}^* = \mathbf{DR}[\mathsf{DS}]$ *be obtained from signature scheme* DS *as in Fig. 9. Let* $\overline{\mathcal{A}}$ *be a* UF*-adversary against* DS^* *that makes* q_h *queries to* H *and* q_s *queries to* SIGN*. Then from* $\mathcal{A}$ *we can construct* UUF*-adversary* $\mathcal{A}$ *such that*

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}^*}(\overline{\mathcal{A}}) \leq 2 \cdot \mathbf{Adv}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A}) \ . \tag{4}$$

Adversary $\mathcal{A}$ *makes* q_s *queries to* SIGN*. It has running time about that of* $\overline{\mathcal{A}}$ *plus the time for* q_h *invocations of* $\mathsf{DS.Sig}$ *and* $\mathsf{DS.Vf}$*.*

Proof: Game G_0 of Fig. 10 includes the boxed code, while game G_1 does not. We assume (wlog) that $\overline{\mathcal{A}}$ does not repeat a query to its H oracle. We assume the message m in the forgery (m, σ) returned by $\overline{\mathcal{A}}$ was not queried to SIGN. Game G_0 is equivalent to the UF game of Fig. 2 with the algorithms of DS^* plugged in. Games $\mathrm{G}_0, \mathrm{G}_1$ are identical until bad. By the Fundamental Lemma of Game Playing [10] we have

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}^*}(\overline{\mathcal{A}}) = \Pr[\mathrm{G}_0] = \Pr[\mathrm{G}_1] + (\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]) \leq \Pr[\mathrm{G}_1] + \Pr[\mathrm{G}_1 \text{ sets } \mathsf{bad}] \ .$$

We construct adversaries $\mathcal{A}_0, \mathcal{A}_1$ such that

$$\Pr[\mathrm{G}_1] \leq \mathbf{Adv}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A}_1) \quad \text{and} \quad \Pr[\mathrm{G}_1 \text{ sets } \mathsf{bad}] \leq \mathbf{Adv}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A}_0) \ .$$

Game $\boxed{\mathrm{G}_0}$, G_1

$M \leftarrow \emptyset$
$(vk, sk) \leftarrow\!\!\$\ \mathsf{DS.Kg}$
$(m, \sigma) \leftarrow\!\!\$\ \overline{\mathcal{A}}^{\textsc{Sign},\mathrm{H}}(vk)$
Return $\mathsf{DS.Vf}(vk, m, \sigma)$

$\textsc{Sign}(m)$
If $m \in M$: Return $\mathrm{ST}[m]$
$r \leftarrow\!\!\$\ \{0,1\}^{\mathsf{DS.rl}}$
If $\mathrm{HT}[sk, m]$:
  $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{r \leftarrow \mathrm{HT}[sk, m]}$
$\mathrm{HT}[sk, m] \leftarrow r$
$\sigma \leftarrow \mathsf{DS.Sig}(vk, sk, m; r)$
$\mathrm{ST}[m] \leftarrow \sigma$; $M \leftarrow M \cup \{m\}$
Return σ

$\mathrm{H}(x)$
$sk^* \| m \leftarrow x$; $r \leftarrow\!\!\$\ \{0,1\}^{\mathsf{DS.rl}}$
If $\mathrm{HT}[sk^*, m]$:
  $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{r \leftarrow \mathrm{HT}[sk^*, m]}$
$\mathrm{HT}[sk^*, m] \leftarrow r$
Return $\mathrm{HT}[sk^*, m]$

Adversary $\mathcal{A}_0^{\textsc{Sign}}(vk)$

$S \leftarrow \emptyset$; $(m, \sigma) \leftarrow\!\!\$\ \overline{\mathcal{A}}^{\textsc{Sign}^*,\mathrm{H}^*}(vk)$
Pick some $m^* \notin M$
For $sk^* \in S$ do
  $\sigma^* \leftarrow\!\!\$\ \mathsf{DS.Sig}(vk, sk^*, m^*)$
  If $\mathsf{DS.Vf}(vk, m^*, \sigma^*)$: Return (m^*, σ^*)
Return $\perp$

Adversary $\mathcal{A}_1^{\textsc{Sign}}(vk)$

$S \leftarrow \emptyset$; $(m, \sigma) \leftarrow\!\!\$\ \overline{\mathcal{A}}^{\textsc{Sign}^*,\mathrm{H}^*}(vk)$
Return (m, σ)

$\textsc{Sign}^*(m)$
If $m \notin M$: $\mathrm{ST}[m] \leftarrow\!\!\$\ \textsc{Sign}(m)$
Return $\mathrm{ST}[m]$

$\mathrm{H}^*(x)$
$sk^* \| m \leftarrow x$; $S \leftarrow S \cup \{sk^*\}$
$\mathrm{HT}[sk^*, m] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{DS.rl}}$
Return $\mathrm{HT}[sk^*, m]$

Figure 10: Games and adversaries for proof of Theorem 4.

The adversaries are shown in Fig. 10. They have access to a SIGN oracle for the DS scheme. They do not have access to H because of our simplifying assumption that DS is a standard-model scheme. They execute $\overline{\mathcal{A}}$, responding to H and SIGN queries of the latter via the shown procedures H^* and $\textsc{Sign}^*$, respectively. These procedures are subroutines in the code of $\mathcal{A}_0$ and $\mathcal{A}_1$, and both incorporate and use them. Procedure $\textsc{Sign}^*$ ensures that the adversaries are uuf, meaning do not submit a particular message twice to their own SIGN oracle that is invoked by $\textsc{Sign}^*$. Adversary $\mathcal{A}_1$ simulates for $\overline{\mathcal{A}}$ the environment of game G_1 and returns the forgery that $\overline{\mathcal{A}}$ returns. Adversary $\mathcal{A}_0$ also simulates for $\overline{\mathcal{A}}$ the environment of game G_1. The code of H^* records all candidate secret keys. At the end, $\mathcal{A}_0$ attempts a forgery under each of them, returning one that is successful. In G_1, we claim that if bad is set, either by SIGN or by H, it must be that there was a H query of the form $sk \| m$ for some m. To justify this, consider two cases. First, if bad is set by SIGN, it could only be because of a prior H query of the form $sk \| m$. Second, consider bad being set by H. We have assumed that $\overline{\mathcal{A}}$ does not repeat a query to H. Thus, if a query $sk^* \| m$ to H sets bad, it must be that $\mathrm{HT}[sk^*, m]$ was defined by SIGN, in which case it must be that $sk^* = sk$. Thus if G_1 sets bad then $sk \in S$, so $\mathcal{A}_0$ is successful. Finally we define $\mathcal{A}^{\textsc{Sign}}(vk)$ to pick a random bit $c \leftarrow\!\!\$\ \{0,1\}$ and return $\mathcal{A}_c^{\textsc{Sign}}(vk)$. This means

$$\mathbf{Adv}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A}_0) + \mathbf{Adv}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A}_1) = 2 \cdot \mathbf{Adv}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A}) \ ,$$

which completes the proof. ∎

We remark that adversary $\mathcal{A}_0$ actually violates key recovery security of DS, not just its UUF security.

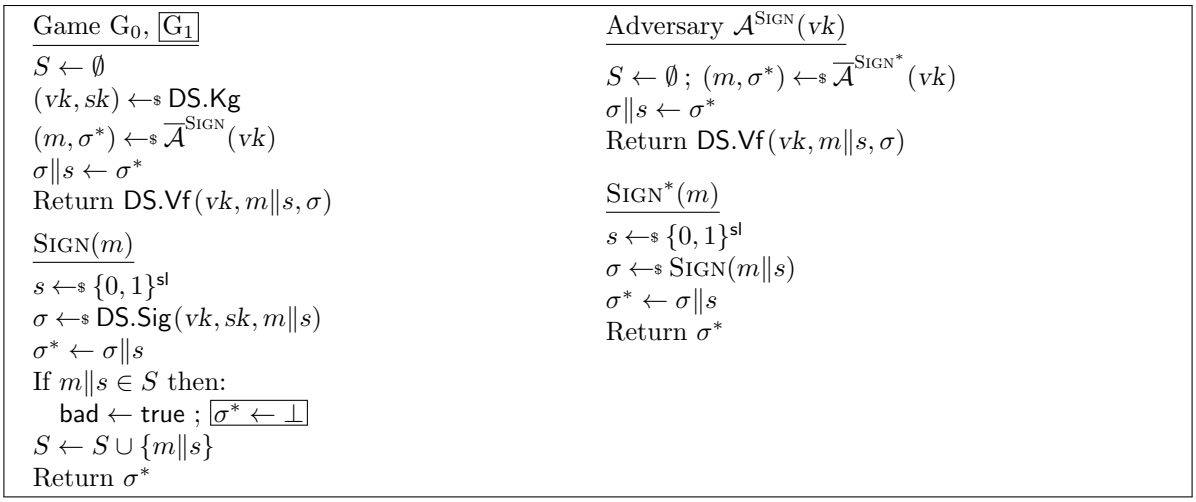


Figure 11: Games and adversaries for proof of Theorem 5.

5.2 From UUF to UF by adding randomness

A complementary and natural method for constructing UF signatures from UUF ones is by adding randomness: before being signed, the message is concatenated with a random seed s, so even for the same message, the inputs to the UUF signing algorithm are (with high probability) distinct. Compared to derandomization, the drawback of this method is that the signature size increases because the seed must be included in the signature. The potential advantage is that the transform is standard model, not using a random oracle, while preserving the secret key. (Derandomization can be done in the standard model via a PRF, but this requires augmenting the signing key with the PRF key.)

THE CONSTRUCTION. Let signature scheme $\mathsf{DS}^* = \mathbf{DR}[\mathsf{DS}]$ be obtained from DS as in Fig. 9. As above, DS is for simplicity assumed to be a standard-model scheme, so that its algorithms do not have access to H. The transform itself does not use H.

UNFORGEABILITY. The following says that the constructed scheme DS^* is UF-secure assuming the starting scheme DS was UUF secure, with a tight reduction. The reason is quite simple, namely that unless seeds collide, the messages being signed are distinct.

Theorem 5 *Let signature scheme $\mathsf{DS}^* = \mathbf{AR}[\mathsf{DS}, \mathsf{sl}]$ be obtained from signature scheme DS and seed length $\mathsf{sl} \in \mathbb{N}$ as in Fig. 9. Let $\overline{\mathcal{A}}$ be a* UF*-adversary against DS^* making q_s queries to its* SIGN *oracle. Then from $\mathcal{A}$ we construct a* UUF *adversary $\mathcal{A}$ such that*

$$\mathbf{Adv}^{\text{uf}}_{\mathsf{DS}^*}(\mathcal{A}) \leq \mathbf{Adv}^{\text{uuf}}_{\mathsf{DS}}(\mathcal{A}) + \frac{q_s^2}{2^{\mathsf{sl}+1}} \ .$$

Adversary $\mathcal{A}$ makes q_s queries to its SIGN *oracle and has about the same running time as $\overline{\mathcal{A}}$.*

Proof of Theorem 5: Game G_1 of Fig. 11 includes the boxed code, while game G_0 does not. Game G_0 is equivalent to the UF game of Fig. 2 with the algorithms of DS^* plugged in. We assume the message m in the forgery (m, σ) returned by $\overline{\mathcal{A}}$ was not queried to SIGN. Games G_0, G_1 are

$\underline{\mathsf{DS.Kg}^{\mathrm{H}}}$
$(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$
$vk \leftarrow ivk\ ;\ sk \leftarrow (isk, itk)$
Return (vk, sk)

$\underline{\mathsf{DS.Vf}^{\mathrm{H}}(vk, m, \sigma)}$
$ivk \leftarrow vk\ ;\ (c, z) \leftarrow \sigma$
$Y \leftarrow \mathrm{H}(m)$
Return $\mathsf{ID.Vf}(ivk, Y \| c \| z)$

$\underline{\mathsf{DS.Sig}^{\mathrm{H}}(vk, sk, m)}$
$ivk \leftarrow vk\ ;\ (isk, itk) \leftarrow sk$
$Y \leftarrow \mathrm{H}(m)$
$y \leftarrow\!\!\$\ \mathsf{ID.Ct}^{-1}(ivk, itk, Y)$
$c \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$
$z \leftarrow \mathsf{ID.Rp}(ivk, isk, c, y)$
$\sigma \leftarrow (c, z)$
Return σ

Figure 12: The construction of signature scheme $\mathsf{DS} = \mathbf{MdCmt}[\mathsf{ID}]$ from trapdoor identification scheme ID. By $\mathrm{H}(\cdot)$ we denote $\mathrm{H}(\cdot, \mathsf{ID.CS}(ivk))$.

identical until bad. By the Fundamental Lemma of Game Playing [10] we have

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}^*}(\overline{\mathcal{A}}) = \Pr[\mathrm{G}_0] = \Pr[\mathrm{G}_1] + (\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]) \le \Pr[\mathrm{G}_1] + \Pr[\mathrm{G}_0 \text{ sets } \mathsf{bad}]$$
$$\le \Pr[\mathrm{G}_1] + \frac{q_s^2}{2^{\mathsf{sl}+1}}\ .$$

We construct adversary $\mathcal{A}$ such that

$$\Pr[\mathrm{G}_1] \le \mathbf{Adv}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A})\ .$$

Adversary $\mathcal{A}$ is shown in Fig. 11. It has access to a Sign oracle for the DS scheme. It executes $\overline{\mathcal{A}}$, responding to Sign queries of the latter via the shown procedure Sign*. This procedure is a subroutine in the code of $\mathcal{A}$. In the execution of game $\mathbf{G}^{\mathrm{uuf}}_{\mathsf{DS}}(\mathcal{A})$, if $\mathcal{A}$ repeats a query to Sign, the reply will be $\perp$. This means that $\mathcal{A}$ is providing $\overline{\mathcal{A}}$ the environment of G_1. ▌

6 Signatures from identification

We specify our three new transforms of identification schemes to signature schemes, namely the ones of rows 2,3,4 of the table of Fig. 1. In each case, we give a security proof based on the assumption P_{id} listed in the 1st column of the corresponding row of the table, so that we give transforms from CIMP-UC, CIMP-UU and CIMP-CC. It turns out that these transforms naturally achieve UUF rather than UF, and this is what we prove, with tight reductions of course. The transformation UF→UUF can be done at the level of signatures, not referring to identification, in generic and simple ways, and also with tight reductions, as detailed in Section 5. We thus get UF-secure signatures with tight reductions to each of CIMP-UC, CIMP-UU and CIMP-CC.

6.1 From CIMP-UC identification to UUF signatures: MdCmt

MdCmt transforms a CIMP-UC trapdoor identification scheme to a UUF signature scheme using message-dependent commitments.

The construction. Let ID be a trapdoor identification scheme and ID.cl its challenge length. Our **MdCmt** (message-dependent commitment) transform associates to ID the signature scheme $\mathsf{DS} = \mathbf{MdCmt}[\mathsf{ID}]$. The algorithms of DS are defined in Fig. 12. By $\mathrm{H}(\cdot)$ we denote $\mathrm{H}(\cdot, \mathsf{ID.CS}(ivk))$, meaning the range is set to $\mathsf{ID.CS}(ivk)$. Signatures are effectively identification transcripts, but the commitments are chosen in a particular way. Recall that with trapdoor ID schemes it is the same whether one executes $(Y, y) \leftarrow\!\!\$\ \mathsf{ID.Ct}$ directly, or samples $Y \leftarrow\!\!\$\ \mathsf{ID.CS}(ivk)$ followed by computing

Adversary $\mathcal{P}^{\text{Tr},\text{Ch}}(ivk)$	$\underline{\text{Sign}(m)}$ // $\mathcal{P}$
$M \leftarrow \emptyset$; $i \leftarrow 0$	If $m \in M$: Return $\perp$
$(m, \sigma) \leftarrow\!\!\$\ \mathcal{A}^{\text{Sign},\text{H}}(ivk)$	$Y \leftarrow \text{H}(m)$
$Y \leftarrow \text{H}(m)$	$l \leftarrow \text{ST}[m]$; $\sigma \leftarrow (c_l, z_l)$
$(c, z) \leftarrow \sigma$; $l \leftarrow \text{ST}[m]$	$M \leftarrow M \cup \{m\}$; Return σ
$c' \leftarrow \text{Ch}(l, c)$ // $c' = c$	$\underline{\text{H}(m)}$ // $\mathcal{P}$
Return $(1, z)$	If $\text{HT}[m]$: Return $\text{HT}[m]$
	$i \leftarrow i + 1$; $\text{ST}[m] \leftarrow i$
	$Y_i \| c_i \| z_i \leftarrow\!\!\$\ \text{Tr}()$
	$\text{HT}[m] \leftarrow Y_i$
	Return $\text{HT}[m]$

Figure 13: CIMP-UC adversary for proof of Theorem 6.

$y \leftarrow\!\!\$\ \mathsf{ID.Ct}^{-1}(Y)$. Our construction exploits this: To each message m it assigns an individual commitment $Y \leftarrow \text{H}(m)$. The signing algorithm, using the trapdoor, completes this commitment to a transcript (Y, c, z) and outputs the pair c, z as the signature. Verification then consists of recomputing Y from m and invoking the verification algorithm of the ID scheme.

Unforgeability. The following theorem establishes that the (unique) unforgeability of a signature scheme constructed with **MdCmt** tightly reduces to the CIMP-UC security of the underlying ID scheme, in the random oracle model.

Theorem 6 *Let signature scheme* $\mathsf{DS} = \mathbf{MdCmt}[\mathsf{ID}]$ *be obtained from trapdoor identification scheme* ID *as in Fig. 12. Let* $\mathcal{A}$ *be a* UUF*-adversary against* DS*. Suppose the number of queries that* $\mathcal{A}$ *makes to its* H *and* Sign *oracles are* q_h *and* q_s*, respectively. Then from* $\mathcal{A}$ *we construct a* CIMP-UC *adversary* $\mathcal{P}$ *such that*

$$\mathbf{Adv}^{\text{uuf}}_{\mathsf{DS}}(\mathcal{A}) \leq \frac{\mathbf{Adv}^{\text{cimp-uc}}_{\mathsf{ID}}(\mathcal{P})}{1 - 2^{-\mathsf{ID.cl}}} \ . \tag{5}$$

Adversary $\mathcal{P}$ *makes* $q_h + q_s + 1$ *queries to* Tr *and one query to* Ch*. Its running time is about that of* $\mathcal{A}$*.*

The bound of Equation (5) may be a bit hard to estimate. The following simpler bound is also true and may be easier to use:

$$\mathbf{Adv}^{\text{uuf}}_{\mathsf{DS}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{cimp-uc}}_{\mathsf{ID}}(\mathcal{P}) + \frac{1}{2^{\mathsf{ID.cl}}} \ . \tag{6}$$

We now explain why Equation (5) implies Equation (6) and also compare the bounds. For notational ease let $p = \mathbf{Adv}^{\text{cimp-uc}}_{\mathsf{ID}}(\mathcal{P})$ and $\epsilon = 2^{-\mathsf{ID.cl}}$. Then consider two cases. First, if $p \geq 1 - \epsilon$ then the right side of Equation (6) is at least 1 so the equation is trivially true. Now suppose $p \leq 1 - \epsilon$. This means $p/(1 - \epsilon) \leq 1$. Then from Equation (5) we have

$$\mathbf{Adv}^{\text{uuf}}_{\mathsf{DS}}(\mathcal{A}) \leq \frac{p}{1 - \epsilon} = \frac{p(1 - \epsilon) + p\epsilon}{1 - \epsilon} = p + \epsilon \cdot \frac{p}{1 - \epsilon} \leq p + \epsilon \ ,$$

which establishes Equation (6). Now, to compare, although both bounds hold, Equation (5) will be better when p is small, specifically, $p < 1 - \epsilon$. But this is usually the case since otherwise we are not indicating much security.

Proof of Theorem 6: Adversary $\mathcal{P}$ is shown in Fig. 13. It executes $\mathcal{A}$, responding to H and SIGN queries of the latter via the shown procedures, which are subroutines in the code of $\mathcal{P}$. For each H query, it uses its TR oracle to generate a transcript, assigning the commitment from this transcript as the answer to the query. The reason for the "$Y \leftarrow \mathrm{H}(m)$" statements in the code of SIGN and following the execution of $\mathcal{A}$ is to ensure that the value of $\mathrm{H}(m)$ is assigned. We assume the message m in the forgery (m, σ) returned by $\mathcal{A}$ was not queried to SIGN and is not in the set M, since otherwise $\mathcal{A}$ would automatically lose. Adversary $\mathcal{P}$ identifies the transcript index l corresponding to the forgery message, and calls its CH oracle with this l and the challenge c in the forged signature. The CH oracle will assign this session identifier $j = 1$ and now $\mathcal{P}$ can provide the response as per the forged signature. Adversary $\mathcal{A}$ wins if the forgery is successful and $c \neq c_l$. The events being independent we have

$$\mathbf{Adv}^{\text{cimp-uc}}_{\mathsf{ID}}(\mathcal{P}) \geq (1 - 2^{-\mathsf{ID.cl}}) \cdot \mathbf{Adv}^{\text{uuf}}_{\mathsf{DS}}(\mathcal{A}) \ .$$

Transposing terms yields Equation (5). ▌

6.2 From CIMP-UU identification to UUF signatures: MdCmtCh

MdCmtCh transforms a CIMP-UU trapdoor identification scheme to a UUF signature scheme using message-dependent commitments and challenges.

THE CONSTRUCTION. Our **MdCmtCh** (message-dependent commitment and challenge) transform associates to trapdoor identification scheme ID the signature scheme $\mathsf{DS} = \mathbf{MdCmtCh}[\mathsf{ID}]$ whose algorithms are defined in Fig. 14. Here we specify the commitment Y as a hash of the message alone, then use the trapdoor property to allow our signer to obtain $y \leftarrow\!\!\$\ \mathsf{ID.Ct}^{-1}(ivk, itk, Y)$. We then specify the challenge as a randomized hash of the message. (Unlike in the **FS** transform, the commitment is not hashed along with the message.) The randomization is captured by a one-bit seed s. The construction, and proof below, both use the technique of KW [23].

By $\mathrm{H}_1(\cdot)$ we denote random oracle $\mathrm{H}(\cdot, \mathsf{ID.CS}(ivk))$ with range $\mathsf{ID.CS}(ivk)$ and by $\mathrm{H}_2(\cdot)$ we denote random oracle $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{ID.cl}})$ with range $\{0,1\}^{\mathsf{ID.cl}}$. We assume $\mathsf{ID.CS}(ivk) \neq \{0,1\}^{\mathsf{ID.cl}}$ so that these random oracles are independent. In case $\mathsf{ID.CS}(ivk) = \{0,1\}^{\mathsf{ID.cl}}$, the scheme should be modified to use domain separation, for example prefix a 1 to any query to H_1 and a 0 to any query to H_2.

Notice that the signature consists of a response plus a bit. It is thus shorter than for **MdCmt** (where it is a response plus a challenge) or for **FS** (where it is a response plus a commitment or, in the more compact form, a response plus a challenge). These shorter signatures are a nice feature of **MdCmtCh**.

UNFORGEABILITY OF OUR CONSTRUCTION. The following shows that unique unforgeability of our signature tightly reduces to the CIMP-UU security of the underlying ID scheme. Standard unforgeability follows immediately (and tightly) by applying one of the UUF-to-UF transforms in Section 5.

Theorem 7 *Let signature scheme* $\mathsf{DS} = \mathbf{MdCmtCh}[\mathsf{ID}]$ *be obtained from trapdoor identification scheme* ID *as in Fig. 14. Let* $\mathcal{A}$ *be a* UUF*-adversary against* DS*. Suppose the number of queries that* $\mathcal{A}$ *makes to its* H_1 *and* H_2 *oracles is* q_h*, and the number to its* SIGN *oracle is* q_s*. Then from* $\mathcal{A}$ *we can construct* CIMP-UU *adversary* $\mathcal{P}$ *such that*

$$\mathbf{Adv}^{\text{uuf}}_{\mathsf{DS}}(\mathcal{A}) \quad \leq \quad 2 \cdot \mathbf{Adv}^{\text{cimp-uu}}_{\mathsf{ID}}(\mathcal{P}) \ . \tag{7}$$

Adversary $\mathcal{P}$ *makes* $q_h + q_s + 1$ *queries to* TR *and* $q_h + q_s$ *queries to* CH*. It has running time about that of* $\mathcal{A}$*.*

$\underline{\mathsf{DS.Kg}^{\mathrm{H}}}$
$(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$
$vk \leftarrow ivk\ ;\ sk \leftarrow (isk, itk)$
Return (vk, sk)

$\underline{\mathsf{DS.Vf}^{\mathrm{H}}(vk, m, \sigma)}$
$ivk \leftarrow vk\ ;\ (z, s) \leftarrow \sigma$
$Y \leftarrow \mathrm{H}_1(m)$
$c \leftarrow \mathrm{H}_2(m\|s)$
Return $\mathsf{ID.Vf}(ivk, Y\|c\|z)$

$\underline{\mathsf{DS.Sig}^{\mathrm{H}}(vk, sk, m)}$
$s \leftarrow\!\!\$\ \{0,1\}$
$ivk \leftarrow vk\ ;\ (isk, itk) \leftarrow sk$
$Y \leftarrow \mathrm{H}_1(m)$
$y \leftarrow\!\!\$\ \mathsf{ID.Ct}^{-1}(ivk, itk, Y)$
$c \leftarrow \mathrm{H}_2(m\|s)$
$z \leftarrow \mathsf{ID.Rp}(ivk, isk, c, y)$
$\sigma \leftarrow (z, s)$; Return σ

Figure 14: Our construction of signature scheme $\mathsf{DS} = \mathbf{MdCmtCh}[\mathsf{ID}]$ from a trapdoor identification scheme ID. By $\mathrm{H}_1(\cdot)$ we denote random oracle $\mathrm{H}(\cdot, \mathsf{ID.CS}(ivk))$ with range $\mathsf{ID.CS}(ivk)$ and by $\mathrm{H}_2(\cdot)$ we denote random oracle $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{ID.cl}})$ with range $\{0,1\}^{\mathsf{ID.cl}}$.

$\underline{\text{Adversary } \mathcal{P}^{\mathrm{Tr},\mathrm{Ch}}(ivk)}$
$vk \leftarrow ivk\ ;\ i \leftarrow 0\ ;\ M \leftarrow \emptyset$
$(m, \sigma) \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{Sign},\mathrm{H}}(vk)$
$(z, s) \leftarrow \sigma$
$Y \leftarrow \mathrm{H}_1(m)$
$j \leftarrow \mathrm{Ind}(m)$
Return (j, z)

$\underline{\mathrm{Sign}(m)}\ /\!/\ \mathcal{P}$
If $m \in M$: Return $\perp$
$M \leftarrow M \cup \{m\}$
$Y \leftarrow \mathrm{H}_1(m)\ ;\ l \leftarrow \mathrm{Ind}(m)$
$\sigma \leftarrow (z_l, s_l)\ ;$ Return σ

$\underline{\mathrm{H}_1(m)}\ /\!/\ \mathcal{P}$
If $\mathrm{HT}_1[m]$: Return $\mathrm{HT}_1[m]$
$i \leftarrow i + 1\ ;\ Y_i\|c_i\|z_i \leftarrow\!\!\$\ \mathrm{Tr}()$
$\mathrm{HT}_1[m] \leftarrow Y_i\ ;\ \mathrm{Ind}(m) \leftarrow i$
$s_i \leftarrow\!\!\$\ \{0,1\}\ ;\ \mathrm{HT}_2[m\|s_i] \leftarrow c_i$
$\mathrm{HT}_2[m\|\overline{s}_i] \leftarrow\!\!\$\ \mathrm{Ch}(i)$
Return $\mathrm{HT}_1[m]$

$\underline{\mathrm{H}_2(x)}\ /\!/\ \mathcal{P}$
If $\mathrm{HT}_2[x]$: Return $\mathrm{HT}_2[x]$
$m\|s \leftarrow x\ ;\ Y \leftarrow \mathrm{H}_1(m)$
Return $\mathrm{HT}_2[x]$

Figure 15: Adversary for proof of Theorem 7.

Proof of Theorem 7: Adversary $\mathcal{P}$ is shown in Fig. 15. It executes $\mathcal{A}$, responding to H_1, H_2 and Sign queries of the latter via the shown procedures, which are subroutines in the code of $\mathcal{P}$. (Recall that $\mathrm{H}_1(\cdot)$ denotes $\mathrm{H}(\cdot, \mathsf{ID.CS}(ivk))$ and $\mathrm{H}_2(\cdot)$ denotes $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{ID.cl}})$.) We assume the message m in the forgery (m, σ) returned by $\mathcal{A}$ was not queried to Sign and is not in the set M, since otherwise $\mathcal{A}$ would automatically lose. The "$Y \leftarrow \mathrm{H}_1(m)$" instructions in the code of Sign, the code of H_2 and following the execution of $\mathcal{A}$ ensure that $\mathrm{H}_1(m)$ is queried at this point. Each time a new $\mathrm{H}_1(m)$ query is made, a transcript is generated by $\mathcal{P}$ using its Tr oracle. The commitment in this transcript is the reply to the $\mathrm{H}_1(m)$ query. Additionally, however, steps are taken to ensure that, if, later, a $\mathrm{Sign}(m)$ query is made, then a signature to return is available. This is done by picking a random one-bit seed s_i and assigning $\mathrm{H}_2(m\|s_i)$ the value c_i. At the time of a signing query, one can use s_i as the seed and use the response of the corresponding transcript to create the signature. To be able to win via the forgery, $\mathrm{H}_2(m\|\overline{s}_i)$ is assigned a challenge via Ch, where $\overline{s}_i$ denotes the complement of the bit s_i. Now, when the forgery $(m, (z, s))$ is obtained from $\mathcal{A}$, the associated index j is computed, and then z is returned as a response for that session. Adversary $\mathcal{P}$ will be

$\underline{\mathsf{DS.Kg}^{\mathrm{H}}}$	$\underline{\mathsf{DS.Sig}^{\mathrm{H}}(vk, sk, m)}$
$(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$	$ivk \leftarrow vk\ ;\ isk \leftarrow sk$
$vk \leftarrow ivk\ ;\ sk \leftarrow isk$	$s \leftarrow\!\!\$\ \{0,1\}^{\mathsf{sl}}$
Return (vk, sk)	$(Y, y) \leftarrow\!\!\$\ \mathsf{ID.Ct}(ivk)$
$\underline{\mathsf{DS.Vf}^{\mathrm{H}}(vk, m, \sigma)}$	$c \leftarrow \mathrm{H}(m\|s)$
$ivk \leftarrow vk\ ;\ (Y, s, z) \leftarrow \sigma$	$z \leftarrow \mathsf{ID.Rp}(ivk, isk, c, y)$
$c \leftarrow \mathrm{H}(m\|s)$	$\sigma \leftarrow (Y, s, z)$
Return $\mathsf{ID.Vf}(ivk, Y\|c\|z)$	Return σ

Figure 16: The construction of signature scheme $\mathsf{DS} = \mathbf{MdCh}[\mathsf{ID}, \mathsf{sl}]$ from identification scheme ID and seed length sl. By $\mathrm{H}(\cdot)$ we denote random oracle $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{ID.cl}})$ with range $\{0,1\}^{\mathsf{ID.cl}}$.

successful as long as $\mathcal{A}$ is successful and $s = \overline{s}_j$. The events being independent we have

$$\mathbf{Adv}^{\text{cimp-uu}}_{\mathsf{ID}}(\mathcal{P}) \geq \frac{1}{2} \cdot \mathbf{Adv}^{\text{uuf}}_{\mathsf{DS}}(\mathcal{A})\ .$$

Transposing terms yields Equation (7). ▌

6.3 From CIMP-CC identification to UF signatures: MdCh

The **MdCmt** and **MdCmtCh** transforms described above rely on the trapdoor property of the underlying identification scheme and achieve UUF rather than UF. The **MdCh** transform we describe here does not have these limitations. (It does not require the identification scheme to be trapdoor, and it directly achieves UF.) However, amongst the security notions for ID schemes that we defined, **MdCh** assumes the strongest one: CIMP-CC.

The construction. Our **MdCh** (message-dependent challenge) transform associates to identification scheme ID and a seed length $\mathsf{sl} \in \mathbb{N}$ the signature scheme $\mathsf{DS} = \mathbf{MdCh}[\mathsf{ID}, \mathsf{sl}]$ whose algorithms are defined in Fig. 16. Signing picks the commitment directly rather than (as in our prior transforms) specifying it as the hash of the message. The challenge is derived as a randomized hash of the message, the randomization is captured by the seed s whose length sl is a parameter of the transform. By $\mathrm{H}(\cdot)$ we denote random oracle $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{ID.cl}})$ with range $\{0,1\}^{\mathsf{ID.cl}}$.

Unforgeability. As we prove below (with tight reduction), the **MdCh** construction yields a UF-secure signature scheme if the underlying identification scheme offers CIMP-CC security.

Theorem 8 *Let signature scheme* $\mathsf{DS} = \mathbf{MdCh}[\mathsf{ID}, \mathsf{sl}]$ *be obtained from identification scheme* ID *and seed length* $\mathsf{sl} \in \mathbb{N}$ *as in Fig. 16. Let* $\mathcal{A}$ *be a* UF*-adversary against* DS *making* q_h *queries to its* H *oracle and* q_s *queries to its* Sign *oracle. Then from* $\mathcal{A}$ *we construct a* CIMP-CC *adversary* $\mathcal{P}$ *such that*

$$\mathbf{Adv}^{\text{uf}}_{\mathsf{DS}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{cimp-cc}}_{\mathsf{ID}}(\mathcal{P}) + \frac{q_h q_s}{2^{\mathsf{ID.cl}}} + \frac{(q_h + q_s) q_s}{2^{\mathsf{sl}}}\ . \tag{8}$$

Adversary $\mathcal{P}$ *makes* q_s *queries to* Tr *and one query to* Ch *and has running time about that of* $\mathcal{A}$*.*

Proof of Theorem 8: Game G_0 of Fig. 17 includes the boxed code, while game G_1 does not. Game G_0 is precisely the UF game of Fig. 2 with the algorithms of DS plugged in. We assume the message m in the forgery (m, σ) returned by $\mathcal{A}$ was not queried to Sign. Games $\mathrm{G}_0, \mathrm{G}_1$ are

Game $\boxed{\mathrm{G}_0}$, G_1

$(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$
$(m, \sigma) \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Sign},\mathrm{H}}(ivk)$
$(Y, s, z) \leftarrow \sigma$
$c \leftarrow \mathrm{H}(m\|s)$
Return $\mathsf{ID.Vf}(ivk, Y\|c\|z)$

$\textsc{Sign}(m)$

$s \leftarrow\!\!\$\ \{0,1\}^{\mathsf{sl}}$
$(Y, y) \leftarrow\!\!\$\ \mathsf{ID.Ct}(ivk)$
$c \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$
If $\mathrm{HT}[m, s]$:
  $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{c \leftarrow \mathrm{HT}[m, s]}$
$\mathrm{HT}[m, s] \leftarrow c$
$z \leftarrow \mathsf{ID.Rp}(ivk, isk, c, y)$
$\sigma \leftarrow (Y, s, z)$
Return σ

$\mathrm{H}(x)$

$m\|s \leftarrow x$
If not $\mathrm{HT}[m, s]$: $\mathrm{HT}[m, s] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$
Return $\mathrm{HT}[m, s]$

Adversary $\mathcal{P}^{\textsc{Tr},\textsc{Ch}}(ivk)$

$(m, \sigma) \leftarrow\!\!\$\ \mathcal{A}^{\textsc{Sign},\mathrm{H}}(ivk)$
$(Y, s, z) \leftarrow \sigma$
$c \leftarrow \mathrm{H}(m\|s)$
$c' \leftarrow \textsc{Ch}(Y, c)$ // $c' = c$
Return $(1, z)$

$\textsc{Sign}(m)$

$s \leftarrow\!\!\$\ \{0,1\}^{\mathsf{sl}}$
$Y\|c\|z \leftarrow\!\!\$\ \textsc{Tr}()$
$\mathrm{HT}[m, s] \leftarrow c$
$\sigma \leftarrow (Y, s, z)$
Return σ

$\mathrm{H}(x)$

$m\|s \leftarrow x$
If not $\mathrm{HT}[m, s]$: $\mathrm{HT}[m, s] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$
Return $\mathrm{HT}[m, s]$

Figure 17: Games and adversary for proof of Theorem 8.

identical until bad. By the Fundamental Lemma of Game Playing [10] we have

$$\begin{aligned}\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) &= \Pr[\mathrm{G}_0] = \Pr[\mathrm{G}_1] + (\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]) \le \Pr[\mathrm{G}_1] + \Pr[\mathrm{G}_1 \text{ sets } \mathsf{bad}] \\ &\le \Pr[\mathrm{G}_1] + \frac{(q_h + q_s)q_s}{2^{\mathsf{sl}}} \ .\end{aligned}$$

Adversary $\mathcal{P}$ of Fig. 17 executes $\mathcal{A}$, responding to H and $\textsc{Sign}$ queries of the latter via the shown procedures, which are subroutines in the code of $\mathcal{P}$. (Recall that $\mathrm{H}(\cdot)$ denotes $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{ID.cl}})$.) Adversary $\mathcal{P}$ simulates for $\mathcal{A}$ the environment of game G_1. In the execution of game $\mathbf{G}^{\text{cimp-cc}}_{\mathsf{ID}}(\mathcal{P})$ of Fig. 4, let B denote the event that $(Y, c) \in \mathrm{S}$, where Y, c is the argument to the single $\textsc{Ch}$ query made by our $\mathcal{P}$. Then

$$\Pr[\mathrm{G}_1] \le \mathbf{Adv}^{\text{cimp-cc}}_{\mathsf{ID}}(\mathcal{P}) + \Pr[\mathrm{B}] \ .$$

To complete the proof, it suffices to show that

$$\Pr[\mathrm{B}] \le \frac{q_h q_s}{2^{\mathsf{ID.cl}}} \ .$$

We bound $\Pr[\mathrm{B}]$ by the probability that c is a challenge in one of the transcripts. The message m in the forgery is assumed not one of those signed so $\mathrm{HT}[m\|s]$ was not set by $\textsc{Sign}$ and is thus independent of the transcript challenges. There are at most q_s transcript challenges and at most q_h queries to H, so $\Pr[\mathrm{B}] \le q_h q_s / 2^{\mathsf{ID.cl}}$. ▌

6.4 From CIMP-CU identification to UF signatures: FS

The first proofs of UF-security of **FS**-based signatures used a Forking Lemma and were quite complex [34]. More modular approaches were given in OO [32] and AABN [1]. AABN reduce

$\underline{\mathsf{DS.Kg}^{\mathrm{H}}}$
$(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$
$vk \leftarrow ivk\ ;\ sk \leftarrow isk$
Return (vk, sk)

$\underline{\mathsf{DS.Vf}^{\mathrm{H}}(vk, m, \sigma)}$
$ivk \leftarrow vk\ ;\ (Y, z) \leftarrow \sigma$
$c \leftarrow \mathrm{H}(Y \| m)$
Return $\mathsf{ID.Vf}(ivk, Y \| c \| z)$

$\underline{\mathsf{DS.Sig}^{\mathrm{H}}(vk, sk, m)}$
$ivk \leftarrow vk\ ;\ isk \leftarrow sk$
$(Y, y) \leftarrow\!\!\$\ \mathsf{ID.Ct}(ivk)$
$c \leftarrow \mathrm{H}(Y \| m)$
$z \leftarrow \mathsf{ID.Rp}(ivk, isk, c, y)$
$\sigma \leftarrow (Y, z)$
Return σ

Figure 18: The construction of signature scheme $\mathsf{DS} = \mathbf{FS}[\mathsf{ID}]$ from identification scheme ID. By $\mathrm{H}(\cdot)$ we denote random oracle $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{ID.cl}})$ with range $\{0,1\}^{\mathsf{ID.cl}}$.

UF-security of the signature scheme to IMP-PA security of the identification scheme. (The latter is established separately via the Reset Lemma of [6].) The reduction of AABN is not tight.

Our framework allows a tight reduction of the UF security of **FS**-based signatures to the CIMP-CU-security of the underlying identification scheme. The reason for this is simple, namely that CIMP-CU is the multi-impersonation version of IMP-PA. The proof is implicit in AABN [1]. We give a proof however for completeness and to illustrate how much simpler this proof is to prior ones.

We note that this tighter reduction does not change overall tightness. That is, in AABN, $\mathrm{P}_{\mathrm{sig}} \to \mathrm{P}_{\mathrm{id}}$ was not tight, while for us, it is, but the tightness of the overall $\mathrm{P}_{\mathrm{sig}} \to \mathrm{P}_{\mathrm{alg}}$ reduction remains the same in both cases.

The construction. The **FS** transform [17] associates to identification scheme ID the signature scheme $\mathsf{DS} = \mathbf{FS}[\mathsf{ID}]$ whose algorithms are defined in Fig. 18. Signing picks the commitment directly. The challenge is derived as a hash of the commitment and message. By $\mathrm{H}(\cdot)$ we denote random oracle $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{ID.cl}})$ with range $\{0,1\}^{\mathsf{ID.cl}}$.

Unforgeability. The following theorem says that the **FS** construction yields a UF-secure signature scheme if the underlying ID scheme offers CIMP-CU security and the commitment (as generated by the prover) is uniformly distributed over a large space. The latter condition is true for typical identification schemes. The intuition of the proof is that signing queries are answered via transcripts and hash queries are mapped to challenge queries, this failing only if commitments collide.

Theorem 9 *Let signature scheme* $\mathsf{DS} = \mathbf{FS}[\mathsf{ID}]$ *be obtained from identification scheme* ID *as in Fig. 18. Assume that for all ivk the distribution of* Y *induced by* $(Y, y) \leftarrow\!\!\$\ \mathsf{ID.Ct}(ivk)$ *is uniform over* $\mathsf{ID.CS}(ivk)$*. Let* $\mathsf{ID.CSS} = \min\{\, |\mathsf{ID.CS}(ivk)| \ :\ (ivk, isk, itk) \in [\mathsf{ID.Kg}]\,\}$*. Let* $\mathcal{A}$ *be a* UF*-adversary against* DS *making* q_h *queries to its* H *oracle and* q_s *queries to its* Sign *oracle. Then from* $\mathcal{A}$ *we construct a* CIMP-CU *adversary* $\mathcal{P}$ *such that*

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) \le \mathbf{Adv}^{\text{cimp-cu}}_{\mathsf{ID}}(\mathcal{P}) + \frac{q_s(2q_h + q_s - 1)}{2 \cdot \mathsf{ID.CSS}} \quad . \tag{9}$$

Adversary $\mathcal{P}$ *makes* q_s *queries to* Tr *and* $q_h + 1$ *queries to* Ch *and has running time about that of* $\mathcal{A}$*.*

Proof of Theorem 9: Game G_0 of Fig. 19 includes the boxed code, while game G_1 does not. Game G_0 is precisely the UF game of Fig. 2 with the algorithms of DS plugged in. We assume

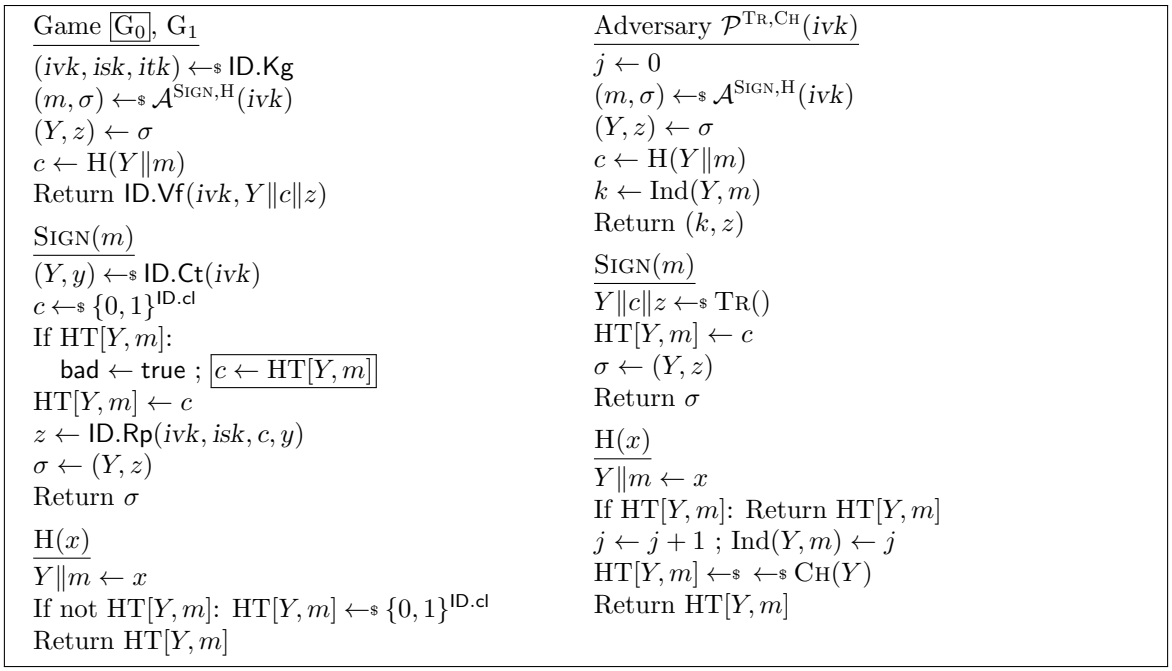


Figure 19: Games and adversary for proof of Theorem 9.

---

the message m in the forgery (m, σ) returned by $\mathcal{A}$ was not queried to $\textsc{Sign}$. Games $\mathrm{G}_0, \mathrm{G}_1$ are identical until bad. By the Fundamental Lemma of Game Playing [10] we have

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) = \Pr[\mathrm{G}_0] = \Pr[\mathrm{G}_1] + (\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]) \leq \Pr[\mathrm{G}_1] + \Pr[\mathrm{G}_1 \text{ sets } \mathsf{bad}] \ .$$

We assumed that for all ivk the distribution of Y induced by $(Y, y) \leftarrow\$ \mathsf{ID.Ct}(ivk)$ is uniform over $\mathsf{ID.CS}(ivk)$. This implies that

$$\Pr[\mathrm{G}_1 \text{ sets } \mathsf{bad}] \leq \sum_{i=0}^{q_s - 1} \frac{q_h + i}{\mathsf{ID.CSS}} = \frac{q_s q_h + q_s(q_s - 1)/2}{\mathsf{ID.CSS}} = \frac{q_s(2q_h + q_s - 1)}{2 \cdot \mathsf{ID.CSS}} \ .$$

Now consider adversary $\mathcal{P}$ of Fig. 19. It executes $\mathcal{A}$, responding to H and $\textsc{Sign}$ queries of the latter via the shown procedures, which are subroutines in the code of $\mathcal{P}$. (Recall that $\mathrm{H}(\cdot)$ denotes $\mathrm{H}(\cdot, \{0,1\}^{\mathsf{ID.cl}})$.) Adversary $\mathcal{P}$ simulates for $\mathcal{A}$ the environment of game G_1. When $\mathcal{A}$ queries $Y \| m$ to H, adversary $\mathcal{P}$ calls $\textsc{Ch}$ on Y and returns the resulting challenge as the answer to the H query. When $\mathcal{P}$ obtains $\mathcal{A}$'s forgery $(m, (Y, z))$, it looks up the partial transcript corresponding to Y, m and offers z as the corresponding response. A subtle point is that this partial transcript is defined because m was not queried to $\textsc{Sign}$. We have

$$\Pr[\mathrm{G}_1] \leq \mathbf{Adv}^{\text{cimp-cu}}_{\mathsf{ID}}(\mathcal{P}) \ ,$$

which completes the proof. ∎

6.5 From CIMP-UC identification to UF signatures: Swap

Micali and Reyzin [29] use the term "swap" for a specific construction of a signature scheme that they give with a tight reduction to the hardness of factoring. Folklore, and hints in the literature [2],

$\underline{\mathsf{DS.Kg}^{\mathrm{H}}}$
$(ivk, isk, itk) \leftarrow\!\!\$\ \mathsf{ID.Kg}$
$vk \leftarrow ivk\ ;\ sk \leftarrow (isk, itk)$
Return (vk, sk)

$\underline{\mathsf{DS.Vf}^{\mathrm{H}}(vk, m, \sigma)}$
$ivk \leftarrow vk\ ;\ (c, z) \leftarrow \sigma$
$Y \leftarrow \mathrm{H}(m\|c)$
Return $\mathsf{ID.Vf}(ivk, Y\|c\|z)$

$\underline{\mathsf{DS.Sig}^{\mathrm{H}}(vk, sk, m)}$
$ivk \leftarrow vk\ ;\ (isk, itk) \leftarrow sk$
$c \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$
$Y \leftarrow \mathrm{H}(m\|c)$
$y \leftarrow\!\!\$\ \mathsf{ID.Ct}^{-1}(ivk, itk, Y)$
$z \leftarrow \mathsf{ID.Rp}(ivk, isk, c, y)$
$\sigma \leftarrow (c, z)\ ;$ Return σ

Figure 20: The construction of signature scheme $\mathsf{DS} = \mathbf{Swap}[\mathsf{ID}]$ from a trapdoor identification scheme ID. By $\mathrm{H}(\cdot)$ we denote $\mathrm{H}(\cdot, \mathsf{ID.CS}(ivk))$.

indicate that researchers understand the method is more general. But exactly how general was not understood or determined before, perhaps for lack of definitions. Our definition of trapdoor identification and the CIMP-XY framework allows us to fill this gap and give a characterization of the swap method and also better understand it.

In this Section we define a transform of trapdoor identification schemes to signature schemes that we call **Swap** (cf. Fig. 20). We show that it yields UF-signature signatures if the identification scheme is CIMP-UC secure.

The construction. The **Swap** transform associates to trapdoor identification scheme ID the signature scheme $\mathsf{DS} = \mathbf{Swap}[\mathsf{ID}]$ whose algorithms are defined in Fig. 20.

Recall that in Section 6.1 we gave the **MdCmt** transform that constructs UUF-secure signatures from CIMP-UC-secure identification. Further, in Section 5 we proposed two generic techniques that convert UUF signatures to signatures with full UF security. One of the latter, **AR**, achieves its goal by adding randomness to signed messages as follows: for signing m, it picks a fresh random seed s and signs $m\|s$ instead. The seed is included in the signature. Overall, the combination of **MdCmt** with **AR** yields tightly secure signatures of the form $(c, \mathsf{ID.Rp}(c, \mathsf{ID.Ct}^{-1}(\mathrm{H}(m, s))), s)$. **Swap** (cf. Fig. 20) effectively, says that it is safe to choose c and s to be identical. Thus it can be viewed as an optimization of **MdCmt** + **AR**, giving up on modularity to achieve more compact UF secure signatures.

We note however that our **MdCmtCh** transform coupled with our **DR** UUF-to-UF transform yields UF signatures that seem superior in every way: they are shorter (response plus a bit as opposed to response plus a challenge), the (tight) reduction is to the weaker CIMP-UU notion, and the efficiency is the same. Thus we would view **Swap** at this point as of mostly historical interest.

Unforgeability. The following theorem says that the **Swap** construction yields a UF-secure signature scheme if the underlying ID scheme offers CIMP-UC security and has sufficiently large challenge length.

Theorem 10 *Let signature scheme* $\mathsf{DS} = \mathbf{Swap}[\mathsf{ID}]$ *be obtained from trapdoor identification scheme* ID *as in Fig. 20. Let* $\mathcal{A}$ *be a* UF*-adversary against* DS*. Suppose the number of queries that* $\mathcal{A}$ *makes to its* H *oracle is* q_h *and the number of queries it makes to* Sign *is* q_s*. Then from* $\mathcal{A}$ *we construct a* CIMP-UC *adversary* $\mathcal{P}$ *such that*

$$\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) \le \mathbf{Adv}^{\text{cimp-uc}}_{\mathsf{ID}}(\mathcal{P}) + \frac{(q_h + q_s)q_s + 1}{2^{\mathsf{ID.cl}}} \quad . \tag{10}$$

Adversary $\mathcal{P}$ *makes* $q_h + q_s + 1$ *queries to* Tr *and one query to* Ch*. Its running time is about that of* $\mathcal{A}$*.*

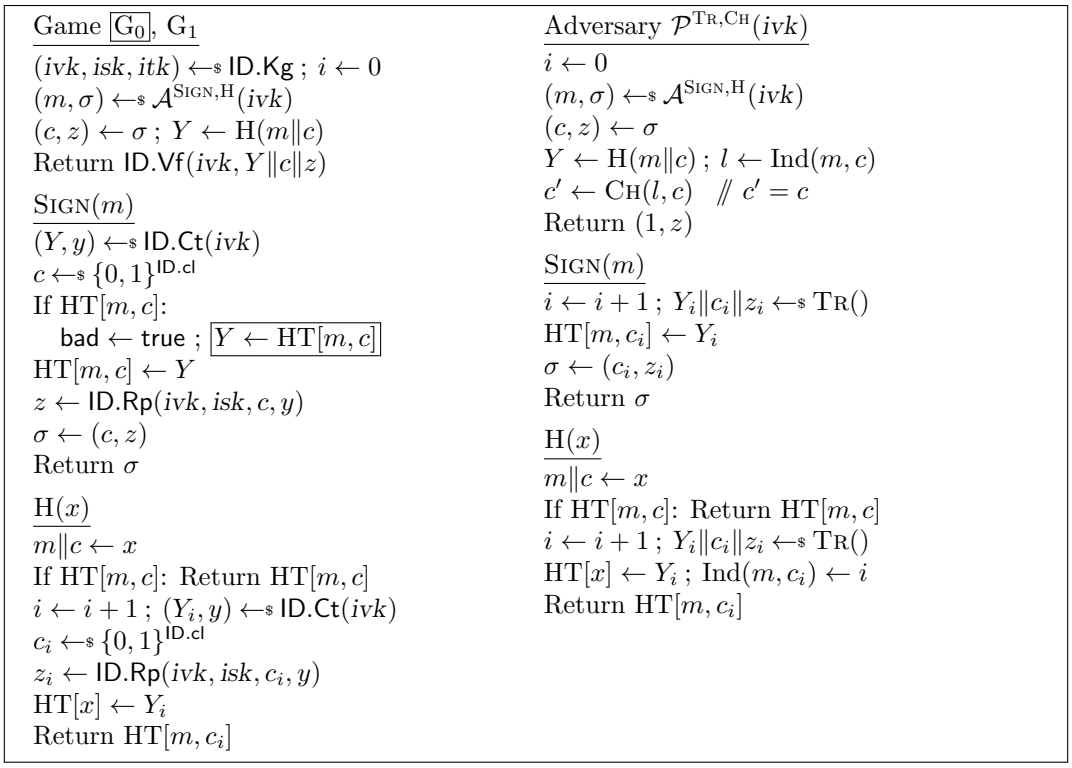


Figure 21: Games and adversary for the proof of Theorem 10.

Proof of Theorem 10: Game G_0 of Fig. 21 includes the boxed code, while game G_1 does not. Game G_0 is precisely the UF game of Fig. 2 with the algorithms of DS plugged in. We assume the message m in the forgery (m, σ) returned by $\mathcal{A}$ was not queried to $\textsc{Sign}$. Games $\mathrm{G}_0, \mathrm{G}_1$ are identical until bad. By the Fundamental Lemma of Game Playing [10] we have

$$\begin{aligned}\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A}) &= \Pr[\mathrm{G}_0] = \Pr[\mathrm{G}_1] + (\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]) \le \Pr[\mathrm{G}_1] + \Pr[\mathrm{G}_1 \text{ sets } \mathsf{bad}] \\ &\le \Pr[\mathrm{G}_1] + \frac{(q_h + q_s)q_s}{2^{\mathsf{ID.cl}}} \ .\end{aligned}$$

Adversary $\mathcal{P}$ of Fig. 21 executes $\mathcal{A}$, responding to H and $\textsc{Sign}$ queries of the latter via the shown procedures, which are subroutines in the code of $\mathcal{P}$. (Recall that $\mathrm{H}(\cdot)$ denotes $\mathrm{H}(\cdot, \mathsf{ID.CS}(ivk))$.) Adversary $\mathcal{P}$ simulates for $\mathcal{A}$ the environment of game G_1. In the execution of game $\mathbf{G}^{\text{cimp-cc}}_{\mathsf{ID}}(\mathcal{P})$ of Fig. 4, let B denote the event that $c = c_l$, where l, c is the argument to the single $\textsc{Ch}$ query made by our $\mathcal{P}$. Then

$$\Pr[\mathrm{G}_1] \le \mathbf{Adv}^{\text{cimp-uc}}_{\mathsf{ID}}(\mathcal{P}) + \Pr[\mathrm{B}] \ .$$

To complete the proof, it suffices to show that

$$\Pr[\mathrm{B}] \le \frac{1}{2^{\mathsf{ID.cl}}} \ .$$

The message m in the forgery is assumed not one of those signed so $\mathrm{HT}[m\|c]$ was set by H and not set or overwritten by $\textsc{Sign}$. So $\Pr[\mathrm{B}] \le 1/2^{\mathsf{ID.cl}}$. ▌

Signature scheme DS	$\mathrm{P_{id}}$	Bound on $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A})$	Sig. size	Equations
DR[**MdCmt**[ID]]	CIMP-UC	$2\epsilon/(1-2^{-l})$	$k+l$	(4),(5),(1)
Swap[ID]	CIMP-UC	$\epsilon + (q_h q_s + q_s^2 + 1)\cdot 2^{-l}$	$k+l$	(10),(1)
DR[**MdCmtCh**[ID]]	CIMP-UU	$4\epsilon + 4(q_h+q_s)\cdot 2^{-l}$	$k+1$	(4),(7),(2)
FS[ID]	CIMP-CU	$(q_h+1)(\sqrt{\epsilon}+2^{-l}) + (2q_hq_s+q_s^2)/2C$	$k+c$	(9),(3)

Figure 22: UF **signature schemes obtained from identification scheme** ID**.** We show bounds on the uf advantage of an adversary $\mathcal{A}$ making q_h queries to H and q_s queries to SIGN. Here $\epsilon = \mathbf{Adv}^{\mathrm{kr}}_{\mathsf{ID}}(\mathcal{I})$ is the kr-advantage of an adversary $\mathcal{I}$ of roughly the same running time as $\mathcal{A}$. By l, c, k we denote the lengths of the challenge, commitment and response, respectively. By C we denote the size of the commitment space. By $\mathrm{P_{id}}$ we denote the notion of identification security used in the $\mathrm{P_{sig}}\to\mathrm{P_{id}}$ reduction.

6.6 From identification to UF signatures: Summary

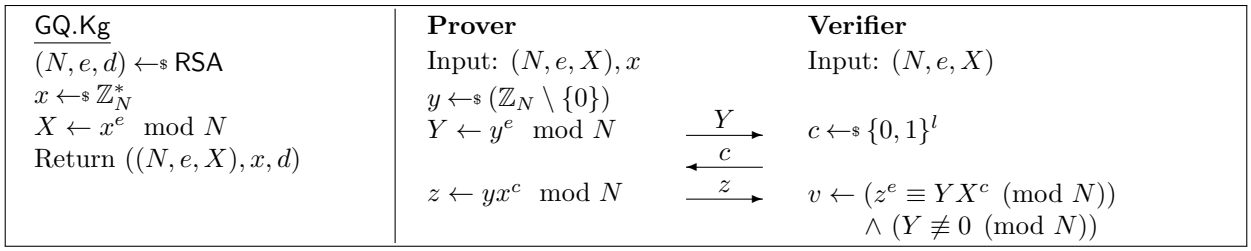
Fig. 22 puts things together. We consider obtaining a UF (not just UUF) signature scheme DS from a given identification scheme ID via the various transforms in this paper. In the first three rows, the identification scheme is assumed to be trapdoor. Whenever a transform achieves UUF, we apply **DR** on top to get UF. We give bounds on the uf-advantage $\mathbf{Adv}^{\mathrm{uf}}_{\mathsf{DS}}(\mathcal{A})$ of an adversary $\mathcal{A}$ making q_h queries to H and q_s queries to SIGN. By $l = \mathsf{ID.cl}$ we denote the challenge length of ID, and by $C = \mathsf{ID.CSS}$ the size of the commitment space. We show the full $\mathrm{P_{sig}}\to\mathrm{P_{alg}}$ reduction, so that the bounds are in terms of the kr-advantage $\epsilon = \mathbf{Adv}^{\mathrm{kr}}_{\mathsf{ID}}(\mathcal{I})$ of a kr-adversary $\mathcal{I}$ having about the same running time as $\mathcal{A}$. The bounds are obtained by combining the various relevant theorems, referring to the indicated equations. We show the notion $\mathrm{P_{id}}$ of identification security used as an intermediate point, namely $\mathrm{P_{sig}} \to \mathrm{P_{id}} \to \mathrm{P_{alg}}$. Signature size is shown as a function of the size k of a response, the challenge length l and the size c of a commitment. In summary, the bounds in the first three rows are tight, but the transform of the third row has the added advantage of shorter signatures and a linear (as opposed to quadratic) additive term in the bound. We do not show the **MdCh** transform from CIMP-CC because the latter is not achieved by Sigma protocols. We note that the bound for **FS** is that same as in [1]. (Our $\mathrm{P_{sig}}\to\mathrm{P_{id}}$ reduction, unlike theirs, is tight, but there is no change in the tightness of the full $\mathrm{P_{sig}}\to\mathrm{P_{alg}}$ reduction.)

7 Signatures from GQ

Our framework applies to a large number of identification schemes to turn them into signature schemes with tight reductions. As an illustrative and canonical example, we discuss the GQ identification scheme. We show how it is a trapdoor identification scheme and discuss its security, obtaining various signature schemes based on the one-wayness of RSA.

RSA GENERATORS. An RSA generator RSA for a modulus length k is an algorithm that returns a tuple (N, e, d) where modulus $N = pq$ in the range $2^{k-1} < N < 2^k$ is the product of distinct, odd primes p and q, encryption and decryption exponents e, d are in $\mathbb{Z}^*_{\varphi(N)}$ and $ed \equiv 1 \pmod{\varphi(N)}$. The OW (one-wayness) advantage of an adversary $\mathcal{A}$ against RSA by $\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{RSA}}(\mathcal{A}) = \Pr[\mathbf{G}^{\mathrm{ow}}_{\mathsf{RSA}}(\mathcal{A})]$ where the game is in Fig. 23. One-wayness is the standard assumption on RSA.

GQ. Fig. 23 shows the GQ identification scheme [22] associated to RSA and a challenge length $l < k$ such that $\gcd(e, c) = 1$ for all $c \in \{0,1\}^l$ and all $(N, e, d) \in [\mathsf{RSA}]$. The commitment space



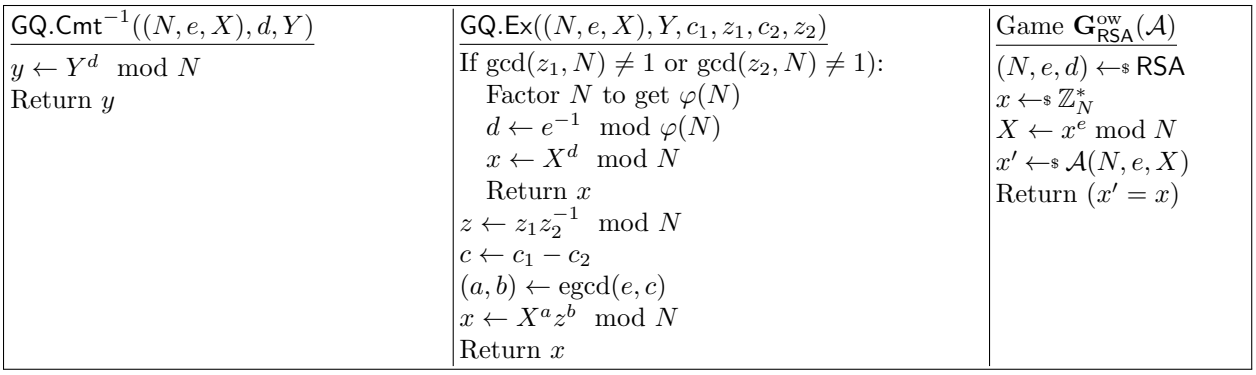


Figure 23: Identification scheme GQ associated to RSA generator RSA with modulus length k, and challenge length l. **Bottom right:** Game defining one-wayness of RSA generator RSA.

---

is $\mathbb{Z}_N \setminus \{0\}$. We can see via Algorithm $\mathsf{GQ.Cmt}^{-1}$ of Fig. 23 that GQ is trapdoor. Note that this requires putting the decryption exponent d in the secret key, a change from the classic GQ scheme.

In order to apply Theorem 2 to tightly obtain a signature scheme from GQ using our transformations, we must show that GQ satisfies several conditions. Most of these are standard observations. Then, since we have tight reductions $\mathrm{P}_{\mathrm{sig}} \to \mathrm{P}_{\mathrm{id}}$ and $\mathrm{P}_{\mathrm{id}} \to \mathrm{P}_{\mathrm{alg}}$, we can pick the RSA modulus based on the assumption that the NFS is the best factoring method.

CIMP-XY. In order to use GQ in our transforms, we need to establish its security under the notions in our framework. We will apply Theorem 2 to show that CIMP-UC and CIMP-UU are obtained tightly from the one-wayness of RSA. This involves noting that (1) key recovery for GQ is exactly one-wayness of RSA (2) GQ is HVZK (3) GQ is extractable. Claims (2), (3) are well known but for completeness we recall why they hold. First, to establish HVZK, given a public key (N, e, X), transcripts can be simulated as follows. Sample $c \leftarrow_{\$} \{0,1\}^l$ and $z \leftarrow_{\$} \mathbb{Z}_N \setminus \{0\}$. Compute $Y \leftarrow z^e / X^c \mod N$. The transcript is $Y \| c \| z$. This has the same distribution as transcripts generated in Fig. 23: c is clearly identically distributed in Fig. 23; if y is uniform on $\mathbb{Z}_N \setminus \{0\}$ then so is Y since exponentiation by e is a permutation on $\mathbb{Z}_N$; and thus z is also uniform on $\mathbb{Z}_N \setminus \{0\}$. Next we recall why it is extractable. Algorithm GQ.Ex of Fig. 23 provides extractability of the GQ secret key x given two accepting transcripts with the same commitment but distinct challenges. By egcd we denote the extended gcd algorithm that given relatively prime inputs e, c returns a, b such that $ae + bc = 1$. Claim (1) is detailed below. The definition of kr-security was given in Section 4.

Theorem 11 *Let* GQ *be the identification scheme associated to RSA generator* RSA *with modulus length k and challenge length l as above. Let $\mathcal{I}$ be a* KR *adversary. Then from $\mathcal{I}$ we can construct* OW *adversary $\mathcal{A}$ such that*

$$\mathbf{Adv}^{\mathrm{kr}}_{\mathsf{GQ}}(\mathcal{I}) \leq \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{RSA}}(\mathcal{A}) \ . \tag{11}$$

The running time of $\mathcal{A}$ is that of $\mathcal{I}$.

The proof of Theorem 11 is simple. $\mathcal{A}$ immediately provides the (N, e, X) values from the OW-challenger for RSA to $\mathcal{I}$. For simulating KR, this is sufficient: in particular, $\mathcal{A}$ does not have to simulate transcript queries. $\mathcal{I}$ returns the secret key of the ID scheme directly, which is the solution to the RSA challenge. Note that key recovery only requires recovery of *isk*, not of the trapdoor *itk*.

Hashing onto commitment space. Several constructions require a random oracle with range $\mathsf{ID.CS}(ivk)$, which is $\mathbb{Z}_N \setminus \{0\}$, which we can easily build.

Bounds for concrete instantiations. Signature sizes and unforgeability (that is, UF, not UUF) bounds obtained by constructing concrete signature schemes from the GQ scheme via the transforms in this paper can be obtained by instantiating ID with GQ in Fig. 22. Replace $\epsilon = \mathbf{Adv}^{\mathrm{kr}}_{\mathsf{ID}}(\mathcal{I})$ by $\epsilon = \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{RSA}}(\mathcal{I})$, the advantage of an adversary $\mathcal{I}$, having about the same running time as $\mathcal{A}$, in breaking one-wayness of RSA. The value of C is 2^{k-1}. The summary is that the bounds are tight in the first three cases, but the third scheme has the added advantage of shorter signatures.

With regard to signatures from RSA, we already have other schemes with reduction to the one-wayness of RSA in the ROM. For FDH [8], the reduction is not tight [8, 13], while for PSS [8] and the KW scheme [23], it is. Signature size is k bits for the first two and $k + 1$ for the third. The GQ-based schemes we have illustrated do not thus appear to yield anything we do not have for RSA-based signatures. (**FS** can have faster signing, but the looseness of the reduction may make this moot.) We have discussed GQ-based signatures here merely as the simplest illustrative example for applying our framework.

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, F. Ben Hamouda, and D. Pointcheval. Tighter reductions for forward-secure signature schemes. In K. Kurosawa and G. Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 292–311. Springer, Heidelberg, Feb. / Mar. 2013.

[3] 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.

[4] A. Bagherzandi, J. H. Cheon, and S. Jarecki. Multisignatures secure under the discrete logarithm assumption and a generalized forking lemma. In P. Ning, P. F. Syverson, and S. Jha, editors, *ACM CCS 08*, pages 449–458. ACM Press, Oct. 2008.

[5] M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In A. Juels, R. N. Wright, and S. Vimercati, editors, *ACM CCS 06*, pages 390–399. ACM Press, Oct. / Nov. 2006.

[6] M. Bellare and A. Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 162–177. Springer, Heidelberg, Aug. 2002.

[7] 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.

[8] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. D. Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 92–111. Springer, Heidelberg, May 1995.

[9] M. Bellare and P. Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In U. M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416. Springer, Heidelberg, May 1996.

[10] 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.

[11] 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.

[12] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 402–414. Springer, Heidelberg, May 1999.

[13] J.-S. Coron. On the exact security of full domain hash. In M. Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 229–235. Springer, Heidelberg, Aug. 2000.

[14] R. Cramer. *Modular Design of Secure, yet Practical Protocls.* PhD thesis, University of Amsterdam, 1996.

[15] U. Feige, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. In A. Aho, editor, *19th ACM STOC*, pages 210–217. ACM Press, May 1987.

[16] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[17] 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.

[18] E.-J. Goh and S. Jarecki. A signature scheme as secure as the Diffie-Hellman problem. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 401–415. Springer, Heidelberg, May 2003.

[19] S. Goldwasser and Y. T. Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th FOCS*, pages 102–115. IEEE Computer Society Press, Oct. 2003.

[20] 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.

[21] S. Goldwasser and R. Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In E. F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 228–245. Springer, Heidelberg, Aug. 1993.

[22] L. C. Guillou and J.-J. Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In S. Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 216–231. Springer, Heidelberg, Aug. 1990.

[23] J. Katz and N. Wang. Efficiency improvements for signature schemes with tight security reductions. In S. Jajodia, V. Atluri, and T. Jaeger, editors, *ACM CCS 03*, pages 155–164. ACM Press, Oct. 2003.

[24] E. Kiltz, D. Masny, and J. Pan. Optimal security proofs for signatures from identification schemes. Cryptology ePrint Archive, Report 2016/191, 2016. http://eprint.iacr.org/.

[25] N. Koblitz and A. Menezes. The random oracle model: A twenty-year retrospective. Cryptology ePrint Archive, Report 2015/140, 2015. http://eprint.iacr.org/2015/140.

[26] A. Lysyanskaya. Unique signatures and verifiable random functions from the DH-DDH separation. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 597–612. Springer, Heidelberg, Aug. 2002.

[27] 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.

[28] 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.

[29] S. Micali and L. Reyzin. Improving the exact security of digital signature schemes. *Journal of Cryptology*, 15(1):1–18, 2002.

[30] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Heidelberg, Apr. 2012.

[31] D. M'Raïhi, D. Naccache, D. Pointcheval, and S. Vaudenay. Computational alternatives to random number generators. In S. E. Tavares and H. Meijer, editors, *SAC 1998*, volume 1556 of *LNCS*, pages 72–80. Springer, Heidelberg, Aug. 1999.

[32] 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.

[33] H. Ong and C.-P. Schnorr. Fast signature generation with a Fiat-Shamir-like scheme. In I. Damgård, editor, *EUROCRYPT'90*, volume 473 of *LNCS*, pages 432–440. Springer, Heidelberg, May 1991.

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

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

Adversary $\mathcal{P}_1^{\text{Tr},\text{Ch}}(ivk)$	$\text{Ch}^*(Y)$
$k^* \leftarrow\!\!\$\ \{1,\ldots,q\}$; $j \leftarrow 0$	$j \leftarrow j+1$
$(k,z) \leftarrow\!\!\$\ \mathcal{P}^{\text{Tr},\text{Ch}^*}(ivk)$	If $j \neq k^*$: $c \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ID.cl}}$
If $k \neq k^*$: Return $\perp$	Else $c \leftarrow\!\!\$\ \text{Ch}(Y)$
Return $(1,z)$	$\text{CT}[j] \leftarrow Y\|c$; Return c

Figure 24: Adversary for proof of Lemma 12.

A Proof of Theorem 3

The following results imply Theorem 3. They rely on standard techniques and are included here for completeness.

Lemma 12 *Let* ID *be an identification scheme. Let* $\mathcal{P}$ *be a* CIMP-CU*-adversary against* ID *making* q *queries to its* Ch *oracle. Then from* $\mathcal{P}$ *we can construct a* CIMP-CU *adversary* $\mathcal{P}_1$ *making one query to its* Ch *oracle such that*

$$\mathbf{Adv}^{\text{cimp-cu}}_{\mathsf{ID}}(\mathcal{P}) \leq q \cdot \mathbf{Adv}^{\text{cimp-cu}}_{\mathsf{ID}}(\mathcal{P}_1) \ .$$

Adversary $\mathcal{P}_1$ *makes as many queries to its* Tr *oracle as* $\mathcal{P}$ *does. The running time of* $\mathcal{P}_1$ *is about that of* $\mathcal{P}$*.*

Proof of Lemma 12: Adversary $\mathcal{P}_1$ is shown in Fig. 24. It has access to oracles Tr, Ch as per game $\mathbf{G}^{\text{cimp-cu}}_{\mathsf{ID}}(\mathcal{P}_1)$ in which it is executing, but may only make one query to its Ch oracle. It guesses an instance k^* uniformly from $\{1,\ldots,q\}$ and runs $\mathcal{P}$. Adversary $\mathcal{P}_1$ passes $\mathcal{P}$'s Tr queries directly to its own Tr oracle. $\mathcal{P}_1$ simulates answers to $\mathcal{P}$'s queries to its Ch oracle via the subroutine Ch^*, calling its own oracles inside this. Adversary $\mathcal{P}_1$'s simulation is perfect. Since $\mathcal{P}_1$ will guess the instance k^* which $\mathcal{P}$ successfully impersonates with probability $1/q$, adversary $\mathcal{P}_1$'s success probability is at least $1/q$ times that of $\mathcal{P}$. ∎

Lemma 13 *Let* ID *be an identification scheme that is honest verifier zero-knowledge and extractable. Then for any* CIMP-CU *adversary* $\mathcal{P}_1$ *making one query to its* Ch *oracle, we can construct an adversary* $\mathcal{I}$ *such that*

$$\mathbf{Adv}^{\text{cimp-cu}}_{\mathsf{ID}}(\mathcal{P}_1) \leq \frac{1}{2^{\mathsf{ID.cl}}} + \sqrt{\mathbf{Adv}^{\text{kr}}_{\mathsf{ID}}(\mathcal{I})} \ .$$

The running time of $\mathcal{I}$ *is about twice that of* $\mathcal{P}_1$*, plus the time for an execution of* $\mathsf{ID.Ex}$*.*

Lemma 13 follows directly from the reset lemma of [6]. The reset lemma gives a bound on $\mathbf{Adv}^{\text{cimp-cu}}_{\mathsf{ID}}(\mathcal{P})$ based on the probability of obtaining a pair of distinct valid transcripts $Y\|c_1\|z_1$, $Y\|c_2\|z_2$ by running the same adversary twice, the second time where the adversary is "reset" with the same state it had the moment it output its commitment Y. From these distinct valid transcripts and the verification key ivk, we can execute $\mathsf{ID.Ex}$ to obtain the secret key isk, winning the key recoverability game.