Two-Tier Signatures, Strongly Unforgeable Signatures, and Fiat-Shamir Without Random Oracles

Mihir Bellare and Sarah Shoup

Department of Computer Science and Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0404
mihir@cs.ucsd.edu, sshoup@cs.ucsd.edu
www-cse.ucsd.edu/users/mihir, www-cse.ucsd.edu/users/sshoup

Abstract. We provide a positive result about the Fiat-Shamir (FS) transform in the standard model, showing how to use it to convert three-move identification protocols into *two-tier signature schemes* with a proof of security that makes a standard assumption on the hash function rather than modeling it as a random oracle. The result requires security of the starting protocol against *concurrent* attacks. We can show that numerous protocols have the required properties and so obtain numerous efficient two-tier schemes. Our first application is a two-tier scheme based transform of any unforgeable signature scheme into a strongly unforgeable one. (This extends Boneh, Shen and Waters [8] whose transform only applies to a limited class of schemes.) The second application is new one-time signature schemes that, compared to one-way function based ones of the same computational cost, have smaller key and signature sizes.

1 Introduction

Recall that the Fiat-Shamir (FS) transform [18] is a way to obtain a signature scheme from a three-move identification protocol by "collapsing" the interaction via a hash function. (Briefly, the signature consists of the two prover moves corresponding to a verifier challenge set to be the hash of the first prover move and the message being signed.) There are lots of protocols to which the transform can be applied, and the resulting signature schemes include some of the most efficient known (eg. [35,24,23]). Furthermore, due to their algebraic properties, FS-transform-derived signature schemes lend themselves nicely to extensions such as to blind [33], multi [28] or group [7] signatures to name just a few. For these reasons, the transform is popular and widely used.

Naturally, one would like that the constructed signature scheme meets the standard notion of unforgeability under chosen-message attack (uf-cma) of [22]. Results of [33,30,1] say this is true in the random oracle (RO) model (meaning, if the hash function is a random oracle) as long as the starting protocol is itself secure (we will discuss in what sense later). However, Goldwasser and Tauman-Kalai [21] show the existence of a protocol that, under the FS transform, yields a

T. Okamoto and X. Wang (Eds.): PKC 2007, LNCS 4450, pp. 201–216, 2007.
© International Association for Cryptologic Research 2007

signature scheme that is uf-cma secure when the hash function is a RO but is not uf-cma secure for *any* "real" implementation of the hash function. This means that the transform (at least in general) does not yield uf-cma secure schemes in the standard model.

The question we ask is whether the FS transform can, however, yield weaker-than-uf-cma but still useful types of signature schemes in the standard model. We answer this in the affirmative. We show how the FS transform yields *two-tier signature schemes* which are secure assuming only that the hash function is collision-resistant and the starting protocol is secure. We exhibit some applications of two-tier signatures in general and FS-derived ones in particular, namely for an efficient and general transform of uf-cma to strongly unforgeable (suf-cma) signature schemes and to implement one-time signatures that are much shorter than conventional ones of the same computational cost. Let us now look at all this in more detail.

Two-tier schemes. In a two-tier scheme, a signer has a primary public key and matching primary secret key. Each time it wants to sign, it generates a fresh pair of secondary public and secret keys and produces the signature as a function of these, the primary keys and the message. Verification requires not only the primary public key but also the secondary one associated to the message. Security requires that it be computationally infeasible to forge relative to the primary public key and any secondary public key that was generated by the signer, even under a chosen-message attack.

As the reader might rightfully note, two-tier signatures are not well suited for direct signing in the standard PKI, because not just the primary but also the secondary public keys would need to be certified. However, we do not propose to use them in this direct way. Instead what we will see is that they are useful tools in building other primitives.

Building two-tier signatures via FS. We adapt the FS transform in a natural way to convert a three-move identification protocol into a two-tier signature scheme. (Briefly, the first prover move, rather than being in the signature, is now the secondary public key. See Section 4 for details.) We show (cf. Theorem 2) that the constructed two-tier scheme is secure assuming the protocol is secure (we will see exactly what this means below) and the hash function is collision-resistant. So security of FS-based two-tier signatures is guaranteed in the standard model unlike security of FS-based regular signatures which is guaranteed only in the RO model.

Both the security of regular FS-based signatures (in the RO model) [30,1] and the security of our FS-based two-tier signatures (in the standard model) are based on some security assumption about the starting protocol. (Naturally, since otherwise there is no reason for the constructs to be secure.) There is, however, a difference in the two cases. Recall that security of this class of protocols can be considered under three different types of attack: passive, where the adversary merely observes interactions between the prover and honest verifier; active [18,16], where the adversary plays a cheating verifier and engages in sequential interactions with the honest prover; or concurrent [4], where, as a cheating

verifier, the adversary can interact concurrently with different prover clones. For (uf-cma) security of FS-based regular signatures in the RO model, it suffices that the protocol be secure against passive (i.e. eavesdropping) attack [30,1]. Our result showing security of FS-based two-tier signatures requires however that the protocol be secure against *concurrent* attack. Thus, part of what makes it possible to dispense with random oracles is to start from protocols with a stronger property. However, we show that the property is in fact possessed by the bulk of example protocols, so that we lose very little in terms of actual constructions. Specifically it is easy to show appropriate security under concurrent attack for the Schnorr [35], Okamoto [31], and GQ [24,23] protocols as well as others, using techniques from the original papers and more recent analyses [4,2]. Thereby we obtain numerous specific and efficient constructions of two-tier signatures via the FS transform.

We think this is an interesting application of concurrent security of protocols. The latter is usually motivated as being important for certain communication environments such as the Internet, while we are saying it is relevant to the security of a protocol-based signature.

From uf-cma to suf-cma. Returning again to regular (rather than two-tier) signatures, recall that strong unforgeability (suf-cma) is a stronger requirement than the usual uf-cma of [22], requiring not only that the adversary can't produce a signature of a new message but also that it can't produce a new signature of an old message (i.e. one whose signature it has already obtained via its chosen-message attack). The problem we are interested in is to convert a uf-cma scheme into a suf-cma one without using random oracles. Our work is motivated by Boneh, Shen and Waters [8] who turn Waters' uf-cma scheme [38] into an suf-cma one via a transform that applies to a subclass of signature schemes that they call *partitioned*. Unfortunately, there seem to be hardly any schemes in this class besides Waters', so their transform is of limited utility. We, instead, provide a general transform that applies to *any* uf-cma scheme. The transform uses as a tool any two-tier scheme. Instantiating the latter with a FS-based two-tier scheme we obtain efficient, standard model transforms. For example, using the Schnorr scheme, our transform confers suf-cma while adding just one exponentiation to the signing time and increasing the signature size by only two group elements. Briefly, the idea of the transform is to have two signatures, one from the original uf-cma scheme and the other from the two-tier scheme, mutually authenticate each other. This application exploits the fact that our FS-based two-tier signatures are themselves strongly unforgeable due to properties of the starting protocols. (That is, if the adversary has seen the signature of m relative to a secondary public key, it can produce neither a different signature of m nor a signature of some $m' \neq m$ relative to the same secondary key.)

New one-time signatures. A two-tier signature scheme yields a one-time signature scheme as a special case. (Restrict to a single secondary key.) Thus we obtain FS-based strongly unforgeable one-time signatures. These turn out to be interesting because they have smaller key and signature sizes than conventional one-way function based one-time schemes of the same computational

cost. Specifically, say we are signing a 160-bit message (which is the hash of the real message). Our Schnorr-instantiated FS-based one-time scheme implemented over a 160-bit elliptic curve group has key size 480 bits, signature size 160 bits, key-generation time 2 exponentiations, signing time 1 multiplication, and verifying time 1 exponentiation. Let us contrast this with what is achieved by the best one-way function based one-time signature schemes, namely those of [15,6]. Unforgeability is proved by [15] under the assumption that the one-way function is quasi-one-way. We observe that the scheme is strongly unforgeable under the additional assumption that the function is collision-resistant. So, let us use SHA-1 as the one-way function. The resulting schemes exhibit the following size to computation tradeoff. For any positive integer t dividing 160, there is a one time scheme with key and signature size $(1 + 160/t) \cdot 160$ and key-generation, signing and verifying time $(160/t) \cdot 2^t$ hash computations. An implementation with the crypto++ library [12] indicates that an exponentiation in a 160-bit group costs about $3,300$ hashes. To match the key-generation time of 2 exponentiations (which is the largest of the computation times in our algebraic scheme) we thus want to choose t such that $(160/t) \cdot 2^t \approx 6,600$. Let us (generously) set $t = 10$. The key and signature size now becomes $2,720$ bits, which is much more than in our scheme. (And at this point, while key-generation time in the one-way function scheme is essentially the same as in our scheme, signing time is much more.) Note we would get the same efficiency gains using the standard Schnorr [35] scheme instead of our scheme, but the proof of the former uses random oracles [32].

Our new one-time signature scheme is interesting for applications like the DDN and Lindell constructions of IND-CCA public-key encryption schemes [14,27], the IBE-based constructions of IND-CCA schemes of [9], and the composition of encryption schemes [13]. All of these make use of strongly unforgeable one-time signatures, and the reduced key size of the latter results in reduced ciphertext size for the encryption schemes they build.

Alternative two-tier schemes and their implications. We noted above that two-tier schemes yield (strongly unforgeable) one-time ones as a special case. Conversely, however, one can also construct a two-tier scheme from any strongly unforgeable one-time scheme. (Set the primary keys to empty, and use a new instance of the one-time scheme for each secondary key. See [5] for details.)

One implication of this observation is that we can obtain one-way function based constructions of the primitives we have been discussing, thereby answering the main foundational question about their existence. Specifically, as we discuss in more detail in [5], it is easy to build UOWHF [29] based strongly unforgeable one-time schemes. Since UOWHFs exist given any one-way function [34], we obtain one-way function based two-tier schemes. We also obtain a one-way function based transform of uf-cma signature schemes into suf-cma ones. This yields a somewhat simpler construction of a one-way function based suf-cma signature scheme than given by Goldreich in [19].

However, the above observation (that strongly unforgeable one-time schemes yield two-tier schemes) also raises some questions. The first of these is, what is

the point of FS-based two-tier schemes given that there are other ways to build two-tier schemes? However, the same question can be asked about the use of the FS transform to build regular signatures, for of course regular signatures can be built in other ways too. In both cases the point is that FS-based constructs have efficiency or other properties not provided by other constructs. (Specifically, FS-based two-tier schemes have smaller key and signature sizes than two-tier schemes of the same computational cost built from any known strongly unforgeable one-time schemes.) One might also ask what is the point of introducing two-tier schemes at all. Indeed, we could have based our transform (of uf-cma schemes into suf-cma ones) on strongly unforgeable one-time schemes rather than on two-tier schemes, and we could have built FS-based strongly unforgeable one-time schemes directly rather than building two-tier schemes. Two-tier schemes however have the advantage over using one-time schemes that any key information that is long-lived across multiple instances of a one-time scheme can be re-used, resulting in shorter keys. This results in shorter signatures for the suf-cma schemes built by our transform. Another advantage is improved concrete security: the reduction from suf-cma signatures to two-tier and uf-cma signatures is tight, whereas if we had used one-time signatures, we would incur a factor of the number of signing queries. Furthermore our reductions from identification protocols to two-tier schemes derived via the FS transform are tight too. Overall it seemed simple and worthwhile enough to make the optimization (meaning to introduce and use two-tier signatures) and hence we have done so.

Related work. Cramer and Damgård [11] present a non-RO transform of protocols with certain properties into signature schemes. Their transform is not the FS one (it is more complex and less efficient) but they obtain regular unforgeable signature schemes while we obtain only two-tier schemes.

Independently of our work, others have extended [8] to provide general transforms of unforgeable signature schemes into strongly unforgeable ones. The transform of Huang, Wong and Zhao [26] is similar to the special case of ours with a two-tier signature scheme built from a collision-resistant hash function based one-time signature scheme. However, this yields large signatures. Teranishi, Oyama and Ogata [37] present a discrete log, chameleon commitment based transform that is very efficient.

2 Definitions

Notation and conventions. We denote by $a_1 \| \cdots \| a_n$ a string encoding of $a_1, \ldots, a_n$ from which the constituent objects are uniquely recoverable. We denote the empty string by ε. Unless otherwise indicated, an algorithm may be randomized. A collision for a function h is a pair x, y of distinct points in its domain such that $h(x) = h(y)$. If A is a randomized algorithm then $y \xleftarrow{\$} A(x_1, \ldots)$ denotes the operation of running A with fresh coins on inputs $x_1, \ldots$ and letting y denote the output. If S is a (finite) set then $s \xleftarrow{\$} S$ denotes the operation of picking s uniformly at random from S.

SIGNATURES. A (digital) signature scheme $\mathsf{DS} = (\mathsf{KG}, \mathsf{SGN}, \mathsf{VF})$ is specified as usual by three algorithms. Via $(PK, SK) \xleftarrow{\$} \mathsf{KG}$ a prospective signer can generate its public and associated secret key. Via $\sigma \xleftarrow{\$} \mathsf{SGN}(SK, M)$ the signer can produce a signature σ on a message $M \in \{0,1\}^*$. Via $d \leftarrow \mathsf{VF}(PK, M, \sigma)$, a verifier can run the deterministic verification algorithm to get a decision bit $d \in \{0,1\}$. We require perfect consistency, meaning that

$$\Pr\left[\, \mathsf{VF}(PK, M, \sigma) = 1 \;:\; (PK, SK) \xleftarrow{\$} \mathsf{KG}\,;\; \sigma \xleftarrow{\$} \mathsf{SGN}(SK, M) \,\right] \;=\; 1$$

for all messages M. To define security consider the following game involving an adversary A:

$$(PK, SK) \xleftarrow{\$} \mathsf{KG}\,;\; (M, \sigma) \xleftarrow{\$} A^{\mathsf{SGN}(SK,\cdot)}(PK)\;.$$

The adversary is given a signing oracle and the public key, and outputs a message and candidate signature. Let $M_1, \ldots, M_q$ denote the messages queried by A to its oracle in its chosen-message attack, and let $\sigma_1, \ldots, \sigma_q$ denote the signatures returned by the oracle, respectively. We say that A forges if $\mathsf{VF}(PK, M, \sigma) = 1$ and $M \notin \{M_1, \ldots, M_q\}$. We say that A strongly forges if $\mathsf{VF}(PK, M, \sigma) = 1$ and $(M, \sigma) \notin \{(M_1, \sigma_1), \ldots, (M_q, \sigma_q)\}$. We let $\mathbf{Adv}^{\text{uf-cma}}_{\mathsf{DS}}(A)$ and $\mathbf{Adv}^{\text{suf-cma}}_{\mathsf{DS}}(A)$ denote, respectively, the probability that A forges and the probability that it strongly forges. The first measure represents the standard uf-cma notion of [22], while the second represents strong unforgeability (suf-cma).

SYNTAX OF TWO-TIER SIGNATURE SCHEMES. A two-tier signature scheme $\mathsf{ds} = (\mathsf{pkg}, \mathsf{skg}, \mathsf{sgn}, \mathsf{vf})$ is specified by four algorithms. They are called the primary key-generation, secondary key-generation, signing and verifying algorithms, respectively, and the last is deterministic. Via $(ppk, psk) \xleftarrow{\$} \mathsf{pkg}$, a prospective signer generates a primary public key ppk and associated primary secret key psk. Think of these as the keys at the first tier of the two-tier scheme. The signer may then at any time generate a secondary public key spk and associated secondary secret key ssk via $(spk, ssk) \xleftarrow{\$} \mathsf{skg}(ppk, psk)$. These will be the second tier keys, and there can be many of them. Via $s \xleftarrow{\$} \mathsf{sgn}(psk, ssk, m)$ the signer can generate a signature of a message m. Via $d \leftarrow \mathsf{vf}(ppk, spk, m, s)$, a verifier can produce a decision bit $d \in \{0,1\}$ indicating whether or not s is a valid signature of m relative to ppk, spk. We require perfect consistency, meaning that for all messages m, $\mathsf{vf}(ppk, spk, m, s) = 1$ with probability 1 in the following experiment:

$$(ppk, psk) \xleftarrow{\$} \mathsf{pkg}\,;\; (spk, ssk) \xleftarrow{\$} \mathsf{skg}(ppk, psk)\,;\; s \xleftarrow{\$} \mathsf{sgn}(psk, ssk, m)\;.$$

In usage, a signer will have a single primary key pair. It will, however, use a fresh secondary key pair for each message, meaning the secondary key pairs are one-time. Since generation of a secondary key pair does not require knowing the message, this generation can either be done when the message to be signed arrives, or off-line, in advance.

SECURITY OF TWO-TIER SIGNATURE SCHEMES. To define security, consider the following game. We let $(ppk, psk) \xleftarrow{\$} \mathsf{pkg}$, initialize a set U to $\emptyset$ and initialize

Oracle $\text{SPKO}()$	Oracle $\text{SIGNO}(j, m)$
$i \leftarrow i + 1$	If $j > i$ OR $j \in U$ then return $\perp$
$(spk_i, ssk_i) \xleftarrow{\$} \mathsf{skg}(ppk, psk)$	$U \leftarrow U \cup \{j\}$; $m_j \leftarrow m$
Return spk_i	$s_j \xleftarrow{\$} \mathsf{sgn}(psk, ssk_j, m_j)$
	Return s_j

Fig. 1. Oracles for adversary attacking two-tier scheme $\mathsf{ds} = (\mathsf{pkg}, \mathsf{skg}, \mathsf{sgn}, \mathsf{vf})$

a counter i to 0. We then run an adversary A on input ppk with access to the oracles shown in Figure 1. A can obtain a fresh secondary public key at any time by calling its secondary public-key oracle SPKO. A can obtain a signature of a message m of its choice under an already generated secondary public key spk_j by calling the signing oracle SIGNO on inputs j, m, where $j \geq 1$. However, A cannot obtain more than one signature under a particular secondary public key. (This restriction is enforced by the oracle via the set U.) Finally A outputs a forgery, which must be a triple of the form (l, m, s). Let $(j_1, m_1), \ldots, (j_q, m_q)$ denote the queries made by A to its SIGNO oracle in its chosen-message attack, and let $s_1, \ldots, s_q$ denote the signatures returned by the oracle, respectively. We say that A wins if $\mathsf{vf}(ppk, spk_l, m, s) = 1$ and $1 \leq l \leq i$ but $(l, m, s) \notin \{(j_1, m_1, s_1), \ldots, (j_q, m_q, s_q)\}$. Here i is the final value of the counter, meaning the number of queries A made to SPKO. The probability that A wins is denoted $\mathbf{Adv}^{\text{suf-cma}}_{\mathsf{ds}}(A)$.

Notice that this definition is of strong unforgeability, meaning this has been built in as a requirement. We do this because it is what the applications need and also what the FS-based constructs naturally provide.

DISCUSSION. Two-tier schemes are hybrids of regular and one-time schemes. If the secondary keys are empty, we have a regular scheme. If the primary keys are empty, we have multiple instances of a one-time scheme.

3 From uf-cma to suf-cma

Suppose we are given a uf-cma signature scheme DS and want to transform it into a suf-cma signature scheme $\overline{\mathsf{DS}}$, efficiently and without random oracles. This problem was recently considered by [8] who provided a transform that works under the assumption that the starting uf-cma scheme is what they call "partitioned." However, there are few examples of partitioned schemes. In this section, we provide a general transform, namely one that applies to any starting uf-cma scheme. It uses an arbitrary two-tier scheme as an auxiliary tool. The transform does not use random oracles, and, when instantiated with appropriate FS-based two-tier schemes, matches that of [8] in computational overhead while providing signatures that are longer by only one group element.

THE TRANSFORM. Let $\mathsf{DS} = (\mathsf{KG}, \mathsf{SGN}, \mathsf{VF})$ be the given uf-cma scheme. Let $\mathsf{ds} = (\mathsf{pkg}, \mathsf{skg}, \mathsf{sgn}, \mathsf{vf})$ be a (any) given two-tier scheme. We associate to these the signature scheme $\overline{\mathsf{DS}} = (\overline{\mathsf{KG}}, \overline{\mathsf{SGN}}, \overline{\mathsf{VF}})$ defined as follows. The key-generation

algorithm $\overline{\mathsf{KG}}$ runs KG to get (PK, SK), runs pkg to get (ppk, psk), and returns $\overline{PK} = PK \| ppk$ as the public key and $\overline{SK} = SK \| psk$ as the secret key. The new signing and verifying algorithms are as follows:

Algorithm $\overline{\mathsf{SGN}}(\overline{SK}, \overline{M})$
  Parse $\overline{SK}$ as $SK \| psk$
  $(spk, ssk) \xleftarrow{\$} \mathsf{skg}(ppk, psk)$
  $M \leftarrow spk \| \overline{M}$
  $S \xleftarrow{\$} \mathsf{SGN}(SK, M)$
  $s \xleftarrow{\$} \mathsf{sgn}(psk, ssk, S)$
  $\overline{S} \leftarrow S \| spk \| s$
  Return $\overline{S}$

Algorithm $\overline{\mathsf{VF}}(\overline{PK}, \overline{M}, \overline{S})$
  Parse $\overline{PK}$ as $PK \| ppk$
  Parse $\overline{S}$ as $S \| spk \| s$
  $M \leftarrow spk \| \overline{M}$
  If $\mathsf{VF}(PK, M, S) = 0$ then return 0
  If $\mathsf{vf}(ppk, spk, S, s) = 0$ then return 0
  Return 1

The following implies that the constructed scheme $\overline{\mathsf{DS}}$ is strongly unforgeable if DS is unforgeable and the two-tier scheme ds is strongly unforgeable. The proof may be found in [5].

Theorem 1. *Let $\overline{\mathsf{DS}}$ be the signature scheme associated to signature scheme DS and two-tier signature scheme ds as described above. Let $\overline{F}$ be an adversary attacking the strong unforgeability of $\overline{\mathsf{DS}}$ and making at most q signing queries. Then there exist adversaries F, f attacking the unforgeability of DS and the strong unforgeability of ds, respectively, such that*

$$\mathbf{Adv}^{\text{suf-cma}}_{\overline{\mathsf{DS}}}(\overline{F}) \;\le\; \mathbf{Adv}^{\text{uf-cma}}_{\mathsf{DS}}(F) + \mathbf{Adv}^{\text{suf-cma}}_{\mathsf{ds}}(f)\ .$$

Furthermore F and f make at most q signing queries, and their running times are that of $\overline{F}$ plus an overhead that is linear in q. ∎

4 Constructions of Two-Tier Schemes

Canonical identification protocols. The FS transform applies to a class of protocols we call canonical identification protocols [1]. We need to have a general syntax for these protocols since the transform and its proof will refer to this. The protocol can be described as a tuple $\mathsf{ID} = (\mathcal{K}, P, \text{ChSet}, V)$. Via $(pk, sk) \xleftarrow{\$} \mathcal{K}$, the (honest) prover generates its public and secret keys. Now the public key pk is viewed as an input for the verifier, while sk is a private input to the honest prover. The prover can now convince the verifier of its identity via a three move interaction as depicted in Figure 2. We refer to the moves as commitment, challenge, and response. The (honest) prover maintains a state St whose initial value is its secret key sk. In its first move, it applies P to the current conversation (which is ε) and current state ($\text{St} = sk$) to get a commitment Cm and an updated state St. The former is sent to the verifier, who now draws its challenge Ch at random from ChSet and sends this to the prover. The (honest) prover now lets Rp $\xleftarrow{\$} P(\text{Cm} \| \text{Ch}, \text{St})$ and sends Rp back to the verifier. The latter applies the deterministic function V to pk and the transcript $\text{Cm} \| \text{Ch} \| \text{Rp}$ to output the decision Dec $\in \{0, 1\}$. We require *perfect completeness*, meaning

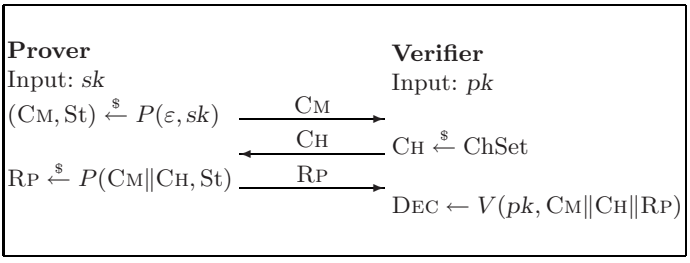

Fig. 2. Canonical Protocol. Keys pk and sk are produced using key generation algorithm $\mathcal{K}$.

that for all (pk, sk) that can be output by $\mathcal{K}$ we have $V(pk, \text{CM}\|\text{CH}\|\text{RP}) = 1$ with probability 1 in the following experiment:

$$(\text{CM}, \text{St}) \xleftarrow{\$} P(\varepsilon, sk)\ ;\ \text{CH} \xleftarrow{\$} \text{ChSet}\ ;\ \text{RP} \xleftarrow{\$} P(\text{CM}\|\text{CH}, \text{St})\ . \qquad (1)$$

Examples of canonical identification protocols include the Schnorr protocol [35] illustrated in Figure 3 and the Okamoto protocol [31] illustrated in Figure 4.

SECURITY NOTIONS. The "master" property of protocols in this domain is *special soundness*. We will consider it under different forms of attack, namely passive, active and concurrent. (We only use the last in our results but for discussions it is useful to see them all.) To define these consider the following game involving an attacker I. The game begins by picking keys via $(pk, sk) \xleftarrow{\$} \mathcal{K}$. Then there are two phases. In the first phase, adversary I gets to mount its attack on the honest prover. In a passive attack, it gets an oracle that upon being invoked (with no arguments) returns a random transcript of an interaction between the honest prover (given input sk) and the verifier (given input pk). In an active or concurrent attack, I gets to play the role of verifier and interact with "clones" of the honest prover. We can imagine a sequence P_j $(j \geq 1)$ of potential clones. Each clone maintains a state St_j and has its own random coins. The game maintains a counter a, initially 0, and a set A of clones that are *activated*, initially empty. Adversary I can ask for a new clone to be activated, in which case the game increments a, computes $(\text{CM}_a, \text{St}_a) \xleftarrow{\$} P(\varepsilon, sk)$, and returns CM_a to I. If the attack is concurrent, it adds a to A, but if the attack is active, it replaces A by $\{a\}$, meaning that only one clone can be activated at any time. If $j \in A$ then I can send clone P_j a message CH_j representing the verifier move. Adversary I can pick this value any way it wishes, in particular not necessarily at random like the honest verifier. The game computes $\text{RP}_j \xleftarrow{\$} P(\text{CM}_j\|\text{CH}_j, \text{St}_j)$, returns RP_j to I, and removes j from A. (Which means no further interaction with P_j is possible.) Note that the difference between an active and concurrent attack is that in the former, the adversary is allowed to have only one clone (namely P_a) activated at any time, corresponding to sequential interactions with the honest prover, while in a concurrent attack, any number of clones may simultaneously be activated, and I can choose a challenge sent to one of them as a function of all communications it has received from all clones so far. Note that in either case, the adversary

does not see or control the internal state of a prover clone. In no case can it reset or backup a clone. After it has completed its attack (of whatever form), we enter the second phase. The adversary outputs a pair $(\text{CM}, \text{CH}_1, \text{RP}_1), (\text{CM}, \text{CH}_2, \text{RP}_2)$ of transcripts where the commitment is the same. It wins if these transcripts are accepting but $(\text{CH}_1, \text{RP}_1) \neq (\text{CH}_2, \text{RP}_2)$. The probability that it wins is denoted $\mathbf{Adv}^{\text{ss-atk}}_{\mathsf{ID}}(I)$, where atk = pa if the attack is passive; atk = aa if the attack is active; and atk = ca if the attack is concurrent.

Discussion. The typical formulation of special soundness is that given a pair $(\text{CM}, \text{CH}_1, \text{RP}_1), (\text{CM}, \text{CH}_2, \text{RP}_2)$ of accepting transcripts where the commitment is the same but $\text{CH}_1 \neq \text{CH}_2$, one can easily find a matching secret key sk. This implies in particular that the protocol is a proof of knowledge of the secret key which in turn is crucial to proving security against impersonation under passive, active or concurrent attack. (Impersonation means that after its attack, meaning in the second phase, rather than outputting a pair of transcripts, the adversary plays the role of prover in an interaction with the honest verifier and wins if it can convice the latter to accept.) For our purposes, however, we work directly with special soundness rather than any of its derivative properties. We directly require that the probability of finding transcripts of the appropriate type is negligible rather than relating this to finding the secret key. This is similar to the security requirement used in [11], though they apply it to a different protocol-based transform. Note we weaken the condition under which the adversary wins from $\text{CH}_1 \neq \text{CH}_2$ to $(\text{CH}_1, \text{RP}_1) \neq (\text{CH}_2, \text{RP}_2)$. We will have to prove that the resulting stronger security requirement is met by the constructs.

A Σ protocol is one that has special soundness and honest-verifier zero-knowledge. We do not explicitly require the latter as part of special soundness, although in establishing special soundness of particular protocols we might use it. Note none of the example protocols in this domain are full (i.e. even against cheating verifiers) zero-knowledge. Indeed, this is ruled out under blackbox simulation [20].

Special soundness is usually considered as a stand-alone property, but it is natural to consider it under the three forms of attack that exist for identification protocols as we have done.

For our particular transform, we require special soundness under concurrent attack, rather than active or passive. This is necessary for our proof due to the nature of two-tier signatures and our security definition. In our transform, each request for a new secondary key will require the instantiation of a new clone. Each clone will be required for a signature using its corresponding key, and since the adversary is not required to sign on a key immediately after acquiring it, it is necessary to have multiple clones active at a time. For this reason, we require security under concurrent attack.

The transform. We now describe how to turn a canonical identification protocol $\mathsf{ID} = (\mathcal{K}, P, \text{ChSet}, V)$ into a two-tier signature scheme $\mathsf{ds} = (\mathsf{pkg}, \mathsf{skg}, \mathsf{sgn}, \mathsf{vf})$ via the Fiat-Shamir transform. We do not use a random oracle but instead a family $H : \{0,1\}^k \times \{0,1\}^* \to \text{ChSet}$ of collision-resistant (CR) hash functions where each k-bit key K specifies a particular hash function $H(K, \cdot)$ with range

the challenge set ChSet. (The keys will be random but public.) The primary key generation algorithm pkg lets $K \xleftarrow{\$} \{0,1\}^k$ and $(pk, sk) \xleftarrow{\$} \mathcal{K}$, and returns $(ppk, psk) \leftarrow (K\|pk, K\|sk)$. The skg, sgn, and vf algorithms are as follows:

$\mathsf{skg}(ppk, psk)$	$\mathsf{sgn}(psk, ssk, m)$	$\mathsf{vf}(ppk, spk, m, s)$
Parse ppk as $K\|pk$	Parse psk as $K\|sk$	Parse ppk as $K\|pk$
Parse psk as $K\|sk$	Parse ssk as $\textsc{Cm}\|\textsc{St}$	$\textsc{Cm} \leftarrow spk$
$(\textsc{Cm}, \textsc{St}) \xleftarrow{\$} P(\varepsilon, sk)$	$\textsc{Ch} \leftarrow H(K, \textsc{Cm}\|m)$	$\textsc{Ch} \leftarrow H(K, \textsc{Cm}\|m)$
$spk \leftarrow \textsc{Cm}$	$\textsc{Rp} \xleftarrow{\$} P(\textsc{Cm}\|\textsc{Ch}, \textsc{St})$	$\textsc{Rp} \leftarrow s$
$ssk \leftarrow \textsc{Cm}\|\textsc{St}$	$s \leftarrow \textsc{Rp}$	$\textsc{Dec} \leftarrow V(pk, \textsc{Cm}\|\textsc{Ch}\|\textsc{Rp})$
Return (spk, ssk)	Return s	Return $\textsc{Dec}$

Note that in generating s, algorithm P will be executed with a challenge that, unlike the one the honest prover expects to receive, is not random. The implications for security are dealt with by the theorem that follows, but at this point we need to first check that it does not lead to a violation of the perfect consistency requirement of two-tier schemes. This is true because the protocol has perfect completeness as per Equation (1), which means for *all* values of the verifier challenge, the prover returns a response that leads the verifier to accept.

Security of the transform. Recall that the cr-advantage of an adversary F attacking H is $\mathbf{Adv}^{\mathrm{cr}}_{H}(F)$, defined as follows:

$$\Pr\left[\, H(K, x_1) = H(K, x_2) \wedge x_1 \neq x_2 \;:\; K \xleftarrow{\$} \{0,1\}^k ;\, (x_1, x_2) \xleftarrow{\$} F(K) \,\right] .$$

The following says that if H is CR and ID is secure against concurrent attack then the two-tier scheme derived via the FS transform is secure. The proof may be found in [5].

Theorem 2. *Let* $\mathsf{ds} = (\mathsf{pkg}, \mathsf{skg}, \mathsf{sgn}, \mathsf{vf})$ *be the two-tier signature scheme associated to canonical identification protocol* $\mathsf{ID} = (\mathcal{K}, P, \mathrm{ChSet}, V)$ *and hash function* $H\colon \{0,1\}^k \times \{0,1\}^* \to \mathrm{ChSet}$ *via the Fiat-Shamir transform as described above. Let* f *be an adversary attacking the strong unforgeability of* ds *and making at most* q *signing queries. Then there exists an adversary* I *attacking the special soundness of* ID *under concurrent attack, and an adversary* F *attacking the collision-resistance of* H*, such that*

$$\mathbf{Adv}^{\text{suf-cma}}_{\mathsf{ds}}(f) \;\leq\; \mathbf{Adv}^{\text{ss}-\text{ca}}_{\mathsf{ID}}(I) + \mathbf{Adv}^{\mathrm{cr}}_{H}(F) .$$

Furthermore I *initiates at most* $q + 1$ *prover clones, and the running time of* I *and* F *is that of* f *plus a constant amount of overhead.* ∎

To instantiate the above we now seek efficient protocols for which we can prove special soundness under concurrent attack. There are actually several such protocols. We illustrate by looking at a pair of examples that are representative due to the proof techniques.

Definitions. In what follows, G denotes a group whose order p is a prime. (For example an appropriate elliptic curve group, or a subgroup of the group of integers modulo some prime q such that p divides $q - 1$.) Let $G^* = G - \{1\}$ be

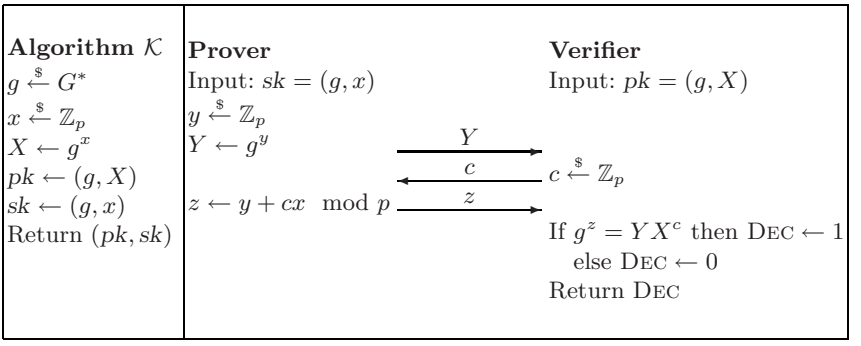

Fig. 3. Schnorr Protocol. Above, G is a group of prime order p, and $\mathrm{ChSet} = \mathbb{Z}_p$.

the set of generators of G, where 1 is the identity element of G. We let $\mathrm{DLog}_g(h)$ denote the discrete logarithm of $h \in G$ to base a generator $g \in G^*$. We assume G, p are fixed and known to all parties. Let

$$\mathbf{Adv}^{\mathrm{dl}}_G(A) \;=\; \Pr\left[\, x' = x \;\;:\;\; g \xleftarrow{\$} G^* \,;\, x \xleftarrow{\$} \mathbb{Z}_p \,;\, x' \xleftarrow{\$} A(g, g^x) \,\right]$$

denote the advantage of an adversary A in attacking the discrete logarithm (dl) problem. An adversary A for the one more dl (omdl) problem [3] is given input a generator $g \in G^*$ and has access to two oracles. The first is a challenge oracle $\textsc{ch}\mathrm{O}()$ that takes no inputs and, every time it is invoked, returns a random element of G. The second is a dl oracle $\mathrm{DLog}_g(\cdot)$ that, given any $W \in G$, returns $\mathrm{DLog}_g(W)$. Let $W_1, \ldots, W_q$ denote the responses to A's queries to its challenge oracle. A's goal is to compute the discrete logarithms of all challenges, meaning output $w_1, \ldots, w_q \in \mathbb{Z}_p$ satisfying $g^{w_i} = W_i$ for all $1 \le i \le q$. Of course this is easy because it has a $\mathrm{DLog}_g(\cdot)$ oracle. To make the task non-trivial, however, we restrict A to make strictly less queries to its $\mathrm{DLog}_G(\cdot)$ oracle than it does to its challenge oracle. Let $\mathbf{Adv}^{\mathrm{omdl}}_G(A)$ be the probability that A wins.

Schnorr identification protocol. The Schnorr identification protocol [35] shown in Figure 3 is probably the most "canonical" example of a canonical identification protocol. It is secure against impersonation under passive attack under the dl assumption [35]. Security against impersonation under active (and concurrent) attack, however, remained an open question for a while. Indeed, it does not seem possible to prove this under the dl assumption. Eventually, however, security against impersonation under active and concurrent attack was proved by [4] under the one more dl (omdl) assumption. However, we need special soundness rather than security under impersonation. Also, we need to show that our strong form of special soundness holds, namely that the adversary not only cannot find a pair of accepting transcripts $(\textsc{Cm}, \textsc{Ch}_1, \textsc{Rp}_1), (\textsc{Cm}, \textsc{Ch}_2, \textsc{Rp}_2)$ with $\textsc{Ch}_1 \neq \textsc{Ch}_2$ but cannot even find such transcripts with $\textsc{Ch}_1 = \textsc{Ch}_2$ as long as $\textsc{Rp}_1 \neq \textsc{Rp}_2$. We revisit the proof to establish these things. We make use of the fact that the protocol has a "unique answer" property. The proof of the following may be found in [5].

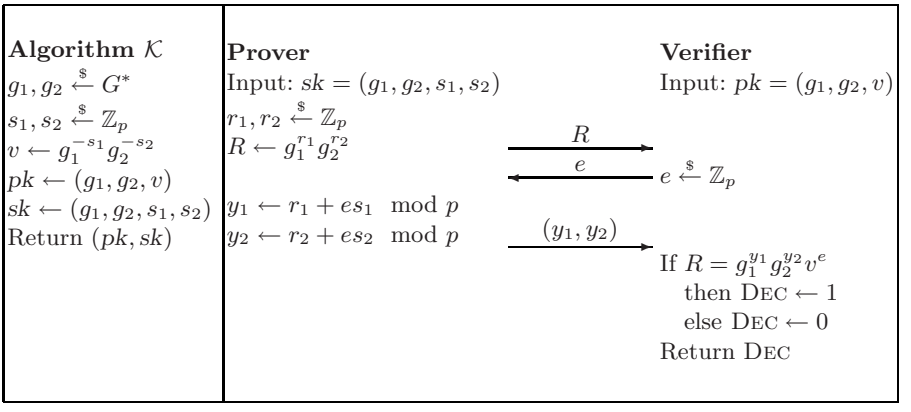

Fig. 4. Okamoto Protocol. Above, G is a group of prime order p, and $\mathrm{ChSet} = \mathbb{Z}_p$.

Proposition 1. *Let* $\mathsf{ID} = (\mathcal{K}, P, \mathrm{ChSet}, V)$ *be the Schnorr identification protocol described in Figure 3. Let* I *be an adversary against the special soundness of* ID *under concurrent attack. Then there exists an omdl adversary* A *such that*

$$\mathbf{Adv}^{\mathrm{ss-ca}}_{\mathsf{ID}}(I) \;\leq\; \mathbf{Adv}^{\mathrm{omdl}}_{G}(A)\,.$$

Furthermore the running time of A *is that of* I *plus some overhead to compute an inverse and product modulo* p*, and if* I *activates* q *clones, then* A *makes* $q+1$ *challenge queries.*

We remark that the reduction is tight. In contrast, in the reductions showing security against impersonation [4], $\mathbf{Adv}^{\mathrm{omdl}}_{G}(A)$ is proportional to the *square* of the probability that I succeeds in impersonation. This is an advantage to working directly with special soundness rather than with impersonation. We now sketch a proof based on the ideas of [4].

The two-tier scheme resulting from our FS-based transform instantiated with the Schnorr protocol is very efficient. Generating a secondary key pair takes just one group exponentiation, while signing only requires a multiplication modulo p. In the context of our uf-cma to suf-cma transform of Section 3, this means that the computational overhead for signing (added cost of signing in the suf-cma scheme versus the uf-cma scheme) is just one group exponentiation and the bandwidth overhead (added length of a signature in the suf-cma scheme compared to that in the uf-cma scheme) is one group element and one integer modulo p.

Okamoto Identification Protocol. Okamoto's protocol [31] is illustrated in Figure 4. Its advantage is that security can be proved under the standard dl assumption rather than the omdl assumption. (Yet in fact the efficiency is not much different as we will see below.) The idea is that there are many secret keys corresponding to a single public key and witness-indistinguishability [17] can be used in the simulation. The protocol was proved in [31] to be secure against impersonation under active attack assuming hardness of the dl problem, and the

proof extends to concurrent attacks. However, again, we need special soundness rather than security under impersonation, and in our new, strong form. The Okamoto protocol, however, does not have the unique answer property. But we can still prove the security we need. We now state the result. The proof may be found in [5].

Proposition 2. *Let* $\mathsf{ID} = (\mathcal{K}, P, \mathrm{ChSet}, V)$ *be the Okamoto identification protocol described in Figure 4. Let* I *be an adversary against the special soundness of* ID *under concurrent attack. Then there exists a dl adversary* A *such that*

$$\mathbf{Adv}^{\mathrm{ss-ca}}_{\mathsf{ID}}(I) \;\leq\; \frac{1}{p} + \mathbf{Adv}^{\mathrm{dl}}_{G}(A)\ .$$

Furthermore the running time of A *is that of* I *plus the time to compute three inverses and three products modulo* p*.* ∎

Again, the reduction is essentially tight due to working with special soundness, whereas the reduction of [31] to establish security against impersonation incurs the square loss we discussed in the context of Schnorr.

In the two-tier scheme resulting from our FS-transform instantiated with the Okamoto protocol, generating a secondary key pair takes one group exponentiation. (It is a multi-exponentiation, which has the same cost as a single one.) Signing requires a couple of multiplications modulo p. So the computational cost is the same as for Schnorr although security relies only on dl rather than omdl. In the context of our uf-cma to suf-cma transform of Section 3, this means that the computational overhead for signing is again just one group exponentiation. But the bandwidth overhead is one group element and two integers modulo p, slightly more than when we used the Schnorr scheme.

Additional Protocols. Above we have discussed two protocols that meet our ss-ca security requirement. We have however identified several more with the property in question. We omit proofs since they are similar to the ones given here, and instead provide a brief discussion. We exclude the Fiat-Shamir protocol from this discussion, as it does not seem to meet our requirements.

The GQ protocol [24] was proved secure against impersonation under concurrent attack in [4] under the assumption that RSA is secure against one more inversion [3]. We can extend this proof to show it is ss-ca under the same assumption in the same way that we extended the proof of the Schnorr scheme. This protocol has Fiat-Shamir like efficiency yet has small key sizes.

Shamir presented an identity-based identification scheme in [36]. A corresponding standard (i.e. not identity-based) version was presented in [2], along with a variant they called Sh* and proved secure against impersonation under concurrent attack assuming security of RSA under one more inversion. This too can be proved ss-ca under the same assumption. The protocol is however a mirror image of GQ and has the same efficiency attributes as the latter.

Then there are pairings-based schemes. Both Hs-SI [2] and ChCh-SI [2] are ss-ca secure under the one more computational Diffie-Hellman assumption. These schemes were presented in [2] and are based upon existing identity-based

signature schemes, namely those of Hess [25] and Cha and Cheon [10]. Again, the proof of ss-ca extends the proofs of security against impersonation of [2].

Acknowledgements

The first author was supported in part by NSF grant CNS-0524765 and a gift from Intel Corporation. The second author was supported by a UCSD Powell fellowship and the above mentioned grants of the first author.

References

1. M. Abdalla, J. H. An, M. Bellare, and C. Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In *EUROCRYPT 2002*, volume 2332 of *LNCS*. Springer-Verlag.
2. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. In *EUROCRYPT 2004*, volume 3027 of *LNCS*. Springer-Verlag.
3. M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, 2003.
4. M. Bellare and A. Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *CRYPTO 2002*, volume 2442 of *LNCS*. Springer-Verlag.
5. M. Bellare and S. Shoup. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. Full version of current paper. Available from authors' web pages.
6. D. Bleichenbacher and U. Maurer. On the efficiency of one-time digital signatures. In *ASIACRYPT'96*, volume 1163 of *LNCS*. Springer-Verlag.
7. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO 2004*, volume 3152 of *LNCS*. Springer-Verlag.
8. D. Boneh, E. Shen, and B. Waters. Strongly unforgeable signatures based on computational diffie-hellman. In *PKC 2006*, volume 3958 of *LNCS*. Springer-Verlag.
9. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004*, volume 3027 of *LNCS*. Springer-Verlag.
10. J. C. Cha and J. H. Cheon. An identity-based signature from gap Diffie-Hellman groups. In *PKC 2003*, volume 2567 of *LNCS*. Springer-Verlag.
11. R. Cramer and I. Damgård. Secure signature schemes based on interactive protocols. In *CRYPTO'95*, volume 963 of *LNCS*. Springer-Verlag.
12. W. Dai. Crypto++ Library. `http://www.cryptopp.com/`
13. Y. Dodis and J. Katz. Chosen-ciphertext security of multiple encryption. In *TCC 2005*, volume 3378 of *LNCS*. Springer-Verlag.
14. D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
15. S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9(1):35–67, 1996.
16. U. Feige, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.

17. U. Feige and A. Shamir. Witness indistinguishable and witness hiding protocols. In *22nd ACM STOC*, 1990. ACM Press.
18. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO'86*, volume 263 of *LNCS*. Springer-Verlag.
19. O. Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004.
20. O. Goldreich and H. Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, Feb. 1996.
21. S. Goldwasser and Y. T. Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th FOCS*, 2003. IEEE Computer Society Press.
22. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, Apr. 1988.
23. L. C. Guillou and J.-J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *EUROCRYPT'88*, volume 330 of *LNCS*. Springer-Verlag.
24. L. C. Guillou and J.-J. Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In *CRYPTO'88*, volume 403 of *LNCS*. Springer-Verlag.
25. F. Hess. Efficient identity based signature schemes based on pairings. In *SAC 2002*, volume 2595 of *LNCS*. Springer-Verlag.
26. Q. Huang, D. S. Wong, and Y. Zhaoe. Generic transformation to strongly unforgeable signatures. Cryptology ePrint Archive, Report 2006/346, 2006. `http://eprint.iacr.org/2006/346`.
27. Y. Lindell. A simpler construction of cca2-secure public-key encryption under general assumptions. *Journal of Cryptology*, 19(3):359–377, 2006.
28. S. Micali, K. Ohta, and L. Reyzin. Accountable-subgroup multisignatures. In *ACM CCS 01*, 2001. ACM Press.
29. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM STOC*, 1989. ACM Press.
30. K. Ohta and T. Okamoto. On concrete security treatment of signatures derived from identification. In *CRYPTO'98*, volume 1462 of *LNCS*. Springer-Verlag.
31. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO'92*, volume 740 of *LNCS*. Springer-Verlag.
32. D. Pointcheval and J. Stern. Security proofs for signature schemes. In *EUROCRYPT'96*, volume 1070 of *LNCS*. Springer-Verlag.
33. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
34. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, 1990. ACM Press.
35. C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
36. A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO'84*, volume 196 of *LNCS*. Springer-Verlag.
37. I. Teranishi, T. Oyama, and W. Ogata. General conversion for obtaining strongly existentially unforgeable signatures. In *INDOCRYPT 2006*, LNCS.
38. B. R. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT 2005*, volume 3494 of *LNCS*.