An extended abstract of this paper appears in Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, Volume 2332 of Lecture Notes in Computer Science, pages 418–433, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer-Verlag, Berlin, Germany. This is the full version.

From Identification to Signatures via the Fiat-Shamir Transform: Minimizing Assumptions for Security and Forward-Security

M. Abdalla* J.H. An† M. Bellare‡ C. Namprempre§

February 2002

Abstract

The Fiat-Shamir paradigm for transforming identification schemes into signature schemes has been popular since its introduction because it yields efficient signature schemes, and has been receiving renewed interest of late as the main tool in deriving forward-secure signature schemes. In this paper, minimal (meaning necessary and sufficient) conditions on the identification scheme to ensure security of the signature scheme in the random oracle model are determined, both in the usual and in the forward-secure cases. Specifically, it is shown that the signature scheme is secure (resp. forward-secure) against chosen-message attacks in the random oracle model *if and only if* the underlying identification scheme is secure (resp. forward-secure) against impersonation under *passive* (i.e., eavesdropping only) attacks, and has its commitments drawn at random from a large space. An extension is proven incorporating a random seed into the Fiat-Shamir transform so that the commitment space assumption may be removed.

Keywords: Signature schemes, identification schemes, Fiat-Shamir transform, forward security, random oracle model, security proofs.

*Departement d'Informatique, École normale supérieure, 45 Rue d'Ulm, 75230 Paris Cedex 05, France. Email: `Michel.Abdalla@ens.fr`. URL: `http://www.di.ens.fr/users/mabdalla`. Part of this work was done while the author was at University of California, San Diego. Supported in part by the third author's grants.

† SoftMax, Inc., 10760 Thornmint Road, San Diego, CA 92128, USA. Email: `jeehea@cs.ucsd.edu`. URL: `http://www.cs.ucsd.edu/users/jeehea`. Work done while at University of California, San Diego.

‡ Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: `mihir@cs.ucsd.edu`. URL: `http://www.cs.ucsd.edu/users/mihir`. Supported by in part by NSF Grants CCR-0098123, CNS-0524765, CNS-0627779, a 1996 Packard Foundation Fellowship in Science and Engineering, an IBM Faculty Partnership Development Award, and a gift from Intel Corporation.

§Electrical Engineering Department, Faculty of Engineering, Thammasat University, Klong Luang, Patumtani 12121, Thailand. Email: `cnamprem@engr.tu.ac.th`. URL: `http://chanathip.ee.engr.tu.ac.th`. Supported in part by the third author's grants and the Thailand Research Fund.

1 Introduction

The Fiat-Shamir method of transforming identification schemes into signature schemes [20] is popular because it yields efficient signature schemes, and has been receiving renewed interest of late as the main tool in deriving forward-secure signature schemes. We find minimal (meaning necessary and sufficient) conditions on the identification scheme to ensure security of the signature scheme in the random oracle model. The conditions are simple and natural. Below we begin with some background and discussion of known results, and then move to our results, considering first the usual and then the forward-secure case.

CANONICAL ID SCHEMES. The Fiat-Shamir (FS) transform applies to identification (ID) schemes having a three-move format that we call *canonical*. The prover, holding a secret key sk, sends a message CMT called a *commitment* to the verifier. The verifier returns a *challenge* CH consisting of a random string of some length. The prover provides a *response* RSP. Finally, the verifier applies a verification algorithm V to the prover's public key pk and the conversation $\mathrm{CMT}\|\mathrm{CH}\|\mathrm{RSP}$ to obtain a *decision* bit, and accepts iff $\mathrm{Dec} = 1$. The length of the challenge is $c(k)$ where k is the security parameter and c is a function associated to the scheme. A large number of canonical ID schemes are known (e.g., [20, 24, 11, 29, 36, 14, 21, 32, 31, 38, 33]) and are candidates for conversion to signature schemes via the FS transform.

THE FS TRANSFORM. The signer has the public and secret keys pk, sk of the prover of the ID scheme. To sign a message M it computes CMT just as the prover would, hashes $\mathrm{CMT}\|M$ using a public hash function $H\colon \{0,1\}^* \to \{0,1\}^{c(k)}$ to obtain a "challenge" $\mathrm{CH} = H(\mathrm{CMT}\|M)$, computes a response RSP just as the prover would, and sets the signature of M to $\mathrm{CMT}\|\mathrm{RSP}$. To verify that $\mathrm{CMT}\|\mathrm{RSP}$ is a signature of M, one first computes $\mathrm{CH} = H(\mathrm{CMT}\|M)$ and then checks that the verifier of the identification scheme would accept, namely $V(pk, \mathrm{CMT}\|\mathrm{CH}\|\mathrm{RSP}) = 1$. Fiat and Shamir's suggestion that one model H as a random oracle [20] is adopted by previous security analyses, both in the standard setting [35, 30] and in the forward-secure setting [5, 1, 25], and also by this paper.

TARGET SECURITY GOAL FOR SIGNATURES. Focusing first on the standard setting (meaning where forward-security is not a goal), the target is to prove that the signature scheme is unforgeable under chosen-message attack [23] in the random oracle model [8]. This requires that it be computationally infeasible for an adversary to produce a valid signature of a new message even after being allowed a chosen-message attack on the signer and provided oracle access to the random hash function.

NON-TRIVIALITY. Previous works [35, 30] have assumed that the ID scheme has the property that the space from which the prover draws its commitments is large, meaning super-polynomial. We refer to a scheme with this property as non-trivial. (A more general definition, in terms of min-entropy, is Definition 3.2.) We point out in Section 6 that non-triviality of the ID scheme is *necessary* for the security of the signature scheme derived via the FS transform, and thus all discussions related to the FS transform below will assume it. (We will see however that this assumption can be removed by considering a randomized generalization of the FS transform.)

1.1 Main result

In this work we find simple and natural assumptions on the ID scheme that are both sufficient and necessary for the security of the signature scheme, and are related to the security of the underlying ID scheme for the purpose for which it was presumably designed, namely identification.

Statement. We prove the following: The signature scheme resulting from applying the FS transform to a non-trivial ID scheme is secure against chosen-message attack in the random oracle model *if and only if* the underlying identification scheme is secure against impersonation under passive attack. A precise statement is Theorem 3.3. Let us recall the notion of security used here, following [19], and then compare this to previous work.

Security of identification schemes. As with any primitive, a notion of security considers adversary goals (what it has to do to win) and adversary capability (what attacks it is allowed). Naturally, for an ID scheme, the adversary goal is impersonation: it wins if it can interact with the verifier in the role of a prover and convince the latter to accept. There are two natural attacks to consider: passive and active. Passive attacks correspond to eavesdropping, meaning the adversary is in possession of transcripts of conversations between the real prover and the verifier. Active attacks mean that it gets to play the role of a verifier, interacting with the real prover in an effort to extract information. Security against impersonation under active attack is the attribute usually desired of an ID scheme to be used in practice for the purpose of identification. It is however the weaker attribute of security against impersonation under passive attack that we show is tightly coupled to the security of the derived signature scheme.

1.2 Comparison with previous work

Much work has been done in the past to study the application of the FS-transform to an ID scheme to obtain a signature scheme. Some of the analyses have identified the sufficient conditions on the ID scheme for the transformation to yield a secure signature scheme. The pioneering work of Pointcheval and Stern [35] assumes that the identification scheme is honest verifier zero-knowledge and also, in their Forking Lemma, assume a property that implies that it is a "proof of knowledge" [19, 4], namely that there is an algorithm that can produce two transcripts which start with the same commitment $(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$, $(\textsc{Cmt}, \textsc{Ch}', \textsc{Rsp}')$ such that, if both are accepted by the verifier V, the underlying secret key can be determined. (This property is called *collision intractability* in [18].) We refer to an ID scheme meeting these conditions as **PS**-secure.

Ohta and Okamoto [30] assume that the identification scheme is honest-verifier (perfect) zero-knowledge and that it is computationally infeasible for a cheating prover to convince the verifier to accept. We refer to such an ID scheme as **OO**-secure.

Relations. Figure 1 puts our result in context with previous works. It considers the three assumptions made on non-trivial identification schemes for the purpose of proving security of the corresponding FS-transform based signature scheme: **PS**-security [35]; **OO**-security [30]; and the assumption of security against impersonation under passive attacks. As the picture indicates, all three suffice to prove security of the signature scheme in the random oracle model. However, the assumption we make is not only necessary but also sufficient, while the others are provably not necessary. Furthermore, our assumption is weaker than the other assumptions, shown to imply them but not be implied by them. Let us discuss this further.

It is well known that **PS** or **OO** security imply security against impersonation under passive attacks. The converse, however, is not true: in Section 4, we present examples that show that a non-trivial ID scheme could be secure against impersonation under passive attack yet be neither **PS** nor **OO** secure. Thus, our assumption on the ID scheme is weaker than previous ones. On the other hand, the fact that this assumption is necessary says that it is minimal. A consequence is that there exist (non-trivial) ID schemes that are neither **PS**-secure nor **OO**-secure, yet the corresponding signature scheme is secure, showing that the previous assumptions

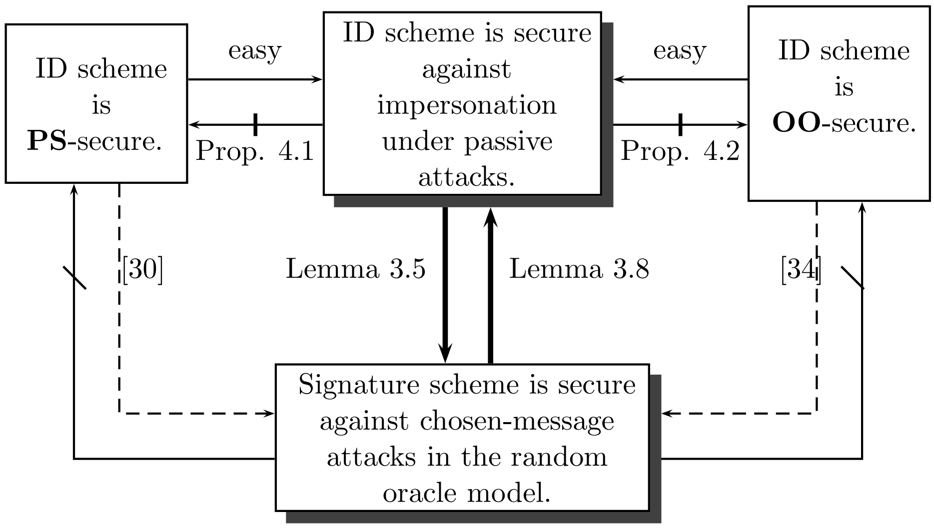

Figure 1: We depict relations among assumptions on non-trivial ID schemes that have been used to prove security of the corresponding signature scheme. An arrow denotes an implication while a barred arrow denotes a separation. The dotted arrows are existing relations, annotated with citations to the papers establishing them. The full arrows are either relations established in this paper, or are easy.

---

are not necessary conditions for the security of the signature scheme.

In practice, these gaps may not be particularly limiting, because practical ID schemes for the most part are **PS**-secure or **OO**-secure. However our result can simplify future or even existing constructions of identification based signature schemes, and clarifies the theoretical picture.

Assumptions related to the problem. Fiat and Shamir [20] suggested that their transform be applied to an ID scheme. However, previous security analyses have made assumptions that are in fact not inherent to the notion of identification itself. By this we mean assumptions such as honest verifier zero-knowledge or that underlying the forking lemma. These types of properties are convenient tools in the analysis of ID schemes, but not the end goals of identification. In particular, as we show in Section 4, there exist ID schemes, secure even against active attack, that are not honest verifier zero-knowledge and fail to meet the conditions of the forking lemma. In contrast, our necessary and sufficient condition, namely security against impersonation under passive attacks, is a natural end goal of identification. Our results thus support the original intuition that seems to have guided [20], namely that the security of the signature scheme stems from the security of the identification scheme relative to the job for which the latter was intended.

1.3 Generalized transform

As previously mentioned, the non-triviality assumption on an ID scheme is necessary to guarantee that the FS transform yields a secure signature scheme. We define a randomized generalization of the Fiat-Shamir transform (described in detail in Construction 3.1). We show that this modification allows the non-triviality assumption to be removed. Specifically, we prove that the signature scheme resulting from our generalized Fiat-Shamir transform is secure against chosen-message attack in the random oracle model *if and only if* the underlying identification scheme is secure against impersonation under passive attack. A precise statement is presented in Theorem 3.4.

We note that the process of applying our generalized transform to a given ID scheme can be alternatively viewed as first modifying the ID scheme by enhancing its commitment space and then applying the FS transform.

1.4 Results for forward security

An important paradigm in the construction of forward-secure signature schemes, beginning with [5] and continuing with [1, 25], has been to first design a forward-secure identification scheme and then obtain a forward-secure signature scheme via the FS transform. The analyses in these works are however ad hoc.

We prove an analogue of our main result that says that the signature scheme resulting from applying FS transform to a non-trivial ID scheme is forward-secure against chosen-message attacks in the random oracle model *if and only if* the underlying identification scheme is forward-secure against impersonation under passive attack. An extension based on the generalized FS transform, analogous to that mentioned above, also holds. This brings the characterization described above to forward-secure signature schemes, and helps to unify previous results [5, 1, 25]. Our result can simplify future or even existing constructions of identification based forward-secure signature schemes, saving repetition in the analytical work. One should note however that non-modular analyses may have the benefit of yielding better concrete security than is obtained by our general result [1, 25].

1.5 Discussion and remarks

THE RANDOM ORACLE DEBATE. It is important to be aware that a proof of security in the random oracle model does not guarantee security when the random oracle is instantiated [16, 22, 3, 26]. The value of the random oracle paradigm, as explained at the time of its introduction in [9], is that the instantiated protocols are more practical than their competitors while possessing a security guarantee that, although not formally well-defined, has proven good in practice. (The counter-examples of [16, 22, 3, 26] are all artificial in one way or another.) FS signatures, more efficient than any competitors with standard model proofs, are a case in point. Beyond this, our work is motivated by the desire to understand phenomena as best as one can. The FS transform is manifestly important. It has been in existence for a long time, is in use, and its use is even expanding into new domains [13]. The use of the random oracle model in this context, following Fiat and Shamir's own suggestion [20], enhances our understanding, and a complete picture of the properties of the FS transform in the random oracle model is valuable in its own right and also as a possible basis for future random oracle avoiding steps, as has happened in the past with other primitives [15, 12].

OTHER TRANSFORMS. There are other methods of transforming ID schemes into signature schemes. A variant of the FS transform suggested by Micali and Reyzin [28] applies only to a subclass of canonical ID schemes. A transform suggested by Cramer and Damgård [18] has the advantage of not requiring random oracles in the analysis, but is relatively inefficient. Overall the FS transform has remained the most attractive, due to its wide applicability, the efficiency of the resulting signature scheme, and its robustness in the face of extra goals such as forward security, and thus is our focus.

THE PROOFS. Our proofs appear to be simpler than previous ones even though our results are stronger. We believe that this is true because our assumptions, although weaker, have extracted more of the properties of the ID scheme that are truly relevant to the security of the signature scheme, thereby leaving less to be proven.

AN (ALMOST) ANALOGOUS RESULT FOR IDENTITY-BASED SYSTEMS. In [6], Bellare, Namprempre, and Neven show that, for a certain class of secure identity-based identification (IBI) schemes, the corresponding identity-based signature (IBS) schemes obtained through the standard FS transform are secure. They show that this result does *not* hold in general, however: there exists a secure IBI scheme whose corresponding IBS scheme is insecure. We note that, although this result is stated with respect to the standard FS transform, with a small modification to the proof, the same result holds with respect to the generalized FS transform presented here as well.

1.6 Organization

Section 2 recalls the formal definition of the following: security of an identification scheme against impersonation under passive attacks; and security of a signature scheme, in the sense of unforgeability against chosen-message attacks, in the random oracle model. Section 3 presents our generalized FS transform, of which the FS transform itself is a special case, and then recalls the definition of min-entropy, on which the definition of non-triviality is based. It states two equivalence theorems, one for the FS transform and one for the generalized FS transform. It then states and proves lemmas used to derive these. Section 4 justifies the separations among security requirements made in previous works and the security of identification schemes. Section 5 presents the formal definitions for forward-secure identification and signatures, and states and proves the forward-security equivalence result. Section 6 explores the security implications of the presence and absence of the non-triviality condition.

2 Definitions

NOTATION. If $A(\cdot,\cdot,\ldots)$ is a randomized algorithm, then $y \leftarrow A(x_1,x_2,\ldots;R)$ means y is assigned the unique output of the algorithm on inputs $x_1,x_2,\ldots$ and coins R, while $y \overset{\$}{\leftarrow} A(x_1,x_2,\ldots)$ is shorthand for first picking R at random and then setting $y \leftarrow A(x_1,x_2,\ldots;R)$. We let $\mathsf{Coins}_A(k)$ denote the space from which R is drawn —it is a set of binary strings of some appropriate length— where k is the underlying security parameter. If S is a set then $s \overset{\$}{\leftarrow} S$ indicates that s is chosen uniformly at random from S. If $x_1,x_2,\ldots$ are strings then $x_1\|x_2\|\cdots$ denotes an encoding under which the constituent strings are uniquely recoverable. It is assumed any string x can be uniquely parsed as an encoding of some sequence of strings. The empty string is denoted ε.

CANONICAL IDENTIFICATION SCHEMES. We use the term *canonical* to describe a three-move protocol in which the verifier's move consists of picking and sending a random string of some length, and the verifier's final decision is a deterministic function of the conversation and the public key (cf. Figure 2). The specification of a *canonical identification scheme* will take the form $\mathcal{ID} = (K,P,V,c)$ where K is the *key generation* algorithm, taking input a security parameter $k \in \mathsf{N}$ and returning a public and secret key pair (pk, sk); P is the *prover* algorithm taking input sk and the current conversation prefix to return the next message to send to the verifier; c is a function of k indicating the length of the verifier's challenge; V is a deterministic algorithm taking pk and a complete conversation transcript to return a boolean decision Dec on whether or not to accept. We associate to $\mathcal{ID}$ and each (pk, sk) a randomized *transcript generation oracle* which takes no inputs and returns a random transcript of an "honest" execution, namely:

$$
\begin{array}{lcl}
\underline{\text{Prover}} & & \underline{\text{Verifier}} \\
\text{Input: } sk & & \text{Input: } pk \\
 & \xrightarrow{\textsc{Cmt}} & \\
 & & \textsc{Ch} \stackrel{\$}{\leftarrow} \{0,1\}^{c(k)} \\
 & \xleftarrow{\textsc{Ch}} & \\
 & \xrightarrow{\textsc{Rsp}} & \\
 & & \mathrm{Dec} \leftarrow V(pk, \textsc{Cmt}\|\textsc{Ch}\|\textsc{Rsp})
\end{array}
$$

Figure 2: A canonical identification protocol.

Function $\mathsf{Tr}^{\mathcal{ID}}_{pk,sk,k}$

$R_P \stackrel{\$}{\leftarrow} \mathsf{Coins}_P(k)$

$\textsc{Cmt} \leftarrow P(sk; R_P)$; $\textsc{Ch} \stackrel{\$}{\leftarrow} \{0,1\}^{c(k)}$; $\textsc{Rsp} \leftarrow P(sk, \textsc{Cmt}\|\textsc{Ch}; R_P)$;

return $\textsc{Cmt}\|\textsc{Ch}\|\textsc{Rsp}$

The scheme must obey a standard completeness requirement, namely that for every k, we have $\Pr[V(pk, \textsc{Cmt}\|\textsc{Ch}\|\textsc{Rsp}) = 1] = 1$, the probability being over $(pk, sk) \stackrel{\$}{\leftarrow} K(k)$ and $\textsc{Cmt}\|\textsc{Ch}\|\textsc{Rsp} \stackrel{\$}{\leftarrow} \mathsf{Tr}^{\mathcal{ID}}_{pk,sk,k}$.

Security against impersonation under passive attacks considers an adversary —here called an impersonator— whose goal is to impersonate the prover without the knowledge of the secret key. In practice, such an adversary generally has access not only to the public key but also to conversations between the real prover and an honest verifier, possibly via eavesdropping over the network. We model this setting by viewing an impersonator as a probabilistic algorithm I and giving to it the public key and the transcript-generation oracle defined above. This oracle gives I the ability to obtain some number of transcripts of honest executions of the protocol. After reviewing the transcripts, the impersonator must then participate in the three-move protocol with an honest verifier and try to get the verifier to accept.

Definition 2.1 [Security of an identification scheme under passive attacks] Let $\mathcal{ID} = (K, P, V, c)$ be a canonical identification scheme, and let I be an impersonator, st be its state, and k be the security parameter. Define the *advantage* of I as

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{ID},I}(k) \;=\; \Pr[\,\mathbf{Exp}^{\text{imp-pa}}_{\mathcal{ID},I}(k) = 1\,]\;,$$

where the experiment in question is

Experiment $\mathbf{Exp}^{\text{imp-pa}}_{\mathcal{ID},I}(k)$

$(pk, sk) \stackrel{\$}{\leftarrow} K(k)$; $st\|\textsc{Cmt} \stackrel{\$}{\leftarrow} I^{\mathsf{Tr}^{\mathcal{ID}}_{pk,sk,k}}(pk)$; $\textsc{Ch} \stackrel{\$}{\leftarrow} \{0,1\}^{c(k)}$

$\textsc{Rsp} \stackrel{\$}{\leftarrow} I(st, \textsc{Ch})$; $\mathrm{Dec} \leftarrow V(pk, \textsc{Cmt}\|\textsc{Ch}\|\textsc{Rsp})$; return Dec

We say that $\mathcal{ID}$ is *polynomially-secure against impersonation under passive attacks* if $\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{ID},I}(\cdot)$ is negligible for every probabilistic poly(k)-time impersonator I. ▮

SIGNATURE SCHEMES. We recall the standard definition of security of a digital signature scheme under chosen-message attacks (cf. [23]) adapted to the random oracle model as per [8].

The specification of a *digital signature scheme* has the form $\mathcal{DS} = (K, S, Vf, c)$ where K is the *key generation* algorithm, taking input a security parameter $k \in \mathsf{N}$ and returning a public and secret key pair (pk, sk); S is the *signing* algorithm taking input sk and a message $M \in \{0,1\}^*$ to be signed and returning a signature; Vf is the *verification* algorithm taking input pk, a message M and a candidate signature σ for M and returning a boolean decision. The signing and verifying algorithms have oracle access to a function H: $\{0,1\}^* \to \{0,1\}^{c(k)}$ (which in the random oracle model will be a random function) so that c in the scheme description is a function of k whose value is the output-length of the hash function being used. The signing algorithm may be randomized, drawing coins from a space $\mathsf{Coins}_S(k)$, but the verification algorithm is deterministic. It is required that valid signatures are always accepted.

The adversary F —called a forger in this setting— gets the usual signing oracle plus direct access to the random oracle and wins if it outputs a valid signature of a new message. Below, we let $[\{0,1\}^* \to \{0,1\}^c]$ denote the set of all maps from $\{0,1\}^*$ to $\{0,1\}^c$. The notation $H \stackrel{\$}{\leftarrow} [\{0,1\}^* \to \{0,1\}^c]$ is used to mean that we select a hash function H at random from this set. The discussion following the definition clarifies how this random selection from an infinite space is implemented.

Definition 2.2 [Security of a digital signature scheme] Let $\mathcal{DS} = (K, S, \mathcal{V}, c)$ be a digital signature scheme, let F be a forger and k the security parameter. Define the experiment

Experiment $\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{DS},F}(k)$
$H \stackrel{\$}{\leftarrow} [\{0,1\}^* \to \{0,1\}^c]$
$(pk, sk) \stackrel{\$}{\leftarrow} K(k)$; $(M, \sigma) \stackrel{\$}{\leftarrow} F^{S^H_{sk}(\cdot), H(\cdot)}(pk)$; Dec $\leftarrow Vf^H(pk, M, \sigma)$
If M was previously queried to $S^H_{sk}(\cdot)$ Then return 0 Else return Dec

Define the *advantage* of F as

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},F}(k) \;=\; \Pr[\,\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{DS},F}(k) = 1\,]\,.$$

$\mathcal{DS}$ is *polynomially-secure against chosen-message attacks* if $\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},F}(\cdot)$ is negligible for every probabilistic poly(k)-time forger F. ▮

A special convention is needed with regard to how one can measure the time taken by the first step of $\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{DS},F}(k)$ where one picks at random a function H from an infinite space. This selection of the hash function is not viewed as being performed all at once. Rather, the hash function is built dynamically using a table. In particular, for each hash-oracle query M, we check if the entry $H(M)$ exists. If so, we return it. Otherwise, we pick a random element y from $\{0,1\}^c$, make a table entry $H(M) = y$, and return y.

Concrete security issues. In addition to our main results which speak in the usual language of polynomial security, we make concrete security statements so as to better gauge the practical impact of our reductions. Below, we discuss the parameters and conventions used.

When we refer to the running time of an adversary such as an impersonator or forger, we mean the time-complexity of the *entire* associated experiment, including the time taken to pick keys, compute replies to oracle queries, implement a random hash function as described above, and even compute the final outcome of the experiment.

For identification, the parameters of interest are the running time of the adversary and the number of queries q it makes to its transcript oracle. For signatures, the parameters of interest are the forger's running time, the number of sign-oracle queries, denoted q_s, and the number of hash-oracle queries, denoted q_h. All of these are functions of the security parameter k.

All query parameters are bounded by the running time, so if the adversary is polynomial time, all the other parameters are poly(k)-bounded. Thus, they can be ignored in the polynomial-time setting.

3 Equivalence Results

To save space (and avoid repetition), we present straightaway our randomized generalization of the Fiat-Shamir transform. The standard Fiat-Shamir transformation is the special case of the construction below in which the seed length $s(k)$ is 0.

Construction 3.1 [Generalized Fiat-Shamir Transform] Let $\mathcal{ID} = (K, P, V, c)$ be a canonical identification scheme and let $s\colon \mathsf{N} \to \mathsf{N}$ be a function which we call the *seed length*. We associate to these a digital signature scheme $\mathcal{DS} = (K, S, Vf, c)$. It has the same key generation algorithm as the identification scheme, and the output length of the hash function equals the challenge length of the identification scheme. The signing and verifying algorithms are defined as follows:

Algorithm $S^H(sk, M)$	Algorithm $Vf^H(pk, M, \sigma)$
$R \xleftarrow{\$} \{0,1\}^{s(k)}$; $R_P \xleftarrow{\$} \mathsf{Coins}_P(k)$	parse σ as $R \Vert \text{CMT} \Vert \text{RSP}$
$\text{CMT} \leftarrow P(sk; R_P)$	$\text{CH} \leftarrow H(R \Vert \text{CMT} \Vert M)$
$\text{CH} \leftarrow H(R \Vert \text{CMT} \Vert M)$	$\text{Dec} \leftarrow V(pk, \text{CMT} \Vert \text{CH} \Vert \text{RSP})$
$\text{RSP} \leftarrow P(sk, \text{CMT} \Vert \text{CH}; R_P)$	return Dec
return $R \Vert \text{CMT} \Vert \text{RSP}$	

Note that the signing algorithm is randomized, using a random tape whose length is $s(k)$ plus the length of the random tape of the prover. Furthermore, the chosen random seed is included as part of the signature, to make verification possible. ▮

We use the concept of min-entropy [17] to measure how likely it is for a commitment generated by the prover of an identification scheme to collide with a fixed value. This is used to provide a more precise definition of what in the Introduction was referred to as a non-trivial ID scheme.

Definition 3.2 [Min-Entropy of Commitments] Let $\mathcal{ID} = (K, P, V, c)$ be a canonical identification scheme. Let $k \in \mathsf{N}$, and let (pk, sk) be a key pair generated by K on input k. Let $\mathcal{C}(sk) = \{P(sk; R_P) : R_P \in \mathsf{Coins}_P(k)\}$ be the set of commitments associated to sk. We define the maximum probability that a commitment takes on a particular value via

$$\alpha(sk) = \max_{\text{CMT} \in \mathcal{C}(sk)} \left\{ \Pr\left[P(sk; R_P) = \text{CMT} \;:\; R_P \xleftarrow{\$} \mathsf{Coins}_P(k) \right] \right\}$$

Then, the *min-entropy* function associated to $\mathcal{ID}$ is defined as follows:

$$\beta(k) = \min_{sk} \left\{ \log_2 \frac{1}{\alpha(sk)} \right\},$$

where minimum is over all (pk, sk) generated by K on input k. We say that $\mathcal{ID}$ is *non-trivial* if $\beta(\cdot) = \omega(\log(\cdot))$ is super-logarithmic. ▮

We remark that for practical identification schemes, the commitment is drawn uniformly from some set. If the size of this set is $\gamma(\cdot)$ then the min-entropy of the scheme is $\log_2(\gamma(\cdot))$. Non-triviality means that this set has super-polynomial size.

The following theorem considers Construction 3.1 above in the special case where $s(k) = 0$. This case is exactly the Fiat-Shamir transform.

Theorem 3.3 [Equivalence Under Standard Fiat-Shamir Transform] Let $\mathcal{ID} = (K, P, V, c)$ be a non-trivial, canonical identification scheme, and let $\mathcal{DS} = (K, S, Vf, c)$ be the associated signature scheme as per Construction 3.1 with $s(k) = 0$. Then $\mathcal{DS}$ is polynomially-secure against chosen-message attacks in the random oracle model if and only if $\mathcal{ID}$ is polynomially-secure against impersonation under passive attacks. ▮

The non-triviality assumption above can be removed if one applies the generalized FS transform with a seed length that is not zero but which, when added to the min-entropy, results in a super-logarithmic function.

Theorem 3.4 [Equivalence Under Generalized Fiat-Shamir Transform] Let $\mathcal{ID} = (K, P, V, c)$ be a canonical identification scheme, let $s(\cdot)$ be a seed length, and let $\mathcal{DS} = (K, S, Vf, c)$ be the associated signature scheme as per Construction 3.1. Let $\beta(\cdot)$ be the min-entropy function associated to $\mathcal{ID}$. Assume $s(\cdot) + \beta(\cdot) = \omega(\log(\cdot))$. Then $\mathcal{DS}$ is polynomially-secure against chosen-message attacks in the random oracle model if and only if $\mathcal{ID}$ is polynomially-secure against impersonation under passive attacks. ▮

Theorem 3.3 is the special case of Theorem 3.4 in which $s(\cdot) = 0$ and $\beta(\cdot)$ is super-logarithmic. Accordingly, it suffices to prove Theorem 3.4. The proof of Theorem 3.4 follows easily from the two lemmas below. The first lemma relates the exact security of the signature scheme to that of the underlying identification scheme.

Lemma 3.5 [ID $\Rightarrow$ SIG] Let $\mathcal{ID} = (K, P, V, c)$ be a canonical identification scheme, let $s(\cdot)$ be a seed length, and let $\mathcal{DS} = (K, S, Vf, c)$ be the associated signature scheme as per Construction 3.1. Let $\beta(\cdot)$ be the min-entropy function associated to $\mathcal{ID}$. Let F be an adversary attacking $\mathcal{DS}$ in the random oracle model, having time-complexity $t(\cdot)$, making $q_s(\cdot)$ sign-oracle queries and $q_h(\cdot)$ hash-oracle queries. Then there exists an impersonator I attacking $\mathcal{ID}$ such that

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},F}(k) \leq (1+q_h(k)) \cdot \mathbf{Adv}^{\text{imp-pa}}_{\mathcal{ID},I}(k) + \frac{[1+q_h(k)+q_s(k)] \cdot q_s(k)}{2^{s(k)+\beta(k)}} . \tag{1}$$

Furthermore, I has time-complexity $t(\cdot)$ and makes at most $q_s(\cdot)$ queries to its transcript oracle. ▮

Games. Our proof will use code-based game-playing [10]. We recall some background here. A game —look at Figure 3 for an example— has an **Initialize** procedure, procedures to respond to adversary oracle queries, and a **Finalize** procedure. A game G is executed with an adversary A as follows. First, **Initialize** executes and its outputs are the inputs to A. Then, the latter executes, its oracle queries being answered by the corresponding procedures of G. When A terminates, its output becomes the input to the **Finalize** procedure. The output of the latter, denoted G^A, is called the output of the game, and we let "$G^A \Rightarrow y$" denote the event that this game output takes value y. The boolean flag bad is assumed initialized to false. Games G_i, G_j are *identical until* bad if their code differs only in statements that follow the setting of bad to true. For examples, games G_1, G_2 of Figure 3 are identical until bad. The following is the Fundamental Lemma of game-playing of [10].

Lemma 3.6 [10] Let G_i, G_j be identical until bad games, and A an adversary. Then

$$\Pr[\,G_i^A \Rightarrow 1\,] - \Pr[\,G_j^A \Rightarrow 1\,] \;\leq\; \Pr[\,G_i \text{ sets } \mathsf{bad}\,]\;. \;▮$$

The following was stated in [7] and its proof is implicit in [10].

Lemma 3.7 [7] Let G_i, G_j be identical until bad games, and A an adversary. Let $\mathsf{Good}_i, \mathsf{Good}_j$ be the events that bad is never set in games G_i, G_j, respectively. Then,

$$\Pr[\,G_i^A \Rightarrow 1 \;\wedge\; \mathsf{Good}_i\,] \;=\; \Pr[\,G_j^A \Rightarrow 1 \;\wedge\; \mathsf{Good}_j\,]\;. \;▮$$

Proof of Lemma 3.5: We first transform F into a forger A with the following properties. The forger A has time-complexity $t(\cdot) + O(q_s)$, makes at most $1 + q_h(\cdot)$ hash queries, makes at most $q_s(\cdot)$ sign queries, has advantage no less than that of F, and additionally has the following properties:

(1) All of its hash queries are of the form $R \| \text{CMT} \| M$ for some $R \in \{0,1\}^{s(k)}$ and $\text{CMT}, M \in \{0,1\}^*$.

(2) Before outputting forgery $(M, R \| \text{CMT} \| \text{RSP})$, the forger A has made a hash query $R \| \text{CMT} \| M$.

(3) If A outputs $(M, R \| \text{CMT} \| \text{RSP})$, then M was never a sign query.

Let us briefly describe A. On input pk, it runs $F(pk)$. When F makes a hash query x, forger A answers using its own hash oracle if x parses as $R \| \text{CMT} \| \text{RSP}$. Otherwise, it answers with a point chosen at random from $\{0,1\}^{c(k)}$. For a sign query M, it stores M in a set S before answering with the answer from its own sign oracle. Once F outputs a forgery $(M, R \| \text{CMT} \| \text{RSP})$, the forger A makes a hash query $R \| \text{CMT} \| M$. It then checks whether $M \in S$. If so, A returns $(M', R \| \text{CMT} \| \text{RSP})$ for some $M' \notin S$. Otherwise, it returns F's forgery as its own. This does not decrease the advantage because if $M \in S$ then F would not win anyway.

Now, we define an impersonator I against $\mathcal{ID}$. It has input pk and access to a transcript oracle $\mathsf{Tr}^{\mathcal{ID}}_{pk,sk,k}$. It begins with the initialization

$$hc \leftarrow 0 \,;\; sc \leftarrow 0 \,;\; fp \stackrel{\$}{\leftarrow} \{1, \ldots, 1+q_h(k)\}$$
$$\text{For } i = 1, \cdots, q_s(k) \text{ do } \text{TCMT}_i \| \text{TCH}_i \| \text{TRSP}_i \stackrel{\$}{\leftarrow} \mathsf{Tr}^{\mathcal{ID}}_{pk,sk,k}$$

Then, it runs A on input pk.

When A makes a hash query x, the impersonator I returns $\mathsf{HT}[x]$ if this value is defined. Otherwise, it increments hc by one. If $hc \neq fp$, it simply picks $\mathsf{HT}[x]$ at random from $\{0,1\}^{c(k)}$ and returns it to A. Otherwise, it parses x as $R \| \text{CMT}^* \| M$ and sends CMT^* to the verifier as the first move of a protocol execution, receiving back a challenge CH^* which it stores as $\mathsf{HT}[fp]$ and also returns to A as the response to hash query x.

When A makes a sign query M, the impersonator I increments sc, picks R at random from $\{0,1\}^{s(k)}$, sets $\mathsf{HT}[R \| \text{TCMT}_{sc} \| M]$ to TCH_{sc} and returns $R \| \text{TCMT}_{sc} \| \text{TRSP}_{sc}$ to A as the signature. Note that it might overwrite $\mathsf{HT}[R \| \text{TCMT}_{sc} \| M]$ in case the latter was defined, which could make the simulation erroneous, but our analysis will show this seldom happens. With the hash of $R \| \text{TCMT}_{sc} \| M$ defined as TCH_{sc}, however, the signature is valid.

Finally, A halts with output a forgery $(M, R \| \text{CMT} \| \text{RSP})$. The impersonator I now sends RSP to the verifier as its final protocol move and halts.

We claim that

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{ID},I}(k) \;\geq\; \frac{1}{1+q_h(k)} \cdot \left(\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},F}(k) - \frac{[1+q_h(k)+q_s(k)] \cdot q_s(k)}{2^{s(k)+\beta(k)}} \right) . \tag{2}$$

Then, Equation (1) follows.

The analysis uses games G_0–G_5 of Figures 3 and 4. For $0 \leq i \leq 5$, let Good_i denote the event that Game G_i never sets bad. We now state a chain of inequalities which we will justify below:

$$\begin{aligned}
\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{ID},I}(k) \;&\geq\; \Pr[\, G_0^A \Rightarrow 1 \;\wedge\; \mathsf{Good}_0 \,] && (3)\\
&=\; \Pr[\, G_1^A \Rightarrow 1 \;\wedge\; \mathsf{Good}_1 \,] && (4)\\
&=\; \Pr[\, G_2^A \Rightarrow 1 \;\wedge\; \mathsf{Good}_2 \,] && (5)\\
&=\; \Pr[\, G_2^A \Rightarrow 1 \,] \cdot \Pr[\, \mathsf{Good}_2 \,] && (6)
\end{aligned}$$

procedure Initialize Game G_0

000 $(pk, sk) \xleftarrow{\$} K(k)\,;\ hc \leftarrow 0\,;\ sc \leftarrow 0$
001 $fp \xleftarrow{\$} \{1, \ldots, 1+q_h(k)\}$
002 $\textsc{Ch}^* \xleftarrow{\$} \{0,1\}^{c(k)}$
003 For $i = 1, \cdots, q_s(k)$ do
004 $\quad R_P^i \xleftarrow{\$} \mathsf{Coins}_P(k)$
005 $\quad \textsc{TCmt}_i \leftarrow P(sk; R_P^i)$
006 $\quad \textsc{TCh}_i \xleftarrow{\$} \{0,1\}^{c(k)}$
007 $\quad \textsc{TRsp}_i \leftarrow P(sk, \textsc{TCmt}_i \| \textsc{TCh}_i; R_P^i)$
008 Return pk

procedure $\mathrm{H}(x)$

010 If $\mathsf{HT}[x] = \perp$ Then
011 $\quad hc \leftarrow hc + 1\,;\ \mathsf{QT}[hc] \leftarrow x$
012 $\quad$ If $hc \neq fp$ Then
013 $\quad\quad y \xleftarrow{\$} \{0,1\}^{c(k)}\,;\ \mathsf{HT}[x] \leftarrow y$
014 $\quad$ Else $\mathsf{HT}[x] \leftarrow \textsc{Ch}^*$
015 return $\mathsf{HT}[x]$

procedure $\textsc{Sign}(M)$

020 $sc \leftarrow sc + 1\,;\ R \xleftarrow{\$} \{0,1\}^s$
021 $x \leftarrow R \| \textsc{TCmt}_{sc} \| M$
022 $\mathsf{HT}[x] \leftarrow \textsc{TCh}_{sc}$
023 return $R \| \textsc{TCmt}_{sc} \| \textsc{TRsp}_{sc}$

procedure Finalize(M, σ)

030 Parse σ as $R \| \textsc{Cmt} \| \textsc{Rsp}$
031 Let i be such that $\mathsf{QT}[i] = R \| \textsc{Cmt} \| M$
032 If $i \neq fp$ Then $\mathsf{bad} \leftarrow \mathsf{true}$
033 return $V(pk, \textsc{Cmt} \| \textsc{Ch}^* \| \textsc{Rsp})$

procedure Initialize Games $\boxed{G_1}$ / G_2

100 $(pk, sk) \xleftarrow{\$} K(k)\,;\ hc \leftarrow 0\,;\ sc \leftarrow 0$
101 For $i = 1, \cdots, q_s(k)$ do
102 $\quad R_P^i \xleftarrow{\$} \mathsf{Coins}_P(k)$
103 $\quad \textsc{TCmt}_i \leftarrow P(sk; R_P^i)$
104 $\quad \textsc{TCh}_i \xleftarrow{\$} \{0,1\}^{c(k)}$
105 $\quad \textsc{TRsp}_i \leftarrow P(sk, \textsc{TCmt}_i \| \textsc{TCh}_i; R_P^i)$
106 Return pk

procedure $\mathrm{H}(x)$

110 If $\mathsf{HT}[x] = \perp$ Then
111 $\quad hc \leftarrow hc + 1\,;\ \mathsf{QT}[hc] \leftarrow x$
112 $\quad \mathsf{HT}[x] \xleftarrow{\$} \{0,1\}^{c(k)}$
113 return $\mathsf{HT}[x]$

procedure $\textsc{Sign}(M)$

120 $sc \leftarrow sc + 1\,;\ R \xleftarrow{\$} \{0,1\}^s$
121 $x \leftarrow R \| \textsc{TCmt}_{sc} \| M$
122 $\mathsf{HT}[x] \leftarrow \textsc{TCh}_{sc}$
123 return $R \| \textsc{TCmt}_{sc} \| \textsc{TRsp}_{sc}$

procedure Finalize(M, σ)

130 Parse σ as $R \| \textsc{Cmt} \| \textsc{Rsp}$
131 Let i be such that $\mathsf{QT}[i] = R \| \textsc{Cmt} \| M$
132 $\textsc{Ch}^* \leftarrow \mathsf{HT}[\mathsf{QT}[i]]$
133 $fp \xleftarrow{\$} \{1, \ldots, 1+q_h(k)\}$
134 If $i \neq fp$ Then
135 $\quad \mathsf{bad} \leftarrow \mathsf{true}$ $\boxed{;\ \textsc{Ch}^* \leftarrow \mathsf{HT}[\mathsf{QT}[fp]]}$
136 return $V(pk, \textsc{Cmt} \| \textsc{Ch}^* \| \textsc{Rsp})$

Figure 3: Games G_0, G_1, and G_2.

Game G_0 simulates the execution environment of I. The interaction with the verifier is not explicit. Instead, the verifier's challenge $\textsc{Ch}^*$ is chosen in line 002 of **Initialize**. Lines 004–007 generate the transcripts that play the role of the ones that I obtains from its oracle, but G_0 generates them explicitly using the secret key chosen at line 000. Parsing $\mathsf{QT}[fp]$ as $R\|\textsc{Cmt}^*\|M$, the value $\textsc{Cmt}^*$ plays the role of the commitment sent by I to the verifier, but is not made explicit. If i (generated at line 031) equals fp, then I's conversation with the verifier is $\textsc{Cmt}\|\textsc{Ch}^*\|\textsc{Rsp}$. (In this case, $\textsc{Cmt} = \textsc{Cmt}^*$.) So I succeeds when $V(pk, \textsc{Cmt}\|\textsc{Ch}^*\|\textsc{Rsp}) = 1$. (We do not have to check whether M in the forgery is new due to the property **3** above.) We have just justified Equation (3).

Since $\textsc{Ch}^*$ in G_0 is just a random value, game G_1 does not choose it in **Initialize**, but instead assigns it the value $\mathsf{HT}[fp]$ in **Finalize**. Lines 132, 134, and 135 do this because the boxed code is included in G_1. Since fp is now not used in replying to hash queries, G_1 delays its choice until line 133. This explains Equation (4).

procedure Initialize Games $\boxed{G_3}$ / G_4

300 $(pk, sk) \xleftarrow{\$} K(k)$; $hc \leftarrow 0$; $sc \leftarrow 0$
301 Return pk

procedure $\mathrm{H}(x)$

310 If $\mathsf{HT}[x] = \perp$ Then
311 $\quad hc \leftarrow hc + 1$; $\mathsf{QT}[hc] \leftarrow x$
312 $\quad \mathsf{HT}[x] \xleftarrow{\$} \{0,1\}^{c(k)}$
313 return $\mathsf{HT}[x]$

procedure $\textsc{Sign}(M)$

320 $sc \leftarrow sc + 1$; $R \xleftarrow{\$} \{0,1\}^{s}$
321 $R_P^i \xleftarrow{\$} \mathsf{Coins}_P(k)$
322 $\mathrm{T}\textsc{Cmt}_{sc} \leftarrow P(sk; R_P^i)$; $\mathrm{T}\textsc{Ch}_{sc} \xleftarrow{\$} \{0,1\}^{c(k)}$
323 $x \leftarrow R \| \mathrm{T}\textsc{Cmt}_{sc} \| M$
324 If $\mathsf{HT}[x] \neq \perp$ Then
325 $\quad \mathsf{bad} \leftarrow \mathsf{true}$ $\boxed{;\ \mathrm{T}\textsc{Ch}_{sc} \leftarrow \mathsf{HT}[x]}$
326 $\mathrm{T}\textsc{Rsp}_{sc} \leftarrow P(sk, \mathrm{T}\textsc{Cmt}_{sc} \| \mathrm{T}\textsc{Ch}_{sc}; R_P^i)$
327 $\mathsf{HT}[x] \leftarrow \mathrm{T}\textsc{Ch}_{sc}$
328 return $R \| \mathrm{T}\textsc{Cmt}_{sc} \| \mathrm{T}\textsc{Rsp}_{sc}$

procedure Finalize(M, σ)

330 Parse σ as $R \| \textsc{Cmt} \| \textsc{Rsp}$
331 Let i be such that $\mathsf{QT}[i] = R \| \textsc{Cmt} \| M$
332 $\textsc{Ch}^* \leftarrow \mathsf{HT}[\mathsf{QT}[i]]$
333 return $V(pk, \textsc{Cmt} \| \textsc{Ch}^* \| \textsc{Rsp})$

procedure Initialize Game G_5

500 $(pk, sk) \xleftarrow{\$} K(k)$; $hc \leftarrow 0$; $sc \leftarrow 0$
501 Return pk

procedure $\mathrm{H}(x)$

510 If $\mathsf{HT}[x] = \perp$ Then
511 $\quad hc \leftarrow hc + 1$; $\mathsf{QT}[hc] \leftarrow x$
512 $\quad \mathsf{HT}[x] \xleftarrow{\$} \{0,1\}^{c(k)}$
513 return $\mathsf{HT}[x]$

procedure $\textsc{Sign}(M)$

520 $sc \leftarrow sc + 1$; $R \xleftarrow{\$} \{0,1\}^{s}$
521 $R_P^i \xleftarrow{\$} \mathsf{Coins}_P(k)$
522 $\mathrm{T}\textsc{Cmt}_{sc} \leftarrow P(sk; R_P^i)$
523 $x \leftarrow R \| \mathrm{T}\textsc{Cmt}_{sc} \| M$
524 If $\mathsf{HT}[x] = \perp$ Then $\mathsf{HT}[x] \xleftarrow{\$} \{0,1\}^{c(k)}$
525 $\mathrm{T}\textsc{Ch}_{sc} \leftarrow \mathsf{HT}[x]$
526 $\mathrm{T}\textsc{Rsp}_{sc} \leftarrow P(sk, \mathrm{T}\textsc{Cmt}_{sc} \| \mathrm{T}\textsc{Ch}_{sc}; R_P^i)$
527 return $R \| \mathrm{T}\textsc{Cmt}_{sc} \| \mathrm{T}\textsc{Rsp}_{sc}$

procedure Finalize(M, σ)

540 Parse σ as $R \| \textsc{Cmt} \| \textsc{Rsp}$
541 Let i be such that $\mathsf{QT}[i] = R \| \textsc{Cmt} \| M$
542 $\textsc{Ch}^* \leftarrow \mathsf{HT}[\mathsf{QT}[i]]$
543 return $V(pk, \textsc{Cmt} \| \textsc{Ch}^* \| \textsc{Rsp})$

Figure 4: Game G_3, G_4, and G_5.

Games G_1, G_2 are identical-until-bad, so Equation (5) follows from Lemma 3.7. However, in Game G_2, the boxed statement at line 135 is absent. So *fp* is not used in determining the game output. This means the events Good_2 and "$G_2^A \Rightarrow 1$" are independent, justifying Equation (6).

Now from lines 133–135 of G_2, it is clear that

$$\Pr[\,\mathsf{Good}_2\,] = \frac{1}{1+q_h(k)}\ .$$

The **Finalize** procedure of G_3 has the same output as that of G_2 but drops lines 133–135 that are now redundant. The other change it makes is to delay the choices of lines 101–105 until they are needed in answering sign queries. But given that the boxed code of line 325 is included, these replies are the same as in G_2. The setting of bad does not affect the game output. So, we have

$$\begin{aligned} \Pr[\,G_2^A \Rightarrow 1\,] &= \Pr[\,G_3^A \Rightarrow 1\,] \\ &\geq \Pr[\,G_4^A \Rightarrow 1\,] - \Pr[\,G_4^A \text{ sets } \mathsf{bad}\,] \end{aligned} \tag{7}$$

where Equation (7) follows from Lemma 3.6 because G_3, G_4 are identical-until-bad. The probability that the i-th sign query sets bad in G_4 is at most

$$\frac{1+q_h(k)+(i-1)}{2^{s(k)+\beta(k)}} .$$

So,

$$\begin{aligned} \Pr[\, G_4^A \text{ sets } \mathsf{bad}\,] &\le \sum_{i=1}^{q_s(k)} \frac{1+q_h(k)+(i-1)}{2^{s(k)+\beta(k)}} \\ &= \frac{q_h(k)q_s(k)+q_s(k)(q_s(k)+1)/2}{2^{s(k)+\beta(k)}} \\ &\le \frac{[1+q_h(k)+q_s(k)]q_s(k)}{2^{s(k)+\beta(k)}} . \end{aligned} \tag{8}$$

Given that the boxed code of line 325 is not present in G_4, the code to reply to sign queries is equivalent to that in G_5 barring no longer setting bad. The latter does not affect the game output, so

$$\Pr[\, G_4^A \Rightarrow 1\,] = \Pr[\, G_5^A \Rightarrow 1\,] .$$

But G_5 captures the experiment defining the advantage of A and so

$$\begin{aligned} \Pr[\, G_5^A \Rightarrow 1\,] &= \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},A}(k) & (9) \\ &\ge \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},F}(k) & (10) \end{aligned}$$

the last by the properties of A stated above. Putting together Equations (3), (4), (5), (6), (7), (8), (9), and (10) yields Equation (2). ▮

Going in the opposite direction, the following lemma relates the security of the identification scheme to that of the signature scheme derived from it. In fact, it says that if the signature scheme is secure then so is the identification scheme (regardless of the min-entropy of the ID scheme).

Lemma 3.8 [ID ⇐ SIG] Let $\mathcal{ID} = (K, P, V, c)$ be a canonical identification scheme, let $s(\cdot)$ be a seed length, and let $\mathcal{DS} = (K, S, Vf, c)$ be the associated signature scheme as per Construction 3.1. Let I be an adversary attacking $\mathcal{ID}$, having time-complexity $t(\cdot)$ and making $q(\cdot)$ queries to its transcript oracle. Then, in the random oracle model, there exists a forger F attacking $\mathcal{DS}$ such that

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{ID},I}(k) \le \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},F}(k) . \tag{11}$$

Furthermore, F has time-complexity $t(\cdot)$, makes at most $q(\cdot)$ queries to its sign-oracle and at most $q(\cdot)+1$ queries to its hash-oracle. ▮

Proof of Lemma 3.8: Forger F is presented in Figure 5. It runs the impersonator I as a subroutine, answering the latter's transcript oracle queries via its signing oracle. When I outputs (some state information and) a commitment, F increments M, picks a random seed R, and defines the verifier's challenge via a hash query. It provides this to I, obtains a response Rsp, and uses the latter in its forgery. The messages used in the algorithm are generated by incrementing a counter and interpreting its value as a string. This ensures that the messages are always new, and thus, the forgery is that of a message that has never been queried to the signing oracle before. ▮

Algorithm $F^{S^H_{sk}(\cdot),H(\cdot)}(pk)$

$M \leftarrow 0$ ▷ *Initialize the message*

run $I(pk)$ answering to its transcript queries as follows:

$M \leftarrow M + 1$ ▷ *Generate a new message*

$x \xleftarrow{\$} S^H_{sk}(M)$ ▷ *M is interpreted as a string*

parse x as $R \| \textsc{Cmt} \| \textsc{Rsp}$

$\textsc{Ch} \leftarrow H(R \| \textsc{Cmt} \| M)$

return $\textsc{Cmt} \| \textsc{Ch} \| \textsc{Rsp}$ to I

until I outputs $st \| \textsc{Cmt}$ ▷ *Phase 1*

$M \leftarrow M + 1$ ▷ *Generate a new message*

$R \xleftarrow{\$} \{0,1\}^{s(k)}$

$\textsc{Ch} \leftarrow H(R \| \textsc{Cmt} \| M)$

give $(st, \textsc{Ch})$ to I ▷ *Phase 2*

get $\textsc{Rsp}$ from I ▷ *Phase 3*

return $(M, R \| \textsc{Cmt} \| \textsc{Rsp})$ ▷ *Output a forgery*

Figure 5: The forger algorithm F for the proof of Lemma 3.8.

4 Separations among Security Assumptions

In this section, we justify the claimed separations among the security conditions in Figure 1. Specifically, we give an example of an ID scheme that is secure against impersonation under passive attack but is not honest-verifier zero-knowledge, and also an example of an ID scheme that is secure against impersonation under passive attack and is not a proof of knowledge. (In this section, proof of knowledge means proof of knowledge of the secret key. More precisely, it refers to some underlying witness-relation $R(pk, sk)$ depending on the protocol.) Since the **PS** and **OO** assumptions include either an assumption of honest verifier zero-knowledge or an assumption of proof of knowledge, this implies that there exists an identification scheme secure against impersonation under passive attack that is not **PS** secure, and there exists an identification scheme secure against impersonation under passive attack that is not **OO** secure, justifying two of the claimed separations in Figure 1, and showing that our assumption on the ID scheme is strictly weaker than previous ones used to prove security of the signature scheme.

Furthermore, this also justifies two more separations claimed in Figure 1, namely that the signature scheme could be secure even if the ID scheme is not **PS** secure or **OO** secure. This follows simply by logic, because if we assume that security of the signature scheme implies, say, **PS**-security of the ID scheme, the existing arrows say that security against impersonation under passive attack implies **PS**-security, which we know from the above to not be true. The analogous argument applies in the case of **OO**.

We now proceed to the examples. Shoup notes that the 2^m-th root identification (a special case of the identification scheme of Ong and Schnorr [32]) is provably not a proof of knowledge if factoring is hard [37]. However, he shows that this scheme is secure against impersonation under active (and hence certainly under passive) attacks if factoring is hard. Since 2^m-th root scheme is a non-trivial canonical identification scheme, this yields the following:

Proposition 4.1 If factoring is hard, then there exists a non-trivial canonical identification scheme that is secure against impersonation under passive attacks but is not a proof of knowledge. ▮

Similarly, we show that there exists an identification scheme that is secure against impersonation under passive attacks yet is not honest verifier zero-knowledge. We take the following approach in constructing such an identification scheme. We begin with a canonical identification secure against impersonation under passive attacks and modify it so that it remains secure against impersonation under passive attacks but is not zero-knowledge. A detailed construction is presented in the proof below. The example we construct, though contrived, makes the point that zero-knowledge is not strictly necessary in a secure identification scheme. The following proposition states this more precisely.

Proposition 4.2 If factoring is hard, then there exists a non-trivial canonical identification scheme that is secure against impersonation under passive attacks but is not honest-verifier zero-knowledge. ▮

Proof of Proposition 4.2: Given a secure, non-trivial canonical identification scheme $\mathcal{ID} = (K, P, V, c)$ which has been proven secure, we modify it as follows. We extend the given scheme's key generation algorithm K so that, upon input of the security parameter k, it generates an additional value which we call N', which is the product of two large random primes p' and q', each of length k bits. The values p' and q' are now part of the secret key, and their product N' is added to the public key. Finally, we modify the prover algorithm P so that in addition to any other values sent in the response step, the values p' and q' (that is, the factorization of N') are also revealed. We refer to this modified identification scheme as $\mathcal{ID}'$.

We claim that the scheme $\mathcal{ID}'$ is a secure identification scheme. Since revealing the factorization of N' does not interfere with the security of the underlying scheme, the security of $\mathcal{ID}'$ follows directly from that of $\mathcal{ID}$. Furthermore, $\mathcal{ID}'$ is not a zero-knowledge scheme. The knowledge revealed in the scheme is the factorization of N'. Based on the assumption that factoring is hard, it is clear that any computationally bounded adversary could not generate a transcript for the scheme without knowledge of the secret key. ▮

5 Extension to forward security

We prove an extension of Theorem 3.4 to the case where the security requirement is forward security.

Background. Forward-secure signature schemes [5, 2] evolve the signer's secret key with time while leaving the public key fixed. Exposure of the secret key in some time period should not aid the adversary in forging signatures of new messages relative to previous time periods. The designs of forward-secure signature schemes of Bellare and Miner [5] and Abdalla and Reyzin [1] are obtained by first designing forward-secure identification schemes and then applying the Fiat-Shamir transform. The result we prove here generalizes and modularizes such transforms, facilitating the design and analysis of further constructs of this type.

Canonical forward-secure identification schemes. We consider key-evolving identification schemes. The operation of the scheme is divided into time periods, where a different secret is used in each time period. The public key remains the same in every time period. A canonical key-evolving identification scheme is a three-move protocol in which the verifier's only move is to pick and send a random challenge to the prover. Unlike canonical identification schemes with fixed keys, the verifier's final decision, though still deterministic, is not only a function of the conversation with the prover and the public key, but also a function of the the index of the current time period. We say that a canonical key-evolving identification scheme is *forward-secure* if it is infeasible for a passive adversary, even with access to the current secret key, to impersonate the prover with respect to an honest verifier in any of the prior time periods.

As pointed out by Bellare and Miner [5], forward-secure identification schemes are artificial constructs since, due to the online nature of identification protocols, the kind of attack we withstand in this case cannot exist in reality. Nevertheless, the schemes are still very useful in the design of efficient forward-secure signature schemes. We present a formal definition of a key-evolving identification scheme and what it means for it to be forward-secure below.

The specification of a *canonical key-evolving identification scheme* has the form $\mathcal{FID} = (K, P, Vid, Up, c, T)$. T is a function of the security parameter $k \in \mathsf{N}$ indicating the total number of time periods for which the scheme will operate. K is the *key generation* algorithm, taking input k and $T(k)$ and returning a pair (pk, sk) consisting of the public key and the base (initial) secret key. P is the *prover* algorithm taking input the current secret key sk_j, the index j of the current time period, and the current conversation prefix to return the next message to send to the verifier. Vid is a deterministic algorithm taking input pk, the current time period index j, and a complete conversation transcript to return a boolean decision Dec on whether or not to accept. The probabilitic algorithm Up is an update algorithm taking input the old secret sk_{j-1} and time index j and returning the new secret key sk_j. The old secret key is erased after the new one is computed. c is a function of k indicating the length of the verifier's challenge. As in standard canonical identification schemes, we also assume that pk and each sk_j contain the security parameter k. To $\mathcal{FID}$ and to each triple (pk, sk_j, j), consisting respectively of the public key, secret key for time period j and time index j, we associate a randomized *transcript generation oracle* which takes no inputs and returns a random transcript of an "honest" execution, namely:

Function $\mathsf{Tr}^{\mathcal{FID}}_{pk,sk_j,j,k}$
$\quad R_P \stackrel{\$}{\leftarrow} \mathsf{Coins}_P(k)$
$\quad \textsc{Cmt} \leftarrow P(sk_j, j; R_P)$; $\textsc{Ch} \stackrel{\$}{\leftarrow} \{0,1\}^{c(k)}$; $\textsc{Rsp} \leftarrow P(sk_j, j, \textsc{Cmt}\|\textsc{Ch}; R_P)$;
$\quad$ return $\textsc{Cmt}\|\textsc{Ch}\|\textsc{Rsp}$

Let (pk, sk_0) be the base secret-public key pair initially returned by K on inputs k and $T(k)$ and let sk_j be the secret key in time period j obtained via j iterations of the update algorithm, Up. The scheme must still obey a standard completeness requirement, namely that for every triple (pk, sk_j, j), obtained as above on input k, we have

$$\Pr\left[\ Vid(pk, j, \textsc{Cmt}\|\textsc{Ch}\|\textsc{Rsp}) = 1 \ : \ \textsc{Cmt}\|\textsc{Ch}\|\textsc{Rsp} \stackrel{\$}{\leftarrow} \mathsf{Tr}^{\mathcal{FID}}_{pk,sk_j,j,k}\ \right] \ = \ 1\ .$$

In the forward-security model, the adversary I —also called an impersonator in this setting— against the forward security of a key-evolving identification scheme operates in three phases: $\mathsf{passive}$, the passive phase; $\mathsf{breakin}$, the break-in phase; and imp, the impersonation phase. In the passive phase, the adversary I is given the public key pk, the index j of the current time period, and the ability to obtain some number of transcripts of honest executions of the protocol for that time period. At the end of each time period, the impersonator can choose to remain in the $\mathsf{passive}$ phase or switch to a $\mathsf{breakin}$ phase. When it decides to do so, it then receives the secret key sk_j for the current period j and then switches to the impersonation phase, imp. In this last phase, it must then try to impersonate the prover for some time period b *prior* to that of the break-in. The adversary I is considered successful if the verifier accepts at the end of the protocol.

Definition 5.1 [Forward security of an identification scheme under passive attacks] Let $\mathcal{FID} = (K, P, \mathit{Vid}, \mathit{Up}, c, T)$ be a canonical key-evolving identification scheme, and let I be an impersonator and k be the security parameter. Define the experiment

$$
\begin{array}{l}
\text{Experiment } \mathbf{Exp}^{\text{fs-imp-pa}}_{\mathcal{FID},I}(k) \\
\quad (pk, sk_0) \stackrel{\$}{\leftarrow} K(k, T(k))\ ;\ j \leftarrow 0 \\
\quad \text{repeat} \\
\qquad j \leftarrow j+1\ ;\ sk_j \stackrel{\$}{\leftarrow} \mathit{Up}(sk_{j-1}, j) \\
\qquad (d, st) \stackrel{\$}{\leftarrow} I^{\mathsf{Tr}^{\mathcal{FID}}_{pk,sk_j,j,k}}(\mathsf{passive}, pk, st) \\
\quad \text{until } d = \mathsf{breakin} \text{ or } j = T(k) \\
\quad (st, \text{CMT}, b) \stackrel{\$}{\leftarrow} I(\mathsf{imp}, sk_j, st) \\
\quad \text{CH} \stackrel{\$}{\leftarrow} \{0,1\}^{c(k)} \\
\quad \text{RSP} \stackrel{\$}{\leftarrow} I(st, \text{CH}) \\
\quad \text{If } 1 \le b < j \text{ and } \mathit{Vid}(pk, b, \text{CMT}\|\text{CH}\|\text{RSP}) = 1 \\
\qquad \text{Then Dec} \leftarrow 1 \text{ Else Dec} \leftarrow 0 \\
\quad \text{return Dec}
\end{array}
$$

Define the *advantage* of I as

$$\mathbf{Adv}^{\text{fs-imp-pa}}_{\mathcal{FID},I}(k) \quad = \quad \Pr[\,\mathbf{Exp}^{\text{fs-imp-pa}}_{\mathcal{FID},I}(k) = 1\,]\ .$$

We say that $\mathcal{FID}$ is *polynomially-forward-secure against impersonation under passive attacks* if $\mathbf{Adv}^{\text{fs-imp-pa}}_{\mathcal{FID},I}(\cdot)$ is negligible for every probabilistic poly(k)-time impersonator I. ▮

Forward-secure signature schemes. A forward-secure signature scheme is in essence a key-evolving signature scheme in which the secret key is updated periodically. As in standard signature schemes, the public key remains the same throughout the lifetime of the scheme. In each time period, a different secret key is used to sign messages. The verification algorithm checks not only the validity of a signature, but also the particular time period in which it was generated. At the end of each time period, an update algorithm is run to compute the new secret key from the current one, which is then erased. Informally, we say that a key-evolving signature scheme is *forward-secure* under chosen-message attack if it is infeasible for an adversary, even with access to the secret key for the current period and to previously signed messages of its choice, cannot forge signatures for a past time period. For a formal definition of a key-evolving signature scheme and what it means for it to be forward-secure, see below

We recall the definition of forward security of a signature scheme under chosen-message attack in the random oracle model (cf. [5]). The specification of a *key-evolving digital signature scheme* has the form $\mathcal{FSDS} = (K, S, \mathit{VSig}, \mathit{Up}, c, T)$. T is a function of the security parameter $k \in \mathsf{N}$ indicating the total number of time periods for which the scheme will operate. K is the *key generation* algorithm, taking input a k and $T(k)$ and returning a pair (pk, sk_0), consisting of the public key and base secret key. S is the *signing* algorithm taking input sk_j, the index j of the current time period, and a message $M \in \{0,1\}^*$ to be signed and returning a tuple $\langle \sigma, j \rangle$ consisting of the signature and the time index. VSig is the *verification* algorithm taking input pk, a time index j, a message M, and a candidate signature σ for M with respect to time period j and returning a boolean decision. The probabilistic algorithm Up is an *update* algorithm taking input the old secret sk_{j-1} and time index j and returning the new secret key sk_j. The old secret key is erased after the new one is computed. As in the case of standard signature schemes, the signing and verifying algorithms have oracle access to a function $H\colon \{0,1\}^* \to \{0,1\}^{c(k)}$ so that c in the scheme description is a function of k whose value is the output-length of the hash function being used. The signing algorithm may be randomized, drawing coins from a space $\mathsf{Coins}_S(k)$, but the verification algorithm is deterministic. It is required that valid signatures are always accepted.

In the forward-security model, the adversary —also called forger— knows the total number $T(k)$ of time periods, the current time period j, and the public key pk and runs in three phases: cma, the chosen message attack phase; $\mathsf{breakin}$, the break-in phase; and forge, the forgery phase. Like in standard signature schemes, during the cma phase, the adversary is given access to a signing oracle for the current time period. At the end of each time period, the adversary chooses to either remain in the cma phase or switch to a $\mathsf{breakin}$ phase. In the latter case, the adversary is then given the secret key sk_j for the current time period j. We consider the adversary successful if it outputs a valid signature of a new message with respect to some time period $b < j$.

Definition 5.2 [Forward security of a digital signature scheme] Let $\mathcal{FSDS} = (K, S, \mathcal{V}, c, T)$ be a digital signature scheme, let F be a forger and k the security parameter. Define the experiment

Experiment $\mathbf{Exp}^{\text{fs-uf-cma}}_{\mathcal{FSDS},F}(k)$
$H \xleftarrow{\$} [\{0,1\}^* \rightarrow \{0,1\}^c]$
$(pk, sk_0) \xleftarrow{\$} K(k, T(k))$
$j \leftarrow 0$
repeat
$j \leftarrow j+1$; $sk_j \xleftarrow{\$} Up(sk_{j-1}, j)$
$(d, st) \xleftarrow{\$} F^{S^H_{sk_j}(\cdot), H(\cdot)}(pk, T(k), j)$
until $d = \mathsf{breakin}$ or $j = T$
$(M, \langle \sigma, b\rangle) \xleftarrow{\$} F^{H(\cdot)}(\mathsf{forge}, sk_j, st)$
Dec $\leftarrow VSig^H(pk, M, \sigma, b)$
If M was not previously queried to $S^H_{sk_b}(\cdot)$ and $1 \leq b < j$ Then return Dec Else return 0

Define the *advantage* of F as

$$\mathbf{Adv}^{\text{fs-uf-cma}}_{\mathcal{FSDS},F}(k) = \Pr[\,\mathbf{Exp}^{\text{fs-uf-cma}}_{\mathcal{FSDS},F}(k) = 1\,]$$

We say that $\mathcal{FSDS}$ is *polynomially-forward-secure against chosen-message attacks* if $\mathbf{Adv}^{\text{fs-uf-cma}}_{\mathcal{FSDS},F}(\cdot)$ is negligible for every probabilistic poly(k)-time forger F. ▌

The Equivalence. Our transformation of key-evolving ID schemes into key-evolving signature schemes follows the same paradigm of Construction 3.1, in which the challenge becomes the output of a hash function H. The main difference with respect to that construction is that the secret key is no longer fixed but varies according to the time period. As a result, the current time index j is also given as input to the signing algorithm and attached to the signature to allow for correct verification. The current time index j is also added to the input of the hash function, which now becomes $j \| R \| \textsc{Cmt} \| M$. The update algorithm of the key-evolving signature scheme is exactly the same as that of the identification scheme on which it is based. The following theorem, where min-entropy is defined in a manner similar to that for canonical identification schemes, states precisely the equivalence with regard to forward security of the key-evolving ID scheme and the associated key-evolving signature scheme.

Theorem 5.3 [Forward security equivalence theorem] Let $\mathcal{FID} = (K, P, Vid, c, T)$ be a canonical key-evolving identification scheme, let $s(\cdot)$ be a seed length, and let $\mathcal{FSDS} = (K, S,$

$VSig, c, T)$ be the associated key-evolving signature scheme as per the new construction described above. Let $\beta(\cdot)$ be the min-entropy function associated to $\mathcal{FID}$ and assume $s(\cdot) + \beta(\cdot) = \omega(\log(\cdot))$. Then $\mathcal{FSDS}$ is polynomially-forward-secure against chosen-message attack in the random oracle model if and only if $\mathcal{FID}$ is polynomially-forward-secure against impersonation under passive attacks. ▮

As in the standard case, we prove each direction of the "if and only if" statement separately. The following lemma says that if the key-evolving identification scheme is forward-secure then so is the key-evolving signature scheme in the random oracle model.

Lemma 5.4 Let $\mathcal{FID} = (K, P, Vid, c, T)$ be a canonical key-evolving identification scheme, let $s(\cdot)$ be a seed length, and let $\mathcal{FSDS} = (K, S, VSig, c, T)$ be the associated key-evolving signature scheme as per the new construction described above. Let $\beta(\cdot)$ be the min-entropy function associated to $\mathcal{FID}$. Let F be an adversary attacking $\mathcal{FSDS}$ in the random oracle model, having time-complexity $t(\cdot)$, making at most $q_s(\cdot)$ sign-oracle queries per time period and at most $q_h(\cdot)$ hash-oracle queries overall. Then there exists an impersonator I attacking $\mathcal{FID}$ such that

$$
\begin{aligned}
&\mathbf{Adv}^{\text{fs-uf-cma}}_{\mathcal{FSDS},F}(k) \\
&\leq \quad (T(k)+1) \cdot [1+q_h(k)] \cdot \mathbf{Adv}^{\text{fs-imp-pa}}_{\mathcal{FID},I}(k) + \frac{[1+q_h(k)+q_s(k)T(k)] \cdot q_s(k)T(k)}{2^{s(k)+\beta(k)}} .
\end{aligned}
$$

Furthermore, I has time-complexity $t(\cdot)$ and makes at most $q_s(\cdot)$ transcript-oracle queries per time period. ▮

Proof of Lemma 5.4: The proof we present here is a generalization of the proof given in [5] to the case of randomized transformations of ID schemes into signature schemes. Let F be an adversary against the forward security of signature scheme $\mathcal{FSDS}$. Our goal is to construct an impersonator algorithm I against the forward security of $\mathcal{FID}$, using F as a subroutine, and relate its advantage to that of F. Recall that F runs in three phases: cma, breakin, and forge. During the chosen-message attack, cma, F has access to a hash oracle as well as a signing oracle for the current time period. Hence, we need to simulate these oracles. We should also be prepared to feed F with the secret key of the current time period when it decides to break in, switching to breakin phase.

Our algorithm I works in three phases: the passive phase, passive; the break-in phase, breakin; and the impersonation phase, imp. Similarly to the proof of Lemma 3.5, our strategy in constructing I is also to guess which of F's hash queries contains the message on which F will attempt to forge and use that to impersonate the prover. There is one important difference, though, in our case. I cannot wait for F's decision to break in to decide itself when to break in. This is so because all hash queries can be done at the very beginning of F's cma phase and imp needs to interact with the verifier in order to get a challenge $\textsc{Ch}$ to answer the crucial hash query. But that can only be done after imp phase is over. Hence, besides guessing which one is the crucial hash query, I also needs to guess which time period F will break in to be able to feed it the correct secret key.

I will work as follows. It picks b' at random from $\{1, \ldots, T(k)\}$. It then advances up to stage $b'+1$, getting and storing $q_s(k)$ transcripts in each stage. Here it breaks in to obtain $sk_{b'+1}$. That is, it is now in its imp stage with $sk_{b'+1}$ as input. It picks fp at random from $\{1, \ldots, q_h(k)+1\}$ and initializes a counter hc to 0. Only now does it start running F.

When F makes a hash query x, impersonator I returns $\mathsf{HT}[x]$ if the value is defined. If not, it increments hc by one and sets $\mathsf{QT}[hc] \leftarrow x$. If $hc \neq fp$, it picks $\mathsf{HT}[x]$ at random from $\{0,1\}^{c(k)}$ and returns it to F. If $hc = fp$, it parses x as $c \| R \| \textsc{Cmt}^* \| M$. I now sends $\textsc{Cmt}^*$ to the verifier, receiving back a challenge $\textsc{Ch}^*$. (That is, it outputs $\textsc{Cmt}, b'$ and then receives $\textsc{Ch}^*$.) It sets $\mathsf{HT}[fp] \leftarrow \textsc{Ch}^*$ and returns $\mathsf{HT}[fp]$ to F as the response to the oracle query.

When F makes a sign query M, the impersonator I does the following. For phases $1, \ldots, b'$, it answers using its stored transcripts, appropriately programming the random oracle as in the proof of Lemma 3.5. (This might lead to overwriting an existing hash value but again the analysis will show this is unlikely.) For phases $j \geq b' + 1$, it uses sk_j, which it can obtain from $sk_{b'+1}$.

Eventually, F enters the breakin phase in some period j. If $j \geq b' + 1$ then I can provide F with sk_j. If not, it aborts. Now F outputs a forgery $(M, \langle R \| \textsc{Cmt} \| \textsc{Rsp}, b \rangle)$ for some period $b < j$. (In this phase, it may continue to make hash queries, which continue to be answered as above.) We assume the hash query $b \| R \| \textsc{Cmt} \| M$ was made prior to the forgery and let i be such that $\mathsf{QT}[i] = b \| R \| \textsc{Cmt} \| M$.

If $b \neq b'$, then I aborts. Otherwise, it sends $\textsc{Rsp}$ to the verifier as the final move.

The analysis in our case is similar to that of the proof of Lemma 3.5. The only difference is that now we also have to take into account I's guess for F's break-in time period. Note that $j \geq b + 1$, so if $b = b'$, then it must be that $j \geq b' + 1$, meaning if $b = b'$, then neither of the two possible aborts occur. But the chance that $b = b'$ is at least $1/(T(k) + 1)$. The chance of guessing correctly the crucial hash query , meaning that $fp = i$, is still $1/(1+q_h(k))$. Since there are now $q_s(k)$ sign queries per time period rather than in total, the chance of mis-programming the random oracle is at most

$$\frac{[1+q_h(k) + q_s(k)T(k)] \cdot q_s(k)T(k)}{2^{q_s(k)+\beta(k)}} .$$

Hence,

$$\begin{aligned} &\mathbf{Adv}^{\text{fs-imp-pa}}_{\mathcal{FID},I}(k) \\ &\geq \quad \frac{1}{(T(k)+1)(1+q_h(k))} \cdot \left(\mathbf{Adv}^{\text{fs-uf-cma}}_{\mathcal{FSDS},F}(k) - \frac{[1+q_h(k)+q_s(k)T(k)] \cdot q_s(k)T(k)}{2^{s(k)+\beta(k)}} \right) . \end{aligned}$$

The lemma follows directly by transposing terms. ▮

As a side note, Bellare and Miner proved in [5] that the deterministic transformation of key-evolving ID schemes into key-evolving signature schemes preserves forward security in their particular case. Their proof relies on the fact that the given ID scheme is honest-verifier zero-knowledge and that the commitment is chosen at random from a large enough space. While the former is needed in order to allow a successful simulation of the signing oracle, the latter is required to avoid a high probability of collision between the simulations of the signing and hashing oracles. In our case, both requirements are no longer necessary.

The following says that if the key-evolving signature scheme is forward-secure in the random oracle model then so is the key-evolving identification scheme.

Lemma 5.5 Let $\mathcal{FID} = (K, P, \mathit{Vid}, c, T)$ be a canonical key-evolving identification scheme, let $s(\cdot)$ be a seed length, and let $\mathcal{FSDS} = (K, S, \mathit{VSig}, c, T)$ be the associated key-evolving signature scheme as per the new construction described above. Let I be an adversary attacking $\mathcal{FID}$, having time-complexity $t(\cdot)$ and making $q(\cdot)$ transcript-oracle queries across all time periods. Then, in the random oracle model, there exists a forger F attacking $\mathcal{FSDS}$ such that

$$\mathbf{Adv}^{\text{fs-imp-pa}}_{\mathcal{FID},I}(k) \leq \mathbf{Adv}^{\text{fs-uf-cma}}_{\mathcal{FSDS},F}(k) \; .$$

Furthermore, F has time-complexity $t(\cdot)$, makes at most $q(\cdot)$ sign-oracle queries and at most $q(\cdot)+1$ hash-oracle queries across all time periods. ▮

Proof of Lemma 5.5: The proof of this lemma is similar to the proof of Lemma 3.8. F initializes M to 0 and then runs I. When I requests a transcript in a time period i, forger F increments M and requests a signature of M in time period i. Obtaining $\langle R \| \text{CMT} \| \text{RSP}, i \rangle$ in response, it lets $\text{CH} \leftarrow H(i \| R \| \text{CMT} \| M)$ and returns the transcript $\text{CMT} \| \text{CH} \| \text{RSP}$ to I. When I breaks in to get the secret key sk_j of stage j, forger F breaks in too, obtains the key, and returns it to I. Now I will provide CMT and $b < j$ where CMT in the first move in a protocol with the verifier. F chooses R at random, increments M, lets $\text{CH} \leftarrow H(b \| R \| \text{CMT} \| M)$, and returns CH to I as the verifier challenge. When I returns a response RSP, forger F outputs forgery $(M, \langle R \| \text{CMT} \| \text{RSP}, b \rangle)$. ▮

Theorem 5.3 follows easily from Lemma 5.4 and Lemma 5.5. In both lemmas, the adversaries run in poly(k)-time, and it is evident from the bound of the advantages that the if and only if relationship in the theorem follows.

As in the case of standard signature and ID schemes, if we consider key-evolving ID schemes in which the commitment is chosen from a large space (i.e., $\beta(\cdot) = \omega(\log(\cdot))$), then the key-evolving signature scheme resulting from the Fiat-Shamir transform (i.e., $s(k) = 0$) is forward-secure against chosen-message attack in the random oracle model *if and only if* the underlying identification scheme is forward-secure against impersonation under passive attacks.

6 The Non-Triviality Condition

We show that applying the FS transform to a trivial identification scheme can result in an insecure signature scheme, which supports our claim in the Introduction that non-triviality of the ID scheme is necessary for security of the signature scheme obtained via the FS transform. This is implied by the following, whose proof is presented below.

Proposition 6.1 If factoring Williams integers is hard, then there exists a trivial, canonical identification scheme that is secure against impersonation under passive attacks, but the signature scheme resulting from applying the *standard* Fiat-Shamir transform is insecure. ▮

This example also shows why the generalized FS transform that we have introduced is useful. Since the ID scheme is secure against impersonation under passive attacks, the generalized transform does yield a secure signature scheme, even though the triviality of the ID scheme prevented the FS transform from doing so.

Our approach to the proof is as follows. We specify a canonical identification scheme that is trivial. First, we prove that it is indeed secure against impersonation under passive attacks. Then, we prove that the signature scheme obtained by applying the standard FS transform to it is insecure against chosen-message attacks.

Before moving on to the proof, we provide here some number theory basics and introduce relevant notation. Suppose $N = pq$, where p and q are two distinct odd primes, is a Williams integer (i.e. $p \equiv 3 \pmod 8$ and $q \equiv 7 \pmod 8$), then the following holds: for any $x \in \mathsf{Z}_N^*$, exactly one of $x, -x, 2x, -2x$ is a quadratic residue modulo N. We denote this unique square

$$
\begin{array}{rcl}
\underline{\text{Prover}} & & \underline{\text{Verifier}} \\
\text{Input: } N, p, q & & \text{Input: } N \\
& \xrightarrow{\epsilon} & \\
& & \textsc{Ch} \xleftarrow{\$} \mathsf{Z}_N^* \\
& \xleftarrow{\textsc{Ch}} & \\
y \leftarrow \mathsf{SQ}_N(\textsc{Ch}) & & \\
\textsc{Rsp} \xleftarrow{\$} \mathsf{SQR}_N(y) & & \\
& \xrightarrow{\textsc{Rsp}} & \\
& & \text{If } \textsc{Rsp}^2 \in \{\textsc{Ch}, -\textsc{Ch}, 2\textsc{Ch}, -2\textsc{Ch}\}, \\
& & \text{Then accept Else reject}
\end{array}
$$

Figure 6: A canonical identification scheme $\mathcal{ID}_{nc}$. The commitment space is of size 1.

out of the set $\{x, -x, 2x, -2x\}$ by $\mathsf{SQ}_N(x)$. Also, if y is a square modulo N, we denote the set of all four square roots of y by $\mathsf{SQR}_N(y)$. It is well-known that, given the prime factors of N, the task of computing $\mathsf{SQ}_N(\cdot)$ and $\mathsf{SQR}_N(\cdot)$ can be performed in time polynomial in the length of the inputs [27].

Now, we describe the identification scheme $\mathcal{ID}_{nc} = (K, P, V, c)$ illustrated in Figure 6. The key generation algorithm K is a usual one: it returns a secret key $sk = (p, q)$ and a public key $pk = N = pq$ where N is a k-bit Williams integer, and k is a security parameter. The secret key is given to the prover whereas the public key is published. In this scheme, we set the length of a challenge string to k. During the commitment phase, the prover sends an empty string to the verifier. In return, the verifier sends a value randomly chosen from Z_N^* to the prover as a challenge $\textsc{Ch}$. The prover's task is to multiply $\textsc{Ch}$ with 1,-1,2 and -2 modulo N, see which multiplication yields a quadratic residue w, and randomly choose and return one of the four corresponding square roots of w as a response $\textsc{Rsp}$. The verifier accepts $\textsc{Rsp}$ as valid only if its square is equal to any of the values $\textsc{Ch}, -\textsc{Ch}, 2\textsc{Ch}$, and $-2\textsc{Ch}$. Note that we allow the challenge to be chosen from Z_N^*, as opposed to $\{0,1\}^k$, for simplicity. Strictly speaking, the scheme is then not canonical as per our definition in Section 2. However, it can be easily made so, for example, by choosing random values from $\{0,1\}^k$ many times to increase the probability that at least one of the values is in Z_N^*.

We claim that the scheme $\mathcal{ID}_{nc}$ is secure against passive attacks based on the assumption that factoring is hard. Specifically, given a successful impersonator, one can construct an adversary that can factor the modulus N used in the identification scheme. But before discussing security analysis of the scheme, we define precisely what it means for the factoring problem to be hard.

Definition 6.2 [Hardness of Factoring] Let K be the key generation algorithm described previously. Let $\mathit{Fct}(\cdot)$ be an algorithm. Consider the following experiment.

Experiment $\mathbf{Exp}^{\mathrm{fac}}_{\mathit{Fct}}(k)$
  $(N, (p, q)) \xleftarrow{\$} K(k)$
  $(p', q') \xleftarrow{\$} \mathit{Fct}(N)$
  If $p'q' = N$ and $p' \neq 1$ and $q' \neq 1$ Then return 1 Else return 0

We define the advantage of Fct via

$$\mathbf{Adv}^{\mathrm{fac-w}}_{\mathit{Fct}}(k) = \Pr[\,\mathbf{Exp}^{\mathrm{fac}}_{\mathit{Fct}}(k) = 1\,]\;.$$

Algorithm $Fct(N)$

$\alpha \leftarrow 2^{-1} \bmod N$ $\quad \triangleright$ *α is the inverse of 2 in the group Z_N^**

run $I(N)$ answering to its transcript queries as follows:

when I asks for a transcript,

$v \xleftarrow{\$} \mathsf{Z}_N^*$ $\quad \triangleright$ *Pick a response*

$w \xleftarrow{\$} \{v^2, -v^2, \alpha v^2, -\alpha v^2\}$ $\quad \triangleright$ *Then compute a corresponding challenge*

return $\epsilon \| w \| v$ to I

until I outputs $st \| \epsilon$ $\quad \triangleright$ *Phase 1*

$x \xleftarrow{\$} \mathsf{Z}_N^*$; $y \leftarrow x^2 \bmod N$

$\text{CH} \xleftarrow{\$} \{y, -y, \alpha y, -\alpha y\}$

give (st, CH) to I $\quad \triangleright$ *Phase 2*

get RSP from I $\quad \triangleright$ *Phase 3*

If $\text{RSP}^2 = y$ and $\text{RSP} \not\equiv \pm x \bmod N$ $\quad \triangleright$ *Check if* RSP *is a non-trivial square root of* y

Then $p \leftarrow \gcd(\text{RSP} - x, N)$; $q \leftarrow \frac{N}{p}$ Else abort

return p, q $\quad \triangleright$ *Successfully factor if the response is non-trivial*

Figure 7: The factoring algorithm *Fct* for the proof of Claim 6.3.

The factoring problem is said to be *hard* if the function $\mathbf{Adv}^{\text{fac-w}}_{Fct}(\cdot)$ is negligible for any adversary *Fct* whose time-complexity is polynomial in the security parameter k. ▌

The following claim states explicitly the security of the identification scheme in relation to hardness of factoring.

Claim 6.3 Let $\mathcal{ID}_{nc} = (K, P, V, k)$ be the identification scheme described above. Then, if factoring is hard, the scheme $\mathcal{ID}_{nc}$ is secure against passive attacks. Concretely, for any impersonator I with time-complexity polynomial in k, there exists an adversary *Fct* that can factor N so that

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{ID}_{nc},I}(k) \;\le\; 2 \cdot \mathbf{Adv}^{\text{fac-w}}_{Fct}(k)$$

and *Fct* has time-complexity polynomial in k. ▌

Proof of Claim 6.3: The goal of an adversary *Fct* is to factor a modulus N into two distinct odd primes p and q using impersonator I. The adversary *Fct* runs I answering to its queries by simulating the transcript oracle $\mathsf{Tr}^{\mathcal{ID}_{nc}}_{N,(p,q),k}$ where $pq = N$. Then, *Fct* picks a value, squares it, and gets I to give it a square root of the square. With luck, this square root will be "non-trivial", i.e. it is not simply a negation of the square root already known to *Fct*. Once it obtains two non-trivial square roots of a single value modulo N, *Fct* can easily factor N. The details are in Figure 7.

The algorithm *Fct* runs I in the same environment as that of the experiment $\mathbf{Exp}^{\text{imp-pa}}_{\mathcal{ID}_{nc},I}(k)$. In particular, the challenge in phase 2 is a random element of Z_N^*. Furthermore, the transcripts that *Fct* generates are correct and form the same distribution as that of the transcripts generated by actual runs of $\mathcal{ID}_{nc}$. First, they are correct because if the challenge $\text{CH} = w$ is randomly chosen from $\{v^2, -v^2, \alpha v^2, -\alpha v^2\}$ where α is the inverse of 2 in the group Z_N^*, then the response $\text{RSP} = v^2$ is either $w, -w, 2w$, or $-2w$. Thus, the verifier will always accept. Second, the challenges are random elements from Z_N^*, and thus, the distribution of the transcripts is correct.

Algorithm $F^{S^H_{sk}(\cdot),H(\cdot)}(pk; R_F)$

$M \leftarrow 0$

$\sigma_1 \xleftarrow{\$} S^H_{sk}(M)$; $\sigma_2 \xleftarrow{\$} S^H_{sk}(M)$

If $\sigma_1 \equiv \pm\sigma_2 \bmod N$ Then abort

$p \leftarrow \gcd(\sigma_1 - \sigma_2, N)$; $q \leftarrow \frac{N}{p}$

$M' \leftarrow 1$

$v \leftarrow \mathsf{SQ}_N(H(M'))$

$\sigma \xleftarrow{\$} \mathsf{SQR}_N(v) \bmod N$

return (M', σ)

Figure 8: The forger for the proof of Claim 6.4

The adversary *Fct* is successful in factoring as I is successful in its impersonating the prover provided that *Fct* completes the execution without aborting. This occurs with the probability of $\frac{1}{2}$ of the success probability of I. Thus, the probability of success of *Fct* is at least half of that of I. Furthermore, the running time of *Fct* is clearly polynomial in the security parameter k plus the running of I which is also polynomial in k. Thus, Claim 6.3 is justified. ▮

Now, we show that the signature scheme obtained from applying the standard FS transform to $\mathcal{ID}_{nc}$ is completely *insecure* as stated in the following claim.

Claim 6.4 Let $\mathcal{DS}$ be the signature scheme obtained via the standard Fiat-Shamir transformation from the identification scheme $\mathcal{ID}_{nc}$ described above. Then, $\mathcal{DS}$ is *not* a secure signature scheme. Specifically, there exists a forger F that runs in time polynomial in the security parameter k such that $\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{DS},F}(k) = \frac{1}{2}$. ▮

Proof of Claim 6.4: A forger F simply queries the signing oracle on a single message M twice. With probability $\frac{1}{2}$, the returned signatures σ_1 and σ_2 will be non-trivial square roots of the same square, namely $H(M)$. Using these two signatures, the forger can factor N, and then forge a signature of any message of its choice. The details are in Figure 8. Note that the forger F does not make use of the random oracle in any special way other than using it as a given oracle.

On input (M', σ), the verification algorithm computes σ^2 and checks if it is in the set $\{H(M'), -H(M'), 2H(M'), -2H(M')\}$. Since σ is a square root of the unique square in this set, the verification algorithm accepts this forgery as valid. It is well-known that, given the prime factors p and q of N, one can compute both the element $\mathsf{SQ}_N(H(M'))$ and the set $\mathsf{SQR}_N(v)$ in time polynomial in the security parameter k. ▮

Thus, Claim 6.3 proves that the modified trivial, canonical identification scheme remains secure while Claim 6.4 proves that the corresponding signature scheme per the standard FS transform is insecure, and the proof for Proposition 6.1 is complete.

References

[1] Michel Abdalla and Leonid Reyzin. A new forward-secure digital signature scheme. In Tatsuaki Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of

Lecture Notes in Computer Science, pages 116–129, Kyoto, Japan, December 3–7, 2000. Springer-Verlag, Berlin, Germany.

[2] Ross Anderson. Two remarks on public-key cryptology. Manuscript. Relevant material presented by the author in an invited lecture at the 4th ACM Conference on Computer and Communications Security, CCS 1997, Zurich, Switzerland, April 1–4, 1997, September 2000.

[3] Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 171–188, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany.

[4] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420, Santa Barbara, CA, USA, August 16–20, 1992. Springer-Verlag, Berlin, Germany.

[5] Mihir Bellare and Sara K. Miner. A forward-secure digital signature scheme. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 431–448, Santa Barbara, CA, USA, August 15–19, 1999. Springer-Verlag, Berlin, Germany.

[6] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 268–286, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany.

[7] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Unrestricted aggregate signatures. In Christian Cachin, editor, *34rd International Colloquium on Automata, Languages and Programming – ICALP 2007*, Lecture Notes in Computer Science, Wroclaw, Poland, July 9–13, 2007. Springer-Verlag, Berlin, Germany.

[8] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press.

[9] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press.

[10] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer-Verlag, Berlin, Germany.

[11] Thomas Beth. Efficient zero-knowledged identification scheme for smart cards. In C. G. Günther, editor, *Advances in Cryptology – EUROCRYPT'88*, volume 330 of *Lecture Notes in Computer Science*, pages 77–86, Davos, Switzerland, May 25–27, 1988. Springer-Verlag, Berlin, Germany.

[12] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459, Santa Barbara, CA, USA, August 15–19, 2004. Springer-Verlag, Berlin, Germany.

[13] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, Santa Barbara, CA, USA, August 15–19, 2004. Springer-Verlag, Berlin, Germany.

[14] Ernest F. Brickell and Kevin McCurley. An interactive identification scheme based on discrete logarithms and factoring. In Ivan Damgård, editor, *Advances in Cryptology – EUROCRYPT'90*, volume 473 of *Lecture Notes in Computer Science*, pages 63–71, Aarhus, Denmark, May 21–24, 1990. Springer-Verlag, Berlin, Germany.

[15] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469, Santa Barbara, CA, USA, August 17–21, 1997. Springer-Verlag, Berlin, Germany.

[16] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998. ACM Press.

[17] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. In *26th Annual Symposium on Foundations of Computer Science*, pages 429–442, Portland, Oregon, October 21–23, 1985. IEEE Computer Society Press.

[18] Ronald Cramer and Ivan Damgård. Secure signature schemes based on interactive protocols. In Don Coppersmith, editor, *Advances in Cryptology – CRYPTO'95*, volume 963 of *Lecture Notes in Computer Science*, pages 297–310, Santa Barbara, CA, USA, August 27–31, 1995. Springer-Verlag, Berlin, Germany.

[19] Uriel Feige, Amos Fiat, and Adi Shamir. Zero knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[20] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer-Verlag, Berlin, Germany.

[21] Marc Girault. An identity-based identification scheme based on discrete logarithms modulo a composite number. In Ivan Damgård, editor, *Advances in Cryptology – EUROCRYPT'90*, volume 473 of *Lecture Notes in Computer Science*, pages 481–486, Aarhus, Denmark, May 21–24, 1990. Springer-Verlag, Berlin, Germany.

[22] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th Annual Symposium on Foundations of Computer Science*, pages 102–115, Cambridge, Massachusetts, USA, October 11–14, 2003. IEEE Computer Society Press.

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

[24] Louis C. Guillou and Jean-Jacques Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231, Santa Barbara, CA, USA, August 21–25, 1990. Springer-Verlag, Berlin, Germany.

[25] Gene Itkis and Leonid Reyzin. Forward-secure signatures with optimal signing and verifying. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 332–354, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany.

[26] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39, Cambridge, MA, USA, February 19–21, 2004. Springer-Verlag, Berlin, Germany.

[27] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* The CRC Press series on discrete mathematics and its applications. CRC Press, 2000 N.W. Corporate Blvd., Boca Raton, FL 33431-9868, USA, 1997.

[28] Silvio Micali and Leonid Reyzin. Improving the exact security of digital signature schemes. In Rainer Baumgart, editor, *International Exhibition and Congress on Network Security – CQRE'99*, volume 1740 of *Lecture Notes in Computer Science*, pages 167–182, Dsseldorf, Germany, November 30 – December 2, 1999. Springer-Verlag, Berlin, Germany.

[29] Silvio Micali and Adi Shamir. An improvement of the Fiat-Shamir identification and signature scheme. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 244–248, Santa Barbara, CA, USA, August 21–25, 1990. Springer-Verlag, Berlin, Germany.

[30] Kazuo Ohta and Tatsuaki Okamoto. On concrete security treatment of signatures derived from identification. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 354–369, Santa Barbara, CA, USA, August 23–27, 1998. Springer-Verlag, Berlin, Germany.

[31] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53, Santa Barbara, CA, USA, August 16–20, 1992. Springer-Verlag, Berlin, Germany.

[32] H. Ong and Claus-Peter Schnorr. Fast signature generation with a Fiat Shamir–like scheme. In Ivan Damgård, editor, *Advances in Cryptology – EUROCRYPT'90*, volume 473 of *Lecture Notes in Computer Science*, pages 432–440, Aarhus, Denmark, May 21–24, 1990. Springer-Verlag, Berlin, Germany.

[33] David Pointcheval. A new identification scheme based on the perceptrons problem. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology – EUROCRYPT'95*, volume 921 of *Lecture Notes in Computer Science*, pages 319–328, Saint-Malo, France, May 21–25, 1995. Springer-Verlag, Berlin, Germany.

[34] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398, Saragossa, Spain, May 12–16, 1996. Springer-Verlag, Berlin, Germany.

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

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

[37] Victor Shoup. On the security of a practical identification scheme. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 344–353, Saragossa, Spain, May 12–16, 1996. Springer-Verlag, Berlin, Germany.

[38] Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21, Santa Barbara, CA, USA, August 22–26, 1994. Springer-Verlag, Berlin, Germany.