An extended abstract of this paper appears in Advances in Cryptology — ASIACRYPT '06, Lecture Notes in Computer Science Vol. 4284, Xuejia Lai ed., Springer-Verlag, 2006. This is the full version.

On the Generic Construction of Identity-Based Signatures with Additional Properties

David Galindo[1] Javier Herranz[2] Eike Kiltz[3]

[1] University of Luxembourg,
Luxembourg City, Luxembourg
david.galindo@uni.lu
http://www.dgalindo.es/

[2] IIIA-CSIC
Bellaterra, Spain
jherranz@iiia.csic.es
http://www.iiia.csic.es/~jherranz/

[3] CWI Amsterdam
The Netherlands
kiltz@cwi.nl
http://kiltz.net

Abstract

It has been demonstrated by Bellare, Neven, and Namprempre (Eurocrypt 2004) that identity-based signature schemes can be generically constructed from standard digital signature schemes. In this paper we consider the following natural extension: is there a generic construction of "identity-based signature schemes with additional properties" (such as identity-based blind signatures, verifiably encrypted signatures, ...) from standard signature schemes with the same properties? Our results show that this is possible for a number of properties including proxy signatures; (partially) blind signatures; verifiably encrypted signatures; undeniable signatures; forward-secure signatures; (strongly) key insulated signatures; online/offline signatures; threshold signatures; and (with some limitations) aggregate signatures.

Using well-known results for standard signature schemes, we conclude that explicit identity-based signature schemes with additional properties can be constructed, enjoying sometimes better properties than specific schemes proposed until know. In particular, our work implies the existence of identity-based signatures with additional properties that are provably secure in the standard model, do not need bilinear pairings, or can be based on general assumptions.

Keywords: Signatures with Additional Properties, Identity-Based Cryptography.

Contents

1 Introduction **1**

1.1 Our Results . . . 2

1.2 Organization of the Paper . . . 3

2 Digital Signatures **3**

2.1 Standard Signatures . . . 4

2.2 Identity-Based Signatures . . . 4

3 Generic Construction of Identity-based Signatures **5**

4 Generic Construction of Identity-Based Signatures with Additional Properties **5**

4.1 Verifiably Encrypted Signatures . . . 6

4.2 (Partially) Blind Signatures . . . 7

4.3 Undeniable Signatures . . . 8

4.4 Forward-Secure Signatures . . . 10

4.5 (Strongly) Key Insulated Signatures . . . 11

4.6 Proxy Signatures . . . 11

4.7 Online/Offline Signatures . . . 12

4.8 Threshold Signatures . . . 12

4.9 Aggregate Signatures . . . 14

4.10 Limitations and Extensions . . . 15

5 Generic Construction of Identity-Based Blind Signatures **15**

5.1 Blind Signature Schemes . . . 16

5.2 Identity-Based Blind Signature Schemes . . . 17

5.3 Construction . . . 18

5.4 Security Analysis . . . 19

5.5 Example Instantiation . . . 22

6 Conclusions **24**

1 Introduction

Digital signatures are one of the most fundamental concepts of modern cryptography. They provide authentication, integrity and non-repudiation to digital communications, which makes them the most used public key cryptographic tool in real applications. In order to satisfy the needs of some specific scenarios such as electronic commerce, cash, voting, or auctions, the original concept of digital signature has been extended and modified in multiple ways, giving raise to many kinds of what we call "digital signatures with additional properties", e.g. blind signatures, verifiably encrypted signatures, and aggregate signatures.

Initially, all these extensions were introduced for the framework based on standard Public Key Infrastructures (PKI), where each user generates a secret key and publishes the matching public key. In practice, digital certificates linking public keys with identities of users are needed to implement these systems, and this fact leads to some drawbacks in efficiency and simplicity. For this reason, the alternative framework of identity-based cryptography was introduced by Shamir [57]. The idea is that the public key of a user can be directly derived from his identity, and therefore digital certificates are avoidable. The user obtains his secret key by interacting with some trusted master entity. Shamir [57] already proposed an identity-based signature scheme. In contrast, the problem of designing an efficient and secure identity-based encryption scheme remained open until [16, 56]. Identity-based cryptographic tools started to be commercialized recently, and a first standard is forthcoming [1].

From a theoretical point of view, results concerning identity-based encryption schemes are more challenging than those concerning identity-based signatures (IBS). In contrast to the identity-based encryption case, it was already pointed out by Shamir [57] that a standard PKI-based signature scheme already implies an identity-based signature scheme by using the signature scheme twice: for generating user secret keys and for the actual signing process. More precisely, the user secret key of an identity consists of a fresh PKI-based signing/verification key and a certificate proving the validity of the signing key. The latter certificate is established by the master entity by signing (using the master signing key) the new verification key together with the user's identity. In the actual identity-based signing process the user employs the new signing key to sign the message. The identity-based signature itself consists of this signature along with the certificate and the public verification key.

Shamir's certificate-based idea was formalized by Bellare, Neven, and Namprempre in [9], where they proposed a generic and secure construction of identity-based signature schemes from any secure PKI-based signature scheme. However, some specific identity-based signature schemes have been proposed and published, mostly employing bilinear pairings and random oracles, without arguing if the proposed schemes are more efficient than the schemes resulting from the generic construction in [9]. In fact, in many papers the authors do not mention the generic approach from [9] and in spite of Shamir's work from more than two decades ago [57] it still seems to be a popular "opinion" among some researchers that bilinear pairings are inherent to identity-based signatures.

Our observation is that the situation is quite similar when identity-based signature schemes with additional properties are considered. Intuitively, such schemes may be obtained using the same generic approach as in the case of standard identity-based signatures, by combining a digital certificate and a PKI-based signature scheme with the desired additional property. To the best of our knowledge, this intuitive construction was never mentioned before, nor has a formal analysis been given up to now. Furthermore, specific identity-based signature schemes with additional properties keep being proposed and published without arguing which improvements they bring with respect to the possible generic certificate-based approach. Nearly all of these papers employ bilinear pairings and the security proofs are given in the random oracle model [11] (with its well-known limitations [20]).

1.1 Our Results

In this work we formally revisit this intuitive idea outlined in the last paragraph. Namely, if $\mathcal{s}$ is a secure PKI-based signature scheme and $\mathcal{s}_{\mathcal{P}}$ is a PKI-based signature scheme with some additional property $\mathcal{P}$, we pursue the question if for a certain property $\mathcal{P}$ the combination of those two signature schemes can lead to a secure IBS scheme $\mathcal{S}_{\mathcal{P}}$ enjoying the same additional property $\mathcal{P}$. We can answer this question to the positive, giving generic constructions of signature schemes with the following properties:

- Proxy signatures (PS)
- (Partially) blind signatures (PBS/BS)
- Verifiably encrypted signatures (VES)
- Undeniable signatures (US)
- Forward-secure signatures (FSS)
- Strong key insulated signatures (SKIS)
- Online/offline signatures (OOS)
- Threshold signatures (TS)
- Aggregate signatures (AS)[1]

Implications. By considering well-known results and constructions of PKI-based signatures $\mathcal{s}_{\mathcal{P}}$ with the required additional properties, we obtain identity-based schemes $\mathcal{S}_{\mathcal{P}}$ from weaker assumptions than previously known. A detailed overview of our results can be looked up in Table 1 on page 6. To give a quick overview of our results, for nearly every property $\mathcal{P}$ listed above, we obtain (i) the first $\mathcal{S}_{\mathcal{P}}$ scheme secure in the standard model (i.e., without random oracles); (ii) the first $\mathcal{S}_{\mathcal{P}}$ scheme built without using bilinear pairings; and (iii) the first $\mathcal{S}_{\mathcal{P}}$ based on "general assumptions" (e.g. on the sole assumption of the existence of one-way functions), answering the main foundational question with regard to these primitives. Our results therefore implicitly resolve many "open problems" in the area of identity-based signatures with additional properties.

Generic Constructions. For some properties $\mathcal{P}$ the construction of the scheme $\mathcal{S}_{\mathcal{P}}$ is the same as in [9] and a formal security statement can be proved following basically verbatim the proofs given in [9]. But as the limitations of the generic approach indicate, this approach does not work in a black-box way for every possible property $\mathcal{P}$. For some special properties the certificate-based generic construction sketched above has to be (non-trivially) adapted to fit the specific nature of the signature scheme. This is in particular the case for blind signatures and hence in this case we will lay out our construction in complete detail.

Limitations. On the other hand the generic way of constructing identity-based signatures with additional properties is not sound for every property. In particular, it does not seem to be applicable when, in the PKI-based scheme $\mathcal{s}_{\mathcal{P}}$, an additional public key different from that of the signer has to be used in the protocol. This includes ring, designated verifier, confirmer, nominative or chameleon signatures. For these kinds of signatures, therefore, it makes more sense to consider specific constructions in the identity-based framework.

Discussion. We think that in some cases the constructions of identity-based signatures with additional properties implied by our results are at least as efficient as most of the schemes known

[1] We stress that the length of our implied aggregated identity-based signatures is still depending linearly on the number of different signers (optimally it is constant) and therefore our results concerning AS are not optimal.

before. However, because of the huge number of cases to be considered, we decided not to include a detailed efficiency analysis of our generic constructions. Note that, in order to analyze the efficiency of a particular identity-based scheme resulting from our construction, we should first fix the framework: whether we admit the random oracle model, whether we allow the use of bilinear pairings, etc. Then we should take the most efficient suitable PKI-based signature scheme and measure the efficiency of the resulting identity-based signature. Our point is rather that this comparison should be up to the authors proposing new specific schemes: the schemes (explicitly and implicitly) implied by our generic approach should be used as benchmarks relative to which both existing and new practical schemes measure their novelty and efficiency.

We stress that we do not claim the completely novelty of our generic approaches to construct identity-based signatures with additional properties. Similar to [9] we rather think that most of these constructions can be considered as folklore and are known by many researchers. However, the immense number of existing articles neglecting these constructions was our initial motivation for writing this paper. We think that our results may also help better understanding IBS. To obtain a practical IBS with some additional properties the "standard method" in most articles is to start from a standard IBS and try to "add in" the desired additional property. Our results propose that one should rather start from a standard signature scheme with the additional property and try to make it identity-based. We hope that the latter approach may be used to obtain more efficient practical schemes.

1.2 Organization of the Paper

In Section 2 we recall the basic definitions (protocols and security requirements) about signature schemes, in both the PKI-based and the identity-based frameworks. We recall the generic construction of identity-based signatures of [9] in Section 3. Then we present our main results in Section 4: we list those additional properties $\mathcal{P}$ which can be preserved by a generic construction of identity-based signatures and present the transformations. We also discuss why this approach does not seem to work for other additional properties. As a representative example, we give in Section 5 the details concerning the (identity-based) blind signature case, including the generic construction, security analysis, and a specific (and very efficient) realization. Finally, some concluding remarks are given in Section 6.

2 Digital Signatures

In this section we recall the well-known syntax and definition of standard (PKI-based) and identity-based signature schemes. We make the convention that everything related to standard signatures is written in lower-case, whereas for everything related to identity-based signatures we use upper-case.

We introduce some notation to be used throughout this paper. If x is a string, then $|x|$ denotes its length, while if S is a set then $|S|$ denotes its size. If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If S is a set then $s \stackrel{\$}{\leftarrow} S$ denotes the operation of picking an element s of S uniformly at random. We write $\mathsf{A}(x, y, \ldots)$ to indicate that A is an algorithm with inputs $x, y, \ldots$ and by $z \stackrel{\$}{\leftarrow} \mathsf{A}(x, y, \ldots)$ we denote the operation of running A with inputs $(x, y, \ldots)$ and letting z be the output. With PPT we denote probabilistic polynomial time.

2.1 Standard Signatures

A standard signature [36] (SS) scheme is defined as a tuple of PPT algorithms $s = (\mathsf{kg}, \mathsf{sign}, \mathsf{vfy})$. The first two may be randomized but the last is not. The key generation algorithm kg, on input 1^k, generates a key pair (pk, sk). The signer creates a signature on a message m via $sig \xleftarrow{\$} \mathsf{sign}(sk, m)$, and the verifier can check the validity of a signature by testing whether $\mathsf{vfy}(pk, m, sig) = 1$. Correctness requires that $\mathsf{vfy}(pk, m, \mathsf{sign}(sk, m)) = 1$ with probability one for all $m \in \{0,1\}^k$ whenever the keys (pk, sk) are generated as indicated above.

For security we consider the notion of unforgeability under chosen-message attacks [36], which is defined through an experiment with a forger F, parameterized with the security parameter k. The experiment begins with the generation of a fresh key pair $(pk, sk) \xleftarrow{\$} \mathsf{kg}(1^k)$. The forger F is run on input the public key pk, and has access to a signing oracle:

- $\mathsf{sign}(\cdot)$: On input $m \in \{0,1\}^*$, this oracle returns a signature $sig \xleftarrow{\$} \mathsf{sign}(sk, m)$.

At the end of its execution, the forger outputs a message m^* and a forged signature sig^*. We say that F wins the game if sig^* is a valid forgery of message m^* (i.e., if $1 \leftarrow \mathsf{vfy}(pk, m^*, sig^*)$) and if $m^* \neq m$ for all the messages m for which F queried the signature, during the attack. We define the advantage of such a forger F as $\mathbf{Adv}^{\text{forge}}_{s,\mathsf{F}}(k) = \Pr[\,\mathsf{F} \text{ succeeds}\,]$. A SS scheme is called secure if the advantage of any PPT forger F is a negligible function in k.

For the notion of *strong* unforgeability under chosen-message attacks we relax the second condition such that we require $(m^*, sig^*) \neq (m, sig)$ for all the tuples (m, sig) that F has obtained during the attack. Again, a scheme is called strongly secure if the respective advantage is a negligible function in k, for any PPT forger F.

2.2 Identity-Based Signatures

An *identity-based signature (IBS) scheme* is a tuple of PPT algorithms $\mathcal{S} = (\mathsf{KG}, \mathsf{EXT}, \mathsf{SIGN}, \mathsf{VFY})$. The first three may be randomized but the last is not. The trusted key distribution center runs the key-generation algorithm KG on input 1^k to obtain a master public and secret key pair (PK, SK). To generate a user secret key $USK[id]$ for the user with identity $id \in \{0,1\}^*$, it runs the key derivation algorithm EXT on input SK and id. The user secret key is assumed to be securely communicated to the user in question. On input $USK[id]$ and a message m, the signing algorithm SIGN returns a signature SIG of m. On input PK, id, m, and a signature SIG, the verification algorithm VFY returns 1 if SIG is valid for id and m, and returns 0 otherwise. Correctness requires that $\mathsf{VFY}(PK, id, m, \mathsf{SIGN}(USK[id], m)) = 1$ with probability one for all $k \in \mathbb{N}$ and $id, m \in \{0,1\}^*$ whenever the keys $PK, SK, USK[id]$ are generated as indicated above.

For security we consider the notion of existential unforgeability under chosen-message and chosen-identity attacks [35], which is defined through an experiment with a forger F, parameterized with the security parameter k. The experiment begins with the generation of a fresh master key pair $(PK, SK) \xleftarrow{\$} \mathsf{KG}(1^k)$. The forger F is run on input the master public key PK, and has access to the following oracles:

- $\mathsf{EXT}(\cdot)$: On input identity $id \in \{0,1\}^*$, this oracle first checks if it has already established a user secret key $USK[id]$ for id. If so it returns this same user secret key; otherwise, it returns a fresh user secret key $USK[id] \xleftarrow{\$} \mathsf{EXT}(SK, id)$.

- $\mathsf{SIGN}(\cdot,\cdot)$: On input identity $id \in \{0,1\}^*$ and message $m \in \{0,1\}^*$, this oracle returns a signature $SIG \xleftarrow{\$} \mathsf{SIGN}(USK[id], m)$ where $USK[id]$ is computed using the above oracle $\mathsf{EXT}(\cdot)$.

Algorithm $\mathsf{KG}(1^k)$
  $(SK, PK) \xleftarrow{\$} \mathsf{kg}(1^k)$
  Return (SK, PK)

Algorithm $\mathsf{EXT}(SK, id)$
  $(pk, sk) \xleftarrow{\$} \mathsf{kg}'(1^k)$
  $cert \xleftarrow{\$} \mathsf{sign}(SK, id \,\|\, pk)$
  Return $USK[id] \leftarrow (cert, pk, sk)$

Algorithm $\mathsf{SIGN}(id, USK[id], m)$
  Parse $(cert, pk, sk) \leftarrow USK[id]$
  $sig \xleftarrow{\$} \mathsf{sign}'(sk, m)$
  Return $SIG = (cert, pk, sig)$

Algorithm $\mathsf{VFY}(PK, id, m, SIG)$
  Parse $(cert, pk, sig) \leftarrow SIG$
  If $\mathsf{vfy}(PK, id \,\|\, pk, cert) = 0$ then return 0
  If $\mathsf{vfy}'(pk, m, sig) = 0$ then return 0
  Else return 1.

Figure 1: Generic transformation SS-2-IBS from standard signatures to identity-based signatures.

At the end of its execution, the forger outputs an identity id^*, a message m^* and a forged signature SIG^*. The forger is said to win the game if $\mathsf{VFY}(PK, id^*, m^*, SIG^*) = 1$ and F never queried $\mathsf{EXT}(id^*)$ or $\mathsf{SIGN}(id^*, m^*)$. The advantage $\mathbf{Adv}^{\text{forge}}_{\mathcal{S},\mathsf{F}}(k)$ is defined as the probability that F wins the game, and $\mathcal{S}$ is said to be secure if this advantage is negligible in k for any PPT forger F.

3 Generic Construction of Identity-based Signatures

In this section we first outline the generic transformation [57, 9] from two standard signature schemes s, s' into an identity-based signature scheme $\mathcal{S}$. Subsequently we study the question whether, for different types of signature schemes $s_{\mathcal{P}}$ with additional properties $\mathcal{P}$, we have a (similar) generic transformation that combines s with $s_{\mathcal{P}}$ to obtain $\mathcal{S}_{\mathcal{P}}$, where $\mathcal{S}_{\mathcal{P}}$ is an identity-based signature scheme with the same additional property as $s_{\mathcal{P}}$.

Let $s = (\mathsf{kg}, \mathsf{sign}, \mathsf{vfy})$ and $s' = (\mathsf{kg}', \mathsf{sign}', \mathsf{vfy}')$ be two (possibly equal) standard signature schemes. We build $\mathcal{S} = (\mathsf{KG}, \mathsf{EXT}, \mathsf{SIGN}, \mathsf{VFY}) = \mathsf{SS\text{-}2\text{-}IBS}(s, s')$ as given in Figure 1.

Bellare, Namprempre, and Neven [9] prove the following result:

Theorem 3.1 If s and s' are both secure standard signature schemes then $\mathcal{S} = \mathsf{SS\text{-}2\text{-}IBS}(s, s')$ is a secure identity-based signature scheme.

Let $s_{\mathcal{P}}$ be a standard signature scheme with the property $\mathcal{P}$. We extend the above construction to an IBS with additional properties $\mathcal{S}_{\mathcal{P}}$ in a straightforward way: as with signing/verification, all functionality provided by $s_{\mathcal{P}}$ is "lifted" to the identity-based case. That means that (analog to SIGN and VFY) any protocol additionally provided by $s_{\mathcal{P}}$ is executed using the corresponding secret/public key pair (sk, pk) contained in $USK[id]$. We will refer to the latter construction as the "generic construction of identity-based signatures with additional properties" or simply "generic construction".

4 Generic Construction of Identity-Based Signatures with Additional Properties

In this section we will demonstrate that the generic construction and variants of it can indeed be used for many signatures schemes with additional properties: proxy signatures (PS); (partially) blind signatures (BS); verifiable encrypted signatures (VES); undeniable signatures

Signature type	Existence of identity-based signature schemes with additional properties			
	at all (formal proof)?	w/o random oracles?	w/o pairings?	general assumptions?
VES §4.1	⋆	★	★	★
BS §4.2	⋆/★[a]	⋆[b]	★	★
US §4.3	⋆	★	★	–
FSS §4.4	★	★	★	★
SKIS §4.5	⋆	★	★	★
PS §4.6	⋆	★	★	★
OOS §4.7	⋆	★	★	★
TS §4.8	⋆	★	★	★
AS[c] §4.9	⋆	–	–	–

[a]against concurrent adversaries.
[b]recent result, [53].
[c]no interaction among the signers, signature length $\leq \mathcal{O}(\#\text{signers})$.

Table 1: A summary of the practical implications of our results. Here "⋆" means that a scheme was known before, a "★" means that our construction gives the first such scheme, and a "–" means that no such scheme is known.

(US); forward-secure signatures (FSS); strongly key insulated signatures (SKIS); online/offline signatures (OOS); threshold signatures (TS); and aggregate signatures (AS). Since for most properties the generic construction can be applied without many difficulties (maybe excepting undeniable, blind and aggregate signatures), we decided to present our results in a rather informal way. However, as a representative example we will provide a full formal treatment of the generic construction of identity-based blind signatures in Section 5. We stress that we can treat the rest of our results at the same level of formality.

In Table 1 we summarize the practical impact of our results, i.e., we show what types $\mathcal{S}_{\mathcal{P}}$ of new identity-based signature schemes are implied by our general constructions.

4.1 Verifiably Encrypted Signatures

VES schemes enable a user Alice to create a signature encrypted using an adjudicator's public key (the VES signature), so that it can be publicly verified whether the encrypted signature is valid. The adjudicator is a trusted third party, who can reveal the standard signature when needed. VES schemes provide an efficient way to enable fairness in many practical applications such as contract signing.

An efficient VES scheme in the random oracle model based on pairings was given in [17], while the scheme in [49] is secure in the standard model. It was further noted in [49] that VES schemes can be constructed on general assumptions such as trapdoor one-way permutations.

Identity-based verifiably encrypted signature (IB-VES) schemes were introduced in [37], where also a concrete security model was proposed. In contrast to [37], here we only consider a weaker (but still reasonable) model where the adjudicator has a fixed public key *apk*, i.e., it is not identity-based.

Compared to a standard signature scheme s, a VES scheme $\mathit{s}_{\mathcal{VES}} = (\mathsf{kg}_{\mathcal{VES}}, \mathsf{sign}_{\mathcal{VES}}, \mathsf{vfy}_{\mathcal{VES}}, \mathsf{asign}_{\mathcal{VES}}, \mathsf{avfy}_{\mathcal{VES}}, \mathsf{adj}_{\mathcal{VES}})$ has three additional algorithms: VES signing $\mathsf{asign}_{\mathcal{VES}}$ and verification $\mathsf{avfy}_{\mathcal{VES}}$ (both with respect to an adjudicator's public key *apk*), and adjudication $\mathsf{adj}_{\mathcal{VES}}$. Here the adjudication algorithm $\mathsf{adj}_{\mathcal{VES}}$ inputs an adjudicator's secret key *ask* and transforms a VES signature into a standard signature. For our generic construction of an identity-based VES scheme $\mathcal{S}_{\mathcal{VES}} = (\mathsf{KG}_{\mathcal{VES}}, \mathsf{EXT}_{\mathcal{VES}}, \mathsf{SIGN}_{\mathcal{VES}}, \mathsf{VFY}_{\mathcal{VES}}, \mathsf{ASIGN}_{\mathcal{VES}}, \mathsf{AVFY}_{\mathcal{VES}}, \mathsf{ADJ}_{\mathcal{VES}})$, the

Algorithm $\mathsf{ASIGN}_{\mathcal{VES}}(apk, id, USK[id], m)$
  Parse $(cert, pk, sk) \leftarrow USK[id]$
  $sig_{\mathcal{VES}} \stackrel{\$}{\leftarrow} \mathsf{asign}_{\mathcal{VES}}(apk, sk, m)$
  Return $SIG_{\mathcal{VES}} = (cert, pk, sig_{\mathcal{VES}})$

Algorithm $\mathsf{ADJ}_{\mathcal{VES}}(ask, id, SIG_{\mathcal{VES}})$
  Parse $(cert, pk, sig_{\mathcal{VES}}) \leftarrow SIG_{\mathcal{VES}}$
  $sig \stackrel{\$}{\leftarrow} \mathsf{adj}_{\mathcal{VES}}(ask, sig_{\mathcal{VES}}, m)$
  Return $SIG = (cert, pk, sig)$

Algorithm $\mathsf{AVFY}_{\mathcal{VES}}(PK, id, m, SIG_{\mathcal{VES}})$
  Parse $(cert, pk, sig_{\mathcal{VES}}) \leftarrow SIG_{\mathcal{VES}}$
  If $\mathsf{vfy}(PK, id \,\|\, pk, cert) = 0$ then return 0
  If $\mathsf{avfy}_{\mathcal{VES}}(pk, apk, m, sig_{\mathcal{VES}}) = 0$ then return 0
  Else return 1.

Figure 2: Generic construction of identity-based VES.

first four algorithms are the same as in Figure 1. Only $\mathsf{KG}_{\mathcal{VES}}$ aditionally generates the adjudicator key-pair (ask, apk). VES signing and verification algorithms can be lifted to the identity-based case in the same way as in the generic construction, i.e., in the IB-VES signing protocol $\mathsf{ASIGN}_{\mathcal{VES}}$ one replaces sig with its VES counterpart $sig_{\mathcal{VES}}$ obtained by running the VES signing algorithm $\mathsf{asign}_{\mathcal{VES}}$ on sk, m, and the adjudicator's public key apk. IB-VES verification $\mathsf{AVFY}_{\mathcal{VES}}$ first checks the certificate and then the VES signature using the standard VES verification algorithm $\mathsf{avfy}_{\mathcal{VES}}$. Details can be found in Figure 2. We remark that since we only consider a standard (non identity-based) adjudicator there is no need to make the adjudication process "identity-based". We can prove the following theorem:

Theorem 4.1 If s is a secure standard signature scheme and $s_{\mathcal{VES}}$ is a secure verifiably encrypted signature scheme then the generic construction gives a secure identity-based verifiably encrypted signature scheme $\mathcal{S}_{\mathcal{VES}}$.

A pairing-based IB-VES scheme secure in the random oracle model was given in [37]. We note that the IB-VES scheme from [24] does not have a formal security proof. Using our generic construction we get an IB-VES scheme, in the standard model, based on any trapdoor one-way function [49]; a more efficient scheme, in the random oracle model, can be obtained by using the construction in [17].

4.2 (Partially) Blind Signatures

In blind signature (BS) schemes [21] a user can ask a signer to blindly sign a (secret) message m. At the end of the (interactive) signing process, the user obtains a valid signature on m, but the signer has no information about the message he has just signed. A formal security model of blind signatures was introduced in [42, 54]. Partially blind signature schemes are a variation of this concept, where the signer can include some common information in the blind signature, under some agreement with the final receiver of the signature. This concept was introduced in [2] and the security of such schemes was formalized in [3].

The first identity-based blind signature (IB-BS) schemes were proposed in [65, 64]. They employ bilinear pairings, but their security is not formally analyzed. Subsequent schemes were proposed in [26] but security is only provided in a weaker model (i.e., against sequential adversaries). The only IB-BS schemes with provable security in the strongest security model are the one in [61] (random oracle model) and the one in [53] (standard model), both using bilinear pairings. We take the case of blind signatures to exemplify, in detail, how our generic construction of identity-based signature schemes with additional properties works: in Section 5 we give all necessary formal definitions, our generic construction, and a formal security analysis. The case of partially blind signatures can be analyzed in a very similar way. Summing up, and quite informally, we will obtain the following general result (see Section 5 for the details).

Theorem 4.2 If $\mathcal{s}$ is a *strongly secure* standard signature scheme and $\mathcal{s}_{\mathcal{P}}$ is a secure (partially) blind signature scheme then a secure identity-based (partially) blind signature scheme $\mathcal{S}_{\mathcal{P}}$ can be constructed.

Here the IB-BS scheme inherits the security properties of the BS scheme — if BS is secure against concurrent adversaries so is IB-BS. In particular, we obtain new IB-BS schemes provably secure against concurrent adversaries in the standard model (by using the results from [19, 52, 32]); we obtain IB-BS schemes which do not employ bilinear pairings [10] and we obtain IB-BS schemes from any one-way trapdoor permutation [42, 32]. Furthermore, as we will show in Section 5.5, our generic construction, when instatiated with the Boneh-Lynn-Shacham signature scheme [18] and Boldyreva's BS scheme [13], leads to a very practical IB-BS scheme, in the random oracle model.

4.3 Undeniable Signatures

In undeniable signature schemes [23] (US), it is not possible to check the validity or invalidity of a signature without interacting with the signer. Undeniable signatures are used in applications where signed documents carry some private information about the signer and thus it is desirable to limit the ability of verification, to protect the privacy of the signer.

Following [28], an undeniable signature scheme $\mathcal{s}_{\mathcal{US}}$ consists of five algorithms $\mathcal{s}_{\mathcal{US}} = (\mathsf{kg}_{\mathcal{US}}, \mathsf{sign}_{\mathcal{US}}, \mathsf{conf}_{\mathcal{US}}, \mathsf{disav}_{\mathcal{US}}, \mathsf{sim}_{\mathcal{US}})$, where $\mathsf{conf}_{\mathcal{US}}$ and $\mathsf{disav}_{\mathcal{US}}$ are the confirmation and disavowal protocols respectively, both being interactive algorithms run between a prover and a verifier. The inputs of the verifiers in both $\mathsf{conf}_{\mathcal{US}}$ and $\mathsf{disav}_{\mathcal{US}}$ protocols are a message m, an alleged signature sig and a public key pk, while the input for the provers is the secret key sk. If $\mathsf{conf}_{\mathcal{US}}(sk, (pk, sig, m)) = 1$, then the pair (m, sig) is valid; if $\mathsf{disav}_{\mathcal{US}}(sk, (m, sig, pk)) = 1$, then the pair (m, sig) is invalid. The role and syntax of the algorithm $\mathsf{sim}_{\mathcal{US}}$ are explained below.

The basic security properties are (standard) *unforgeability*, *non-transferability* and *simulatability*. By non-transferability it is meant that no adversary should be able to convince any third party of the validity/invalidity of a given message/signature pair after having run the confirmation and disavowal protocols with the legitimate signer. Intuitively, this is captured by requiring the confirmation and disavowal protocols to be "zero-knowledge," so that no information is leaked beyond validity/invalidity. Simulatability is aimed at ensuring that signatures (represented as binary strings) can not be recognized by an attacker as such (i.e., distinguished from a uniform string). More formally, this property is fulfilled if there exists a simulator algorithm $\mathsf{sim}_{\mathcal{US}}$, which on input a public key and a message, outputs a simulated signature sig. This signature should look like a "real undeniable signature" to anyone who only knows public information and has access to confirmation/disavowal oracles. An additional security property for US schemes is that of anonymity. Roughly speaking, a scheme $\mathcal{s}_{\mathcal{US}}$ is said to be *anonymous* [33] if for two randomly generated key pairs $(pk_0, sk_0), (pk_1, sk_1)$ and a message m, it is infeasible to distinguish the two distributions $\mathsf{sign}_{\mathcal{US}}(sk_0, m)$ and $\mathsf{sign}_{\mathcal{US}}(sk_1, m)$.

Extending the previous definition to the identity-based setting, an identity-based undeniable signature (IB-US) scheme consists of a tuple of six algorithms $\mathcal{S}_{\mathcal{US}} = (\mathsf{KG}_{\mathcal{US}}, \mathsf{EXT}_{\mathcal{US}}, \mathsf{SIGN}_{\mathcal{US}}, \mathsf{CONF}_{\mathcal{US}}, \mathsf{DISAV}_{\mathcal{US}}, \mathsf{SIM}_{\mathcal{US}})$, where $\mathsf{CONF}_{\mathcal{US}}$ and $\mathsf{DISAV}_{\mathcal{US}}$ are interactive algorithms run between a prover P and a verifier V. The generic construction of IB-US is as follows. Algorithms $\mathsf{KG}_{\mathcal{US}}$ and $\mathsf{EXT}_{\mathcal{US}}$ are as in Figure 1 and algorithms $\mathsf{SIGN}_{\mathcal{US}}$ and $\mathsf{SIM}_{\mathcal{US}}$ are depicted in Figure 3. It remains to describe $\mathsf{CONF}_{\mathcal{US}}$ and $\mathsf{DISAV}_{\mathcal{US}}$. The interactive algorithm $\mathsf{CONF}_{\mathcal{US}}$ works as follows. The prover, on input $(USK[id], (id, SIG_{\mathcal{US}}, m))$ first parses $(cert, pk, sk) \leftarrow USK[id]$

Algorithm $\mathsf{SIGN}_{\mathcal{US}}(USK[id], m))$	Algorithm $\mathsf{SIM}_{\mathcal{US}}(PK, id, m)$
Parse $(cert, pk, sk) \leftarrow USK[id]$	$(pk', sk') \stackrel{\$}{\leftarrow} \mathsf{kg}_{\mathcal{US}}(1^k)$
$sig_{\mathcal{US}} \stackrel{\$}{\leftarrow} \mathsf{sign}_{\mathcal{US}}(sk, m)$	Return $\mathsf{sim}_{\mathcal{US}}(pk', m)$
Return $SIG_{\mathcal{US}} = sig_{\mathcal{US}}$	

Figure 3: Generic construction of identity-based US.

and sends $(cert, pk)$ to the verifier. The verifier, on input $(id, SIG_{\mathcal{US}}, m)$, receives $(cert, pk)$ and checks the validity of the certificate by running $\mathsf{vfy}(PK, id \,\|\, pk, cert)$. If it is correct, prover and verifier execute the interactive protocol $\mathsf{conf}_{\mathcal{US}}$. The construction of $\mathsf{DISAV}_{\mathcal{US}}$ is exactly the same, with the difference that, in the last step, prover and verifier run the protocol $\mathsf{disav}_{\mathcal{US}}$.

The basic security properties for an IB-US (unforgeability, non-transferability and simulatability), are defined by suitably adapting the standard US security notions to the identity-based scenario. In particular, the *identity-based simulatability* property is defined in terms of the existence of an additional simulation algorithm $\mathsf{SIM}_{\mathcal{US}}$. On input the public system parameters PK, an identity id and a message m, $\mathsf{SIM}_{\mathcal{US}}(PK, id, m)$ outputs a simulated identity-based signature SIG, which is indistinguishable from a real identity-based signature for someone having access to confirmation/disavowal oracles for the identity id.

In contrast to the generic construction from Figure 1, we construct an identity-based undeniable signature as $SIG \stackrel{\$}{\leftarrow} \mathsf{sign}_{\mathcal{US}}(sk, m)$, i.e., certificate $cert$ and pk are not included. Instead, in the interactive identity-based confirmation and disavowal protocols, the signer sends his certificate $cert$ and the public key pk to the verifier, who can then verify the link between SIG and $id \,\|\, pk$. Then prover (using sk) and verifier (using pk) execute the standard US confirmation/disavowal protocol. It is easy to see that a generic construction that would include $cert$ and pk in the signature would not be simulatable. Would this have been the case, the identity-based signature simulator should simulate the certificate $cert$ based on the master public-key only, which is infeasible since the signature scheme $\mathcal{s}$ is assumed to be unforgeable.

Note that we define the output of $\mathsf{SIM}_{\mathcal{US}}(PK, id, m)$ as $\mathsf{sim}_{\mathcal{US}}(pk', m)$, where $(pk', sk') \stackrel{\$}{\leftarrow} \mathsf{kg}_{\mathcal{US}}(1^k)$ is a fresh key pair generated by the simulator. The simulator $\mathsf{SIM}_{\mathcal{US}}(PK, id, m)$ does not input the user secret key $USK[id]$, and therefore the public key pk assigned to id is information theoretically hidden from the simulator's view. In contrast, an adversary may learn this public key pk by running the confirmation/disavowal protocol. It turns out that to ensure that our generic IB-US construction satisfies the simulatability property it is sufficient to require the scheme $\mathcal{US}$ to be anonymous in the sense of [33]. More formally, we can prove the following theorem:

Theorem 4.3 If $\mathcal{s}$ is a secure standard signature scheme and $\mathcal{s}_{\mathcal{US}}$ is a secure anonymous undeniable signature scheme then $\mathcal{S}_{\mathcal{US}}$ as outlined above is a secure identity-based undeniable signature scheme.

To the best of our knowledge, only one IB-US scheme has been previously presented in [48]. This scheme uses bilinear pairings and it is proved secure in the random oracle model. We stress that the security model in [48] seems to be incomplete, as the authors do not consider simulatability.

In [33], an anonymous US scheme based on the RSA primitive was proposed (the security proof uses the random oracle model). A different anonymous US scheme, whose security is proved in the standard model, can be found in [46]; it does not employ bilinear pairings, but the disavowal protocol is quite inefficient. Using these anonymous US schemes [33, 46], we can obtain secure IB-US schemes without bilinear pairings either in the random oracle model or in the standard model.

Algorithm $\mathsf{EXT}_{\mathcal{FSS}}(SK, id)$
  $(pk, sk_0) \stackrel{\$}{\leftarrow} \mathsf{kg}_{\mathcal{FSS}}(1^k)$
  $cert \stackrel{\$}{\leftarrow} \mathsf{sign}(SK, id \,\|\, pk)$
  Return $USK[id]_0 \leftarrow (cert, pk, sk_0)$

Algorithm $\mathsf{UPD}_{\mathcal{FSS}}(USK[id]_{t-1})$
  Parse $(cert, pk, sk_{t-1}) \leftarrow USK[id]_{t-1}$
  $sk_t \stackrel{\$}{\leftarrow} \mathsf{upd}_{\mathcal{FSS}}(sk_{t-1})$
  Return $USK[id]_t = (cert, pk, sk_t)$

Algorithm $\mathsf{SIGN}_{\mathcal{FSS}}(m, t, USK[id]_t)$
  Parse $(cert, pk, sk_t) \leftarrow USK[id]_t$
  $(sig_{\mathcal{FSS}}, t) \stackrel{\$}{\leftarrow} \mathsf{sign}_{\mathcal{TS}}(m, t, sk_t)$
  Return $SIG_{\mathcal{FSS}} = (cert, pk, sig_{\mathcal{FSS}}, t)$

Algorithm $\mathsf{VFY}_{\mathcal{FSS}}(PK, id, m, SIG_{\mathcal{FSS}})$
  Parse $(cert, pk, sig_{\mathcal{FSS}}, t) \leftarrow SIG_{\mathcal{FSS}}$
  If $\mathsf{vfy}(PK, id \,\|\, pk, cert) = 0$ then return 0
  If $\mathsf{vfy}_{\mathcal{FSS}}(pk, m, sig_{\mathcal{FSS}}, t) = 0$ then return 0
  Else return 1

Figure 4: Generic construction of identity-based FSS.

4.4 Forward-Secure Signatures

In a forward-secure signature (FSS) scheme the verification key remains fixed but the signing key is updated at regular intervals, in such a way that compromise of the signing key at a certain time period does not allow to forge signatures pertaining to any previous period.

A forward-secure signature scheme $s_{\mathcal{FSS}} = (\mathsf{kg}_{\mathcal{FSS}}, \mathsf{upd}_{\mathcal{FSS}}, \mathsf{sign}_{\mathcal{FSS}}, \mathsf{vfy}_{\mathcal{FSS}})$ has four algorithms. The key generation algorithm $\mathsf{kg}_{\mathcal{FSS}}$ outputs a public key pk and an initial secret key sk_0. For each time period $t = 1, \ldots, T$, the update protocol $sk_t \stackrel{\$}{\leftarrow} \mathsf{upd}_{\mathcal{FSS}}(sk_{t-1})$ produces a new secret key from the previous one. The protocols for signature and verification must include as input/output the time period where the signature has been computed. FSS schemes were introduced in [8], in order to mitigate the damage caused by key exposure without requiring redistribution of keys. Shortly after their introduction, a construction of FSS schemes from any signature scheme was proposed in [44]. In particular, this result implies that FSS schemes can be obtained from any one-way function.

To the best of our knowledge, the concept of identity-based forward-secure signature (IB-FSS) schemes has not been previously considered in the literature. In a IB-FSS scheme, the identity id of the signer remains fixed, while the signing key for the t-th time interval, $USK[id]_t$, is updated. The initial signing key $USK[id]_0$ is delivered to the user by the master entity, while the signing keys for the subsequent periods are generated by the user itself. Notice that this approach favorably compares with the usual way to defend against key exposure in identity-based cryptography, in which the master entity issues new private keys $USK[id \,\|\, t]$ to the user with identity id at every time period t. The latter approach heavily relies on the master entity and increases the (costly) communication between the entity and the users. Our generic construction allows us to obtain an identity-based forward secure signature scheme $\mathcal{S}_{\mathcal{FSS}} = (\mathsf{KG}_{\mathcal{FSS}},, \mathsf{EXT}_{\mathcal{FSS}}, \mathsf{UPD}_{\mathcal{VES}}, \mathsf{SIGN}_{\mathcal{FSS}}, \mathsf{VFY}_{\mathcal{VES}})$, starting from a standard signature scheme $s = (\mathsf{kg}, \mathsf{sign}, \mathsf{vfy})$ and a standard forward secure signature scheme $s_{\mathcal{FSS}}$. The protocol $\mathsf{KG}_{\mathcal{FSS}}$ is exactly the same as in Figure 1, resulting in a pair of keys (SK, PK) for the master entity. The other protocols are described in Figure 4.

Theorem 4.4 If s is a secure standard signature scheme and $s_{\mathcal{FSS}}$ is a secure forward-secure signature scheme then the generic construction gives a secure identity-based forward-secure signature scheme $\mathcal{S}_{\mathcal{FSS}}$.

As a consequence of this theorem, IB-FSS schemes can be constructed from any one-way function [44].

4.5 (Strongly) Key Insulated Signatures

The concept of (strongly) key insulated signatures (SKIS) was introduced in [30] and is quite similar to the concept of FSS. The main difference is that the update protocol is jointly executed by the user (signer) and an external entity (helper); in this way, compromise of the signing key at a certain time period t^* does not allow now to forge signatures pertaining to any other period $t \neq t^*$. We can easily adapt the generic construction of identity-based FSS that we have described in the previous section and we obtain a generic construction of identity-based SKIS.

SKIS signatures can be built from any one-way function [30], which implies that our generic construction yields identity-based SKIS schemes from any one-way function. Previously, identity-based SKIS using bilinear pairings and random oracles have been proposed in [66, 39].

4.6 Proxy Signatures

In proxy signature (PS) schemes, an original signer A delegates its signing capabilities to a proxy signer B, in such a way that B can sign (some specified set of) messages on behalf of A. The recipient of the final message verifies at the same time that B computed the signature and that A had delegated its signing capabilities to B.

The concept of proxy signatures was introduced in [51]. The first formal analysis of the security of standard proxy signatures was done in [12], where it was shown that a secure proxy signature scheme can be constructed from any secure digital signature scheme (and therefore, in particular, from any one-way function). The first identity-based proxy signature (IB-PS) schemes appeared in [64], but they lacked of a formal security analysis, since the first formal security model for IB-PS (which was adapted from the one in [12]) came later, in [62].

A proxy signature scheme $s_{\mathcal{PS}} = (\mathsf{kg}_{\mathcal{PS}}, \mathsf{sign}_{\mathcal{PS}}, \mathsf{vfy}_{\mathcal{PS}}, \mathsf{deleg}_{\mathcal{PS}}, \mathsf{p.kg}_{\mathcal{PS}}, \mathsf{p.sign}_{\mathcal{PS}}, \mathsf{p.vfy}_{\mathcal{PS}})$ has four additional algorithms, when compared with a standard signature scheme s: PS delegation, $\mathsf{deleg}_{\mathcal{PS}}$, is run by the original signer A, taking as input his secret key sk_A, a warrant ω indicating the general terms of the delegation, and the public key (or identity) of the proxy signer B; PS proxy key generation, $\mathsf{p.kg}_{\mathcal{PS}}$, is run by the proxy signer, taking as input his secret key sk_B and the output of the delegation algorithm $\mathsf{deleg}_{\mathcal{PS}}$, and obtaining a proxy secret key psk as output; the proxy signing protocol, $sig_{\mathcal{PS}} \stackrel{\$}{\leftarrow} \mathsf{p.sign}_{\mathcal{PS}}(m, psk)$, executed by the proxy signer, and the proxy verification protocol, 1 or 0 $\stackrel{\$}{\leftarrow} \mathsf{p.vfy}_{\mathcal{PS}}(m, sig_{\mathcal{PS}}, pk_A, pk_B, \omega)$, complete the picture.

Our generic construction of an identity-based proxy signature (IB-PS) scheme $\mathcal{S}_{\mathcal{PS}} = (\mathsf{KG}_{\mathcal{PS}}, \mathsf{EXT}_{\mathcal{PS}}, \mathsf{SIGN}_{\mathcal{PS}}, \mathsf{VFY}_{\mathcal{PS}}, \mathsf{DELEG}_{\mathcal{PS}}, \mathsf{P.KG}_{\mathcal{PS}}, \mathsf{P.SIGN}_{\mathcal{PS}}, \mathsf{P.VFY}_{\mathcal{PS}})$ from any standard PS scheme works in general, provided the public key of the proxy signer B is not strictly needed in the delegation phase of the underlying standard PS scheme. This is usually the case, because the public key is only used as an identifier of the proxy, and hence pk_B can be replaced with id_B. The first four algorithms are the same as in Figure 1 (note that protocol $\mathsf{EXT}_{\mathcal{PS}}$ is run for both the original id_A and the proxy id_B signers). The final identity-based proxy signature will include the proxy signature resulting from PS, along with the certificates on the messages $pk_A \,\|\, id_A$ and $pk_B \,\|\, id_B$, signed by the master entity. Details can be found in Figure 5.

Summing up, we obtain the following result.

Theorem 4.5 If s is a secure standard signature scheme and $s_{\mathcal{PS}}$ is a secure proxy signature scheme then the generic construction gives a secure identity-based proxy signature scheme $\mathcal{S}_{\mathcal{PS}}$.

All the previous proposals of IB-PS schemes employ bilinear pairings, and their security is proved in the random oracle model. With our generic construction, applied to the schemes in [12], we can easily obtain IB-PS schemes which do not employ bilinear pairings and whose security is proved in the standard model, as well as IB-PS schemes based on any one-way function.

Algorithm $\mathsf{DELEG}_{\mathcal{PS}}(id_A, USK[id_A], \omega, id_B)$
  Parse $(cert_A, pk_A, sk_A) \leftarrow USK[id_A]$
  $del_\omega \stackrel{\$}{\leftarrow} \mathsf{deleg}_{\mathcal{PS}}(sk_A, \omega, id_B)$
  Return $DEL = (id_A, cert_A, pk_A, \omega, del_\omega)$

Algorithm $\mathsf{P.KG}_{\mathcal{PS}}(USK[id_B], DEL)$
  Parse $(id_A, cert_A, pk_A, \omega, del_\omega) \leftarrow DEL$
  If $\mathsf{vfy}(PK, id_A \,\|\, pk_A, cert_A) = 0$ then fail
  Parse $(cert_B, pk_B, sk_B) \leftarrow USK[id_B]$
  $psk \stackrel{\$}{\leftarrow} \mathsf{p.kg}_{\mathcal{PS}}(sk_B, del_\omega)$
  Return $PSK = (\omega, psk, cert_B, pk_B, cert_A, pk_A)$

Algorithm $\mathsf{P.SIGN}_{\mathcal{PS}}(m, PSK)$
  Parse $(\omega, psk, cert_B, pk_B, cert_A, pk_A) \leftarrow PSK$
  If m does not satisfy ω then fail
  $sig_{\mathcal{PS}} \stackrel{\$}{\leftarrow} \mathsf{p.sign}_{\mathcal{PS}}(m, psk)$
  Return $SIG_{\mathcal{PS}} = (cert_B, pk_B, cert_A, pk_A, sig_{\mathcal{PS}})$

Algorithm $\mathsf{P.VFY}_{\mathcal{PS}}(PK, id_A, id_B, \omega, m, SIG_{\mathcal{PS}})$
  Parse $(cert_B, pk_B, cert_A, pk_A, sig_{\mathcal{PS}}) \leftarrow SIG_{\mathcal{PS}}$
  If $\mathsf{vfy}(PK, id_A \,\|\, pk_A, cert_A) = 0$ then return 0
  If $\mathsf{vfy}(PK, id_B \,\|\, pk_B, cert_B) = 0$ then return 0
  If $\mathsf{p.vfy}_{\mathcal{PS}}(m, sig_{\mathcal{PS}}, pk_A, pk_B, \omega) = 0$ then return 0
  Else return 1.

Figure 5: Generic construction of identity-based PS.

4.7 Online/Offline Signatures

In online/offline signatures the signing algorithm is split into two phases: the offline phase and the online phase. The idea is to shift the major computational overhead to the offline phase, which does not need as input(s) the message(s) to be signed in the future, whereas the online phase requires only a very low computational overhead. Online/offline signatures were introduced in [31], were the authors presented a general method for converting any signature scheme into an online/offline signature scheme. This method was later improved, in [58].

The generic construction of identity-based signatures can be directly applied to the case of online/offline signatures, by splitting the corresponding signing protocol into two phases. We therefore omit the explicit description of the protocols in this case.

Theorem 4.6 If $\mathcal{s}$ is a secure standard signature scheme and $\mathcal{s}_{OO}$ is a secure online/offline signature scheme then the generic construction gives a secure online/offline signature scheme $\mathcal{S}_{OO}$.

We are only aware of one identity-based online/offline signature scheme [63] in the literature, which is secure in the random oracle model and uses bilinear pairings. Applying the known generic construction [31, 58] to our construction results in identity-based online/offline signature schemes based on one-way functions. A more efficient scheme can be obtained by using, e.g., the pairing-based online/offline signature scheme from [15].

4.8 Threshold Signatures

Threshold signatures (TS) are used whenever the ability to sign must be decentralized. The idea is to share the signing power among a number of different players, in such a way that signing is possible only when a large enough number of honest players cooperate together.

A threshold signature scheme $\mathcal{s}_{\mathcal{TS}} = (\mathsf{tkg}_{\mathcal{TS}}, \mathsf{part.sign}_{\mathcal{TS}}, \mathsf{comb}_{\mathcal{TS}}, \mathsf{vfy}_{\mathcal{TS}})$ has four algorithms: the threshold key generation algorithm, $\mathsf{tkg}_{\mathcal{TS}}$, takes as input a set of players $\mathcal{P}$ and a threshold t, and outputs a single public key pk and a secret share sk_j for each player $P_j \in \mathcal{P}$. To jointly sign a message m, each player P_j in some subset $A \subset \mathcal{P}$ executes the protocol $sig^{(j)}_{\mathcal{TS}} \stackrel{\$}{\leftarrow} \mathsf{part.sign}_{\mathcal{TS}}(m, sk_j)$, obtaining a partial signature $sig^{(j)}_{\mathcal{TS}}$. After that, if $|A| \geq t$, the combining algorithm $sig_{\mathcal{TS}} \stackrel{\$}{\leftarrow} \mathsf{comb}_{\mathcal{TS}}(\{sig^{(j)}_{\mathcal{TS}}\}_{P_j \in A})$ can be executed to obtain a standard signature $sig_{\mathcal{TS}}$ which can finally be verified with the standard verification algorithm

Algorithm $\mathsf{TKG}_{\mathcal{TS}}(\mathcal{P}_{id}, t, SK)$
  $(pk, \{sk_j\}_{P_j \in \mathcal{P}_{id}}) \xleftarrow{\$} \mathsf{tkg}_{\mathcal{TS}}(\mathcal{P}_{id}, t)$
  $cert \xleftarrow{\$} \mathsf{sign}(SK, id \,\|\, pk)$
  For each $P_j \in \mathcal{P}_{id}$:
    Return $USK[id]_j \leftarrow (cert, pk, sk_j)$

Algorithm $\mathsf{PART.SIGN}_{\mathcal{TS}}(m, USK[id]_j)$
  Parse $(cert, pk, sk_j) \leftarrow USK[id]_j$
  $sig_{\mathcal{TS}}^{(j)} \xleftarrow{\$} \mathsf{part.sign}_{\mathcal{TS}}(m, sk_j)$
  Return $SIG_{\mathcal{TS}}^{(j)} = (cert, pk, sig_{\mathcal{TS}}^{(j)})$

Algorithm $\mathsf{COMB}_{\mathcal{TS}}(\{SIG_{\mathcal{TS}}^{(j)}\}_{P_j \in A})$
  If $|A| < t$ then fail
  Parse $(cert, pk, sig_{\mathcal{TS}}^{(j)}) \leftarrow SIG_{\mathcal{TS}}^{(j)}$, for each $P_j \in A$
  $sig_{\mathcal{TS}} \xleftarrow{\$} \mathsf{comb}_{\mathcal{TS}}(\{sig_{\mathcal{TS}}^{(j)}\}_{P_j \in A})$
  Return $SIG_{\mathcal{TS}} = (cert, pk, sig_{\mathcal{TS}})$

Figure 6: Generic construction of identity-based TS.

1 or 0 $\xleftarrow{\$}$ $\mathsf{vfy}_{\mathcal{TS}}(m, sig_{\mathcal{TS}}, pk)$. A non-interactive threshold signature scheme in the standard model and without pairings has been recently proposed in [27]. Here non-interactivity means that each player can individually compute his partial signature, without interacting with the other players.

Identity-based threshold signatures (IB-TS) were introduced in [6], to be used in a context where the signing key $USK[id]$ is shared by a collective $\mathcal{P}_{id}$ of signers with a common identity id (for example, the name of a company). These users will hold secret shares $USK[id]_j$ of the signing key, and they will be able to use them to compute partial signatures $SIG_{\mathcal{TS}}^{(j)}$ of a message. A sufficiently large fraction of such correct partial signatures can be combined to obtain a full signature $SIG_{\mathcal{TS}}$. More IB-TS schemes have been proposed in [25].

Here we show how to use our generic construction to obtain an identity-based threshold signature (IB-TS) scheme $\mathcal{S}_{\mathcal{TS}} = (\mathsf{KG}_{\mathcal{TS}}, \mathsf{TKG}_{\mathcal{TS}}, \mathsf{PART.SIGN}_{\mathcal{TS}}, \mathsf{COMB}_{\mathcal{TS}}, \mathsf{VFY}_{\mathcal{TS}})$ starting from any standard signature scheme $s = (\mathsf{kg}, \mathsf{sign}, \mathsf{vfy})$ and a standard threshold signature scheme $s_{\mathcal{TS}}$ as described above. The protocols $\mathsf{KG}_{\mathcal{TS}}$ and $\mathsf{VFY}_{\mathcal{TS}}$ work exactly as in Figure 1. After executing $\mathsf{KG}_{\mathcal{TS}}$, the master entity holds a pair of keys (SK, PK). The rest of protocols of $\mathcal{S}_{\mathcal{TS}}$ are described in Figure 6.

Note that, if the signing phase of the standard threshold signature scheme $\mathcal{TS}$ is non-interactive, we obtain a non-interactive identity-based threshold signature scheme (comparable to that in [6]). Furthermore, if scheme $\mathcal{TS}$ has mechanisms to achieve robustness (i.e. to detect incorrect partial signatures coming from dishonest players), then such mechanisms can be easily included in our construction of the scheme $\mathcal{S}_{\mathcal{TS}}$.

Theorem 4.7 If s is a secure standard signature scheme and $s_{\mathcal{P}}$ is a secure threshold signature scheme then the generic construction gives a secure identity-based threshold signature scheme $\mathcal{S}_{\mathcal{P}}$.

As a consequence of this theorem and the work [27], fairly efficient IB-TS schemes can be obtained from RSA or discrete-log based signatures, without resorting to random oracles. A generic construction of (non-efficient) threshold signature schemes from standard signature schemes can be obtained by slightly modifying the generic construction of threshold encryption schemes from standard public key encryption schemes proposed in [29]. This fact, together with our theorem, shows that IB-TS schemes can be built out of any one-way function.

4.9 Aggregate Signatures

The idea of an aggregate signature (AS) scheme is to combine n signatures on n different messages, signed by n (possibly different) signers, in order to obtain a single aggregate signature which provides the same certainty than the n initial signatures. Besides the three algorithms that form a standard signature scheme, an aggregate signature scheme $\mathcal{s}_{\mathcal{AS}} = (\mathsf{kg}_{\mathcal{AS}}, \mathsf{sign}_{\mathcal{AS}}, \mathsf{vfy}_{\mathcal{AS}}, \mathsf{aggreg}_{\mathcal{AS}}, \mathsf{agvfy}_{\mathcal{AS}})$ contains two additional protocols: the aggregation protocol, $\mathsf{aggreg}_{\mathcal{AS}}$, takes as input n tuples $\{(pk_i, m_i, sig_i)\}_{1\leq i\leq n}$ and outputs an aggregate signature $sig_{\mathcal{AS}}$; the aggregate verification protocol, $\mathsf{agvfy}_{\mathcal{AS}}$, takes as input an aggregate signature $sig_{\mathcal{AS}}$, along with n pairs $\{(pk_i, m_i)\}_{1\leq i\leq n}$ and outputs 1 if the aggregate signature is valid, 0 otherwise.

The main goal in the design of such protocols is that the length of $sig_{\mathcal{AS}}$ is constant, independent of the number of messages and signers. Of course, to check correctness of an aggregate signature, the verifier will also need the messages m_i and the public keys pk_i, but this is not taken into account when considering the length of $sig_{\mathcal{AS}}$.

The idea of aggregate signatures was introduced in [17], where a scheme with constant-length aggregate signatures was presented and analyzed, based on the signature scheme of [18]. In the identity-based framework, the only proposal which achieves constant-length aggregation is that of [34]; however, this scheme only works in a more restrictive scenario where some interaction or sequentiality is needed among the signers of the messages which will be aggregated later (in the same direction as [50, 49], for the PKI-based scenario). With respect to strict aggregate signatures (without any kind of interaction among the signers) in the identity-based setting, the most efficient proposal is that in [38], which does not achieve constant-length aggregation: the length of the aggregate signature does not depend on the number of signed messages, but on the number of different signers.

We can achieve exactly the same level of partial aggregation by using our generic construction. We start from any standard AS scheme $\mathcal{s}_{\mathcal{AS}}$ (assumed to produce constant-length aggregate signatures) and obtain an identity-based aggregate signature scheme $\mathcal{S}_{\mathcal{AS}} = (\mathsf{KG}_{\mathcal{AS}}, \mathsf{EXT}_{\mathcal{AS}}, \mathsf{SIGN}_{\mathcal{AS}}, \mathsf{VFY}_{\mathcal{AS}}, \mathsf{AGGREG}_{\mathcal{AS}}, \mathsf{AG.VFY}_{\mathcal{AS}})$. Again, the first four algorithms are the same as in Figure 1, replacing $\mathcal{s}'$ with the protocols $\mathsf{kg}_{\mathcal{AS}}, \mathsf{sign}_{\mathcal{AS}}, \mathsf{vfy}_{\mathcal{AS}}$ of the inherent AS scheme $\mathcal{s}_{\mathcal{AS}}$. In an identity-based aggregate signature $SIG_{\mathcal{AS}}$, besides the standard aggregate signature $sig_{\mathcal{AS}}$, we will have to include the pairs $(cert_i, pk_i)$ for all the signers id_i. If the standard signature scheme $\mathcal{s}$, which is used to generate the signatures $cert_i$, admits constant-length aggregation (for example, if $\mathcal{s} = \mathcal{s}_{\mathcal{AS}}$), then all the $cert_i$ can be aggregated into a single $cert$. But the public keys cannot be aggregated; this is why the length of the identity-based aggregate signature will depend on the number of different signers (and not on the number of signed messages), as it happens in the scheme of [38]. Details of the generic construction of identity-based aggregate signatures can be found in Figure 7.

Summing up, we obtain the following result.

Theorem 4.8 If $\mathcal{s}$ is a secure standard signature scheme and $\mathcal{s}_{\mathcal{AS}}$ is a secure aggregate signature scheme then the generic construction gives a secure identity-based aggregate signature scheme $\mathcal{S}_{\mathcal{AS}}$, where the length of an aggregate signature depends on the number of different signers.

There is only a PKI-based aggregate signature scheme which requires no interaction at all among the signers and produces constant-length signatures, the one in [17], which employs bilinear pairings and is proved secure in the random oracle model. For this reason, our generic construction of IB-AS schemes does not bring new results to Table 1; we achieve the same level as the scheme in [38]. However, any future progress in the area of aggregate signatures for the PKI-based scenario would directly imply new results on IB-AS, via our generic transformation. On the other hand, the generic transformation cannot lead to a solution which achieves completely

Algorithm $\mathsf{AGGREG}_{\mathcal{AS}}(\{(id_i, m_i, SIG_i)\}_{1\le i\le n})$
Parse $(cert_i, pk_i, sig_i) \leftarrow SIG_i$, for each i
If $\mathsf{vfy}(PK, id_i \,\|\, pk_i, cert_i) = 0$, for some i, then fail
$sig_{\mathcal{AS}} \stackrel{\$}{\leftarrow} \mathsf{aggreg}_{\mathcal{AS}}(\{(pk_i, m_i, sig_i)\}_{1\le i\le n})$
$cert \stackrel{\$}{\leftarrow} \mathsf{aggreg}_{\mathcal{AS}}(\{(PK, id_i \,\|\, pk_i, cert_i)\}_{1\le i\le n})$
Return $SIG_{\mathcal{AS}} = (sig_{\mathcal{AS}}, cert, \{pk_i\}_{1\le i\le n})$

Algorithm $\mathsf{AG.VFY}_{\mathcal{AS}}(SIG_{\mathcal{AS}}, \{(id_i, m_i)\}_{1\le i\le n})$
Parse $(sig_{\mathcal{AS}}, cert, \{pk_i\}_{1\le i\le n}) \leftarrow SIG_{\mathcal{AS}}$
If $\mathsf{agvfy}_{\mathcal{AS}}(cert, \{(PK, id_i \,\|\, pk_i)\}_{1\le i\le n}) = 0$ then return 0
If $\mathsf{agvfy}_{\mathcal{AS}}(sig_{\mathcal{AS}}, \{(pk_i, m_i)\}_{1\le i\le n}) = 0$ then return 0
Else return 1.

Figure 7: Generic construction of identity-based AS.

constant-length signatures: this length will always depend on the number of different signers. Therefore, an optimal solution to the problem of identity-based aggregate signatures will have to be based on a different approach.

4.10 Limitations and Extensions

Our generic approach to construct identity-based signature schemes with special properties does not work in situations where the signing procedure (in the corresponding PKI-based scheme) involves other public keys than the one from the signer, and interaction between the signer and the owners of these public keys is not mandatory. Our approach fails in this case because in the identity-based framework the signer only knows the identity of the other users, and needs some interaction with them in order to know the public key that they have received in the key extraction phase.

Some examples of signature schemes with special properties falling inside this group are: ring signatures [55, 64]; designated verifier signatures [40, 59]; confirmer signatures [22]; chameleon signatures [45, 5]; and nominative signatures [60].

We are aware of the fact that the list of properties where the generic approach can be applied is not complete and it obviously can also be applied to other concepts (like one-time signatures [47], strongly secure signatures, homomorphic signatures [41], etc.) as well. Furthermore, combinations of different additional properties are possible. For example, it is possible to give a generic construction of identity-based threshold undeniable signatures based on the existence of threshold undeniable signatures. We also note that our generic construction can be extended to the case of hierarchical identity-based signatures (HIBS), by using certificate-chains [43].

5 Generic Construction of Identity-Based Blind Signatures

In this section we consider in more detail the generic construction in the case of blind signature schemes. We first recall the basic definitions of PKI-based and identity-based blind signature schemes, then we explain and analyze our construction.

5.1 Blind Signature Schemes

Blind signature schemes were introduced in [21] with electronic banking as first motivation. The intuitive idea is that a user asks some signer to blindly sign a (secret) message m. At the end of the process, the user obtains a valid signature on m from the signer, but the signer has no information about the message he has signed. More formally, a blind signature scheme $s_{\mathcal{BS}} = (\mathsf{kg}_{\mathcal{BS}}, \mathsf{sign}_{\mathcal{BS}}, \mathsf{vfy}_{\mathcal{BS}})$ consists of the following (partially interactive) PPT algorithms.

The key generation algorithm $\mathsf{kg}_{\mathcal{BS}}$ generates, on input 1^k, a key pair (sk, pk). The blind signing algorithm $\mathsf{sign}_{\mathcal{BS}}$ is an interactive protocol between a user U and a signer S with public key pk. The input for the user is $Inp_U = (m, pk)$, where m is the message he wants to be signed by the signer. The input Inp_S of the signer is his secret key sk. In the end, the output Out_S of the signer is 'completed' or 'not completed', whereas the output Out_U of the user is either 'fail' or a blind signature sig. We use notation $(Out_U, Out_S) \xleftarrow{\$} \mathsf{sign}_{\mathcal{BS}}(Inp_U, Inp_S)$ to refer to one execution of this interactive protocol. Finally, the verification algorithm $\mathsf{vfy}_{\mathcal{BS}}$ is the same as for standard signatures.

Blindness. Intuitively, the blindness property captures the notion that a signer cannot obtain any information about the messages he is signing for some user. Formally, this notion is defined by the following game that an adversary (signer) B plays against a challenger (who plays the role of a user).

First the adversary B runs the key generation protocol $(sk, pk) \xleftarrow{\$} \mathsf{kg}_{\mathcal{BS}}(1^k)$. Then the adversary B chooses two messages m_0 and m_1 and sends them to the challenger, along with the public key pk. The challenger chooses at random a bit $b \in \{0, 1\}$ and then the interactive signing protocol is executed two times (possibly in a concurrent way), resulting in $(Out_{U,b}, Out_{S,b}) \xleftarrow{\$} \mathsf{sign}_{\mathcal{BS}}(Inp_{U,b}, Inp_{S,b})$ and $(Out_{U,1-b}, Out_{S,1-b}) \xleftarrow{\$} \mathsf{sign}_{\mathcal{BS}}(Inp_{U,1-b}, Inp_{S,1-b})$, where adversary B plays the role of the signer S, and the challenger plays the role of the user, with inputs $Inp_{U,b} = (pk, m_b)$ and $Inp_{U,1-b} = (pk, m_{1-b})$. Finally, the adversary B outputs its guess b'. Note that the adversary in the above security game is in the possession of the secret key sk.

We say that such an adversary B succeeds if $b' = b$ and define its advantage in the above game as $\mathbf{Adv}^{\text{blind}}_{s_{\mathcal{BS}},\mathsf{B}}(k) = |\Pr[\,b' = b\,] - 1/2|$. A scheme $s_{\mathcal{BS}}$ has the blindness property if, for all PPT adversaries B, $\mathbf{Adv}^{\text{blind}}_{s_{\mathcal{BS}},\mathsf{B}}(k)$ is a negligible function (with respect to the security parameter k). If $\mathbf{Adv}^{\text{blind}}_{s_{\mathcal{BS}},\mathsf{B}}(k) = 0$, for any (possibly computationally unbounded) adversary B, then the blindness of the scheme is unconditional.

Unforgeability. Unforgeability captures the intuitive requirement that a user obtains a valid signature from the signer only if they complete together an execution of the blind signature protocol. Among the different (but equivalent) formal definitions of unforgeability for blind signature schemes (see, e.g., [42, 54]), we consider the one from [42], which is given by the following game that an adversary F (user or forger) plays against a challenger (signer).

First the challenger runs the key generation protocol $(pk, sk) \xleftarrow{\$} \mathsf{kg}_{\mathcal{BS}}(1^k)$ and gives pk to F, whereas the secret key sk is kept secret by the challenger. During its execution the forger F adaptively chooses messages m_j, then the interactive signing protocol $(Out_U, Out_S) \xleftarrow{\$} \mathsf{sign}_{\mathcal{BS}}(Inp_U, Inp_S)$ is executed (possibly in a concurrent way), where the adversary F plays the role of the user U, with input $Inp_U = (pk, m_j)$, and the challenger plays the role of the signer, with input the secret key sk. Let ℓ be the number of such queries that finish with $Out_S =$'completed'. Eventually, adversary F outputs a list of ℓ' tuples $\{(m_i, sig_i)\}_{1 \le i \le \ell'}$. We say that F *succeeds* if

- $\ell < \ell'$

- $1 \leftarrow \mathsf{vfy}_{\mathcal{BS}}(pk, m_i, sig_i)$, for all $i = 1, \ldots, \ell'$.

We say that such an adversary F is an (ℓ, ℓ')-forger and define its advantage as $\mathbf{Adv}^{\text{forge}}_{s_{\mathcal{BS}},\mathsf{F}}(k) = \Pr[\,\mathsf{F} \text{ succeeds}\,]$. The scheme $s_{\mathcal{BS}}$ is unforgeable if $\mathbf{Adv}^{\text{forge}}_{s_{\mathcal{BS}},\mathsf{F}}(k)$ is a negligible function in k for all PPT (ℓ, ℓ')-forger F.

5.2 Identity-Based Blind Signature Schemes

Analogously, an identity-based blind signature scheme is defined as a tuple of PPT algorithms $\mathcal{S}_{\mathcal{BS}} = (\mathsf{KG}_{\mathcal{BS}}, \mathsf{EXT}_{\mathcal{BS}}, \mathsf{SIGN}_{\mathcal{BS}}, \mathsf{VFY}_{\mathcal{BS}})$. The first three may be randomized but the last is not. The key generation algorithm $\mathsf{KG}_{\mathcal{BS}}$ generates, on input 1^k, a master key pair (PK, SK). The key extraction algorithm $\mathsf{EXT}_{\mathcal{BS}}$ takes as inputs PK and an identity $id \in \{0,1\}^*$, and returns a secret key $USK[id]$ for the user with this identity. The blind signing algorithm $\mathsf{SIGN}_{\mathcal{BS}}$ is an interactive protocol between a user U and a signer with identity id. The common input for them is PK. The input for the user is $Inp_U = (id, m)$ where m is the message he wants to be signed by id. The input Inp_{id} of the signer is his secret key $USK[id]$. In the end, the output Out_{id} of the signer is 'completed' or 'not completed', whereas the output Out_U of the user is either 'fail' or a signature SIG. We use notation $(Out_U, Out_{id}) \stackrel{\$}{\leftarrow} \mathsf{SIGN}_{\mathcal{BS}}(PK, Inp_U, Inp_{id})$ to refer to one execution of this interactive protocol. Finally, the verification algorithm $\mathsf{VFY}_{\mathcal{BS}}$ takes as input PK, a message m, an identity id and a blind signature SIG; it outputs 1 if the signature is valid with respect to the master public key PK and the identity id, and 0 otherwise.

An identity-based blind signature scheme must satisfy the requirements of correctness, blindness and unforgeability, that we now explain in detail.

Correctness. For any execution of the setup protocol $(SK, PK) \stackrel{\$}{\leftarrow} \mathsf{KG}_{\mathcal{BS}}(1^k)$, the key extraction protocol $USK[id] \stackrel{\$}{\leftarrow} \mathsf{EXT}_{\mathcal{BS}}(SK, id)$, and the interactive signing protocol $(Out_U, Out_{id}) \stackrel{\$}{\leftarrow} \mathsf{SIGN}_{\mathcal{BS}}(PK, Inp_U, Inp_{id})$, where $Inp_U = (id, m)$ and $Inp_{id} = USK[id]$, the following property must be satisfied:

$$Out_{id} = \text{'completed'} \implies \Big(1 \leftarrow \mathsf{VFY}_{\mathcal{BS}}(PK, id, m, Out_U) \Big).$$

Blindness. Blindness of an identity-based blind signature scheme is defined by a game played between a challenger and an adversary. This adversary B models the dishonest behavior of a signer who tries to distinguish which message (between two messages chosen by himself) is being signed in an interactive execution of the signing protocol with a user. The game is as follows.

First the challenger runs the setup protocol $(SK, PK) \stackrel{\$}{\leftarrow} \mathsf{KG}_{\mathcal{BS}}(1^k)$ and gives PK to B. The master secret key SK is kept secret by the challenger. The adversary B is allowed to query for secret keys of identities id of its choice. The challenger runs $USK[id] \leftarrow \mathsf{EXT}_{\mathcal{BS}}(SK, id)$ and gives the resulting secret key $USK[id]$ to B. If the same identity is asked again, the same value $USK[id]$ must be returned by the challenger. At some point, the adversary B chooses an identity id^* and two messages m_0, m_1, and sends these values to the challenger. The challenger chooses at random one bit $b \in \{0,1\}$ and then the interactive signing protocol is executed twice (possibly in a concurrent way), resulting in $(Out_{U,b}, Out_{id^*,b}) \stackrel{\$}{\leftarrow} \mathsf{SIGN}_{\mathcal{BS}}(Inp_{U,b}, Inp_{id^*,b})$ and $(Out_{U,1-b}, Out_{id^*,1-b}) \stackrel{\$}{\leftarrow} \mathsf{SIGN}_{\mathcal{BS}}(Inp_{U,1-b}, Inp_{id^*,1-b})$, where adversary B plays the role of the signer id^*, with input $Inp_{id^*,b} = Inp_{id^*,1-b} = USK[id^*]$, and the challenger plays the role of the user, with inputs $Inp_{U,b} = (m_b, id^*)$ and $Inp_{U,1-b} = (m_{1-b}, id^*)$. Finally, the adversary B outputs its guess b'.

We say that such an adversary B succeeds if $b' = b$ and define its advantage in the above game as $\mathbf{Adv}^{\text{ib-blind}}_{\mathcal{S}_{\mathcal{BS}},\mathsf{B}}(k) = |\Pr[\,b' = b\,] - 1/2|$. A scheme $\mathcal{S}_{\mathcal{BS}}$ has the blindness property if, for all PPT adversaries B, $\mathbf{Adv}^{\text{ib-blind}}_{\mathcal{S}_{\mathcal{BS}},\mathsf{B}}(k)$ is a negligible function (with respect to the security parameter k). If $\mathbf{Adv}^{\text{ib-blind}}_{\mathcal{S}_{\mathcal{BS}},\mathsf{B}}(k) = 0$, for any (possibly computationally unbounded) adversary B, then the blindness of the scheme is unconditional.

Unforgeability. Our definition of unforgeability for identity-based blind signatures is adapted from the concept of (ℓ, ℓ')-unforgeability introduced in [42] for standard PKI-based blind signatures. A forger F_{IB} against the unforgeability property of an identity-based blind signature scheme is defined by means of the following game that it plays against a challenger.

First of all, the challenger runs the setup protocol $(SK, PK) \leftarrow \mathsf{KG}_{\mathcal{BS}}(1^k)$ and gives PK to F. The master secret key SK is kept secret by the challenger. Then the forger F can make two kinds of queries to the challenger. On the one hand, F can ask for the secret key of an identity id of its choice; the challenger runs $USK[id] \stackrel{\$}{\leftarrow} \mathsf{EXT}_{\mathcal{BS}}(SK, id)$ and returns the resulting user secret key $USK[id_i]$ to F. If an identity id is asked twice, the challenger must return the same secret key $USK[id]$. On the other hand, the forger F can ask for the execution of the blind signing protocol: F chooses pairs (id_j, m_j), then the challenger first runs $USK[id_j] \leftarrow \mathsf{EXT}_{\mathcal{BS}}(SK, id_j)$ to get the secret key $USK[id_j]$ for this identity. After that, the interactive signing protocol $(Out_U, Out_{id}) \leftarrow \mathsf{SIGN}_{\mathcal{BS}}(PK, Inp_U, Inp_{id})$ is executed (possibly in a concurrent way), where the adversary F plays the role of the user U, with input $Inp_U = (id_j, m_j)$, and the challenger plays the role of the signer id_j, with input the secret key $USK[id_j]$. Let ℓ be the number of such queries that finish with $Out_{id_j} =$'completed'. Eventually, the adversary F outputs a list of ℓ' tuples $\{(id_i, m_i, SIG_i)\}_{1 \le i \le \ell'}$. We say that F *succeeds* if:

- $\ell < \ell'$;
- $1 \leftarrow \mathsf{VFY}_{\mathcal{BS}}(PK, id_i, m_i, SIG_i)$, for all $i = 1, \ldots, \ell'$;
- the pairs (id_i, m_i) included in the output list are pairwise different; and
- F did not ask a secret key query for any of the identities id_i in the output list.

We say that such an adversary F is an (ℓ, ℓ')-forger and define its advantage as $\mathbf{Adv}^{\text{ib-forge}}_{\mathcal{S}_{\mathcal{BS}},\mathsf{F}}(k) = \Pr[\,\mathsf{F} \text{ succeeds}\,]$. The scheme $\mathcal{S}_{\mathcal{BS}}$ is unforgeable if $\mathbf{Adv}^{\text{ib-forge}}_{\mathcal{S}_{\mathcal{BS}},\mathsf{F}}$ is a negligible function in k for any PPT (ℓ, ℓ')-forger F.

5.3 Construction

Let $s = (\mathsf{kg}, \mathsf{sign}, \mathsf{vfy})$ be a standard signature scheme and let $s_{\mathcal{BS}} = (\mathsf{kg}_{\mathcal{BS}}, \mathsf{sign}_{\mathcal{BS}}, \mathsf{vfy}_{\mathcal{BS}})$ be a blind signature scheme. We construct an identity-based blind signature scheme $\mathcal{S}_{\mathcal{BS}} = (\mathsf{KG}_{\mathcal{BS}}, \mathsf{SIGN}_{\mathcal{BS}}, \mathsf{EXT}_{\mathcal{BS}}, \mathsf{VFY}_{\mathcal{BS}})$ as follows.

The description of the algorithms $\mathsf{KG}_{\mathcal{BS}}$ and $\mathsf{VFY}_{\mathcal{BS}}$ is the same as in the generic construction of identity-based signatures from Figure 1. Recall that the master key pair output by $\mathsf{KG}_{\mathcal{BS}}$ is a key pair (PK, SK) of the standard signature scheme s obtained by running kg. The description of algorithm $\mathsf{EXT}_{\mathcal{BS}}$ is also the same as in Figure 1 with the difference that it makes sure that only one $USK[id]$ is established for each identity id. This can be done, for example, by storing all computed $USK[id]$ in a table. See also Remark 5.2.

The interactive blind signing protocol $\mathsf{SIGN}_{\mathcal{BS}}$ between a user U and a signer with identity id consists of the following steps. Recall that PK is a common input for user and signer, the input of the user is (id, m) and the input of the signer is $USK[id] = (cert, pk, sk)$.

1. User U sends the query $(id, \text{'blindsignature?'})$ to the signer.

2. If the signer does not want to sign, the protocol finishes with $Out_U =$'fail' and $Out_{id} =$'not completed'. Otherwise, the signer sends $(cert, pk)$ back to the user.

3. The user first verifies the certificate on pk by running $\{0,1\} \leftarrow \mathsf{vfy}(PK, id||pk, cert)$. If the output is 0, then the protocol finishes with $Out_U =$'fail' and $Out_{id} =$'not completed'.

   Otherwise, user and signer interact to run the blind signature protocol of $\mathcal{s}_{\mathcal{BS}}$, resulting in $(Out'_U, Out'_{id}) \stackrel{\$}{\leftarrow} \mathsf{sign}_{\mathcal{BS}}(Inp_U, Inp_{id})$, where $Inp_U = (pk, m)$ and $Inp_{id} = sk$. If $Out'_U \neq$'fail', then it consists of a standard blind signature sig on m under secret key sk. The final output for the user is in this case $Out_U = SIG = (cert, pk, sig)$, which is defined to be the identity-based blind signature on message m coming from identity id.

Remark 5.1 If, in an execution of the blind signing protocol $\mathsf{sign}_{\mathcal{BS}}$ of the scheme $\mathcal{s}_{\mathcal{BS}}$, the user does not need to know the public key of the signer when computing the information that he (the user) sends in the first round, then the two first steps of our generic signing protocol $\mathsf{SIGN}_{\mathcal{BS}}$ can be left out. In this case, the sent information and the necessary computations of these two steps can be moved to step 3 (i.e., injected into the execution of the PKI-based protocol $\mathsf{sign}_{\mathcal{BS}}$). In this way, the round complexity of the resulting scheme $\mathcal{S}_{\mathcal{BS}}$ would be exactly the same as in the underlying PKI-based scheme $\mathcal{s}_{\mathcal{BS}}$. This modification does not affect the security analysis that we explain in next section. An example of this fact will be illustrated in the concrete instantiation that we describe in Section 5.5, because Boldyreva's blind signature scheme [13] satisfies the required condition: the user does not need to know the public key of the signer to compute the first message of the interactive signing protocol.

Remark 5.2 In our generic construction, the master entity must store a list of pairs $(id, USK[id])$ to avoid that the same user obtains two certified (signed) but distinct public keys. This is necessary to ensure blindness of the scheme. This solution increases the key management cost for the master entity. To solve this drawback, we can use the following standard technique to make $\mathsf{EXT}_{\mathcal{BS}}$ deterministic. The master secret key SK additionally contains some randomness that is used as the seed of a pseudorandom generator to generate the random coins for $\mathsf{kg}_{\mathcal{BS}}$ when generating the key pairs (sk, pk).

5.4 Security Analysis

In this section we prove that the identity-based blind signature scheme $\mathcal{S}_{\mathcal{BS}}$ constructed in the previous section satisfies the three required security properties. It is very easy to check the correctness of the protocol. Let us prove in detail that blindness and unforgeability also hold, assuming that the schemes $\mathcal{s}$ and $\mathcal{s}_{\mathcal{BS}}$ employed as primitives are secure.

Theorem 5.3 Assume the signature scheme $\mathcal{s}$ is *strongly* unforgeable and the blind signature scheme $\mathcal{s}_{\mathcal{BS}}$ is blind. Then the identity-based blind signature scheme $\mathcal{S}_{\mathcal{BS}}$ constructed in Section 5.3 is blind.

Proof: To prove this result, we show that if there exists a successful adversary B_{IB} against the blindness of the scheme $\mathcal{S}_{\mathcal{BS}}$, then there exists either a successful forger F against the signature scheme $\mathcal{s}$ or a successful adversary B against the blindness of the blind signature scheme $\mathcal{s}_{\mathcal{BS}}$. In particular we show that

$$\mathbf{Adv}^{\text{ib-blind}}_{\mathcal{S}_{\mathcal{BS}},\mathsf{B}_{\mathrm{IB}}}(k) \leq \mathbf{Adv}^{\text{blind}}_{\mathcal{s}_{\mathcal{BS}},\mathsf{B}}(k) + \mathbf{Adv}^{\text{sforge}}_{\mathcal{s},\mathsf{F}}(k).$$

We now construct F and B.

Setup. Forger F receives as initial input some public key pk for the standard signature scheme $\mathcal{s}$. Then we initialize the adversary B_{IB} by providing it with $PK = pk$.

Secret key queries. Adversary B_{IB} is allowed to make secret key queries for identities id of its choice. To answer such a query, we run the key generation protocol of the blind signature scheme $\mathcal{s}_{\mathcal{BS}}$ to obtain $(sk, pk) \xleftarrow{\$} \mathsf{kg}_{\mathcal{BS}}(1^k)$. Then we send the query $id \,\|\, pk$ to the signing oracle associated to the forger F, and obtain as answer a valid signature $cert$, on message $id \,\|\, pk$, with respect to the scheme $\mathcal{s}$ and the master public key PK . Then we send to B_{IB} the consistent answer $USK[id] = (cert, pk, sk)$. We store all this information in some table. If the same identity is asked twice by B_{IB}, then the same secret key is given as answer.

Challenge. At some point, B_{IB} will output some challenge identity id_* and two messages m_0, m_1. Without loss of generality we can assume that B_{IB} had already asked for the secret key of this identity (otherwise, we generate it now and send it to B_{IB}), obtaining $USK[id_*] = (cert_*, pk_*, sk_*)$. Then we start constructing an adversary B against the blindness of the blind signature scheme $\mathcal{s}_{\mathcal{BS}}$, by sending public key pk_* and messages m_0, m_1 to the corresponding challenger.

Now we must execute twice the interactive blind signature protocol with B_{IB}, where B_{IB} acts as a signer and we act as the user. For both executions, we first send (id_*, 'blindsignature?') to B_{IB}. As answers, we will obtain $(cert_*^{(0)}, pk_*^{(0)})$ and $(cert_*^{(1)}, pk_*^{(1)})$ from B_{IB}, where $cert_*^{(j)}$ is a valid signature on $id_* \,\|\, pk_*^{(j)}$, for both $j = 0, 1$.

If $(cert_*^{(j)}, pk_*^{(j)}) \neq (cert_*, pk_*)$ for either $j = 0$ of $j = 1$, then F outputs $cert_*^{(j)}$ as a valid forgery on the message $id_*||pk_*^{(j)}$ for the signature scheme $\mathcal{s}$. This is a valid forgery against signature scheme $\mathcal{s}$, because these signatures were not obtained during the attack. Therefore, in this case we would have a successful forger F against $\mathcal{s}$, contradicting the hypothesis in the statement of the theorem which claims that $\mathcal{s}$ is strongly unforgeable.

From now on we assume that we have $(cert_*^{(j)}, pk_*^{(j)}) = (cert_*, pk_*)$ for both $j = 0, 1$, and therefore the two first steps in the two executions of the interactive signing protocol are identical. Then we run the two executions of the blind signing protocol of scheme $\mathcal{s}_{\mathcal{BS}}$, playing the role of the signer: we obtain from B_{IB} the information that we must send to the challenger (user) of $\mathcal{s}_{\mathcal{BS}}$, and this challenger sends back to us the information that we must provide to B_{IB}. This challenger of $\mathcal{s}_{\mathcal{BS}}$ is the one who chooses the bit $b \in \{0, 1\}$.

At the end, the adversary B_{IB} outputs its guess b'. B outputs the same bit b' as its guess in the blindness game against the blind signature scheme $\mathcal{s}_{\mathcal{BS}}$.

Since the two first steps in the two executions of the interactive signing protocol of $\mathcal{S}_{\mathcal{BS}}$ run between B_{IB} and us are identical, we have that distinguishing between the two executions of $\mathsf{SIGN}_{\mathcal{BS}}$ is equivalent to distinguishing between the two executions of $\mathsf{sign}_{\mathcal{BS}}$.

Summing up, if B_{IB} succeeds in breaking the blindness of $\mathsf{SIGN}_{\mathcal{BS}}$, then we can construct an algorithm B which breaks the blindness of $\mathsf{sign}_{\mathcal{BS}}$, with exactly the same success probability. ▮

We stress that the signature scheme $\mathcal{s}$ really has to be *strongly* unforgeable. Otherwise a signer could break blindness: he could use different versions of $USK[id]$ in different signing sessions and could later use this information to trace the user.

Theorem 5.4 Assume the standard signature scheme $\mathcal{s}$ is unforgeable and the blind signature scheme $\mathcal{s}_{\mathcal{BS}}$ is unforgeable. Then the identity-based blind signature scheme $\mathcal{S}_{\mathcal{BS}}$ from Section 5.3 is unforgeable.

Proof: The proof of Theorem 5.4 is similar to the one of Theorem 5.3. We prove that if there exists a successful adversary F_{IB} against the unforgeability of the scheme $\mathcal{S}_{\mathcal{BS}}$, then there exists either a successful forger F against the unforgeability of the signature scheme $\mathcal{s}$ or a successful adversary F' against the unforgeability of the blind signature scheme $\mathcal{s}_{\mathcal{BS}}$. In particular, we show that

$$\mathbf{Adv}^{\text{forge}}_{\mathcal{S}_{\mathcal{BS}},\mathsf{F}_{\text{IB}}}(k) \leq q \cdot \left(\mathbf{Adv}^{\text{forge}}_{\mathcal{s}_{\mathcal{BS}},\mathsf{F}'}(k) + \mathbf{Adv}^{\text{forge}}_{\mathcal{s},\mathsf{F}}(k)\right),$$

where q is an upper bound for the total number of different identities appearing in F_{IB}'s queries during the security experiment.

Let us assume F_{IB} is an (ℓ, ℓ')-forger for some value ℓ (polynomial in k) and let us construct from it F and F', where at least one of them is successful.

Setup. Forger F receives as initial input some public key pk for the signature scheme $\mathcal{s}$. Then we initialize adversary F' by providing it with $PK = pk$. The adversary F_{IB} is allowed to make two different kinds of queries, secret key queries for identities id_i and blind signature queries for pairs (id_j, m_j). First of all, we choose at random some integer $i_* \in \{1, 2, \ldots, q\}$ (recall that q is an upper bound for the total number of different identities appearing in F_{IB}'s queries). We also start constructing an adversary F' against the unforgeability of the blind signature scheme $\mathcal{s}_{\mathcal{BS}}$, receiving from the corresponding challenger some public key pk_*.

Queries. Each time a new identity id_i appears in some of the queries made by F_{IB}, where the indices refer to the order of appearance (id_1 is the first identity that appears in some F_{IB}'s query, and so on), we act as follows:

- If $i \neq i_*$, then we run the key generation protocol of the blind signature scheme $\mathcal{s}_{\mathcal{BS}}$ to obtain $(sk_i, pk_i) \stackrel{\$}{\leftarrow} \mathsf{kg}_{\mathcal{BS}}(1^k)$. Then we send the query $id_i \,\|\, pk_i$ to the signing oracle associated to the forger F, and we obtain as answer a valid signature $cert_i$ on $id_i \,\|\, pk_i$, with respect to the scheme $\mathcal{s}$ and the public key $PK = pk$.
- For i_*-th identity, we send the query $id_{i_*} \,\|\, pk_*$ to the signing oracle associated to the forger F, and we obtain as answer a valid signature $cert_{i_*}$.

Now we are ready to answer F_{IB}'s queries. If F_{IB} asks for the secret key of id_{i_*}, we abort. Otherwise, if F_{IB} asks for the secret key of id_i, with $i \neq i_*$, then we send back the correct secret key $USK[id_i] = (cert_i, pk_i, sk_i)$.

With respect to blind signature queries (id_j, m_j), if $id_j \neq id_{i_*}$, we can perfectly simulate a running of the blind signing protocol because we know the secret key $USK[id_j]$ for this signer. Otherwise, if $id_j = id_{i_*}$, then the first message $(id_{i_*}, \text{‘blindsignature?}')$ comes from the adversary (acting as a user). We answer by sending back to F_{IB} the values $(pk_*, cert_{i_*})$. For the rest of the protocol execution, we receive messages from F_{IB} and we forward them to the blind signing oracle associated with the adversary F' we are constructing. Since the challenge public key is pk_* (the public key for identity id_{i_*}) the answers that we receive from this oracle are consistent, and we can forward them to F_{IB}.

Let ℓ be the number of such blind signature queries that are successfully completed. With probability $\mathbf{Adv}^{\text{forge}}_{\mathcal{S}_{\mathcal{BS}},\mathsf{F}_{\text{IB}}}(k)$, the adversary F_{IB} succeeds and outputs a valid forgery, i.e. a list of ℓ' tuples $\{(id_i, m_i, SIG_i)\}_{1 \leq i \leq \ell'}$, with $\ell < \ell'$. Since it is not possible that the identities in this output list have been queried by F_{IB} to obtain the corresponding user secret keys, and on the other hand a valid signature SIG_i contains by definition a valid certificate $cert_i$ of the message $id_i \,\|\, pk_i$, under the signature scheme $\mathcal{S}$ and public key $PK = pk$, there are two options.

- If for some of the identities id_i in the output list, no blind signature query including id_i has been made by F_{IB}, then id_i has not appeared during the attack and so we have not asked for a signature on $id_i \,\|\, pk_i$ to the signing oracle associated with forger F. This means that the signature SIG_i contains a valid forgery $cert_i$ against scheme $\mathcal{S}$.

- Otherwise, we have that all the identities id_i in the output list have appeared inside some blind signature query made by F_{IB} during its attack. Since $\ell < \ell'$, there exists at least some identity id in the output list such that the number $\ell(id)$ of completed blind signature queries during the attack involving id is strictly less than the number $\ell'(id)$ of tuples involving identity id in the output list.

  If our guess was correct and $id = id_{i_*}$, then we have completed $\ell(id)$ executions of the blind signature protocol during our attack F' against the blind signature scheme $\mathcal{S}_{\mathcal{BS}}$, with public key pk_*, and we can easily obtain $\ell'(id)$ valid signatures under public key pk_* from the list output by F_{IB}, satisfying $\ell(id) < \ell'(id)$.

Summing up, we guess $id = id_{i_*}$ with probability at least $1/q$; if our guess is correct then we do not abort because the secret key query for identity $id = id_{i_*}$ is not made. In this case, a successful forgery of F_{IB} immediately implies a successful forgery of either the signature scheme $\mathcal{S}$ or the blind signature scheme $\mathcal{S}_{\mathcal{BS}}$. This completes the proof. ∎

We remark that by defining two independent adversaries it is easy to improve the security reduction to $\mathbf{Adv}^{\mathrm{forge}}_{\mathcal{S}_{\mathcal{BS}},\mathsf{F}_{\mathrm{IB}}}(k) \leq q \cdot \mathbf{Adv}^{\mathrm{forge}}_{\mathcal{S}_{\mathcal{BS}},\mathsf{F}'}(k) + \mathbf{Adv}^{\mathrm{forge}}_{\mathcal{S},\mathsf{F}}(k)$. However, since the signature scheme usually has by far better security guarantees than the blind signature scheme, the practical impact of this improvement is almost negligible.

5.5 Example Instantiation

In this section we describe the identity-based blind signature scheme which results from applying our generic construction to the standard signature scheme of Boneh-Lynn-Shacham (BLS, for short) [14] and to the blind signature scheme of Boldyreva [13]. Additionally, since BLS and Boldyreva's blind signatures share the same algebraic structure, we can use signature aggregation [17] to reduce the size of the final identity-based blind signature. The details of the protocols are as follows.

Setup $\mathsf{KG}_{\mathcal{BS}}(1^k)$: on input a security parameter k, the setup and key generation protocols of BLS signature scheme are executed. This results in two multiplicative groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order $q > 2^k$, along with a generator g of $\mathbb{G}$, such that these groups admit a bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, which must be efficiently computable and non-degenerate (that is, $\hat{e}(g, g) \neq 1$). A hash function $H : \{0,1\}^* \to \mathbb{G}$ is chosen which will be modelled as a random oracle. Finally, the master entity chooses an element $x \in \mathbb{Z}_q^*$ at random and computes $X = g^x$. The master public key is defined as $PK = (q, \mathbb{G}, \mathbb{G}_T, \hat{e}, H, X)$, whereas the master secret key is $SK = x$.

Key extraction $\mathsf{EXT}_{\mathcal{BS}}(SK, id)$: when the user secret key $USK[id]$ for some identity id is requested for the first time, the master entity chooses at random $sk \in \mathbb{Z}_q^*$ and computes $pk = g^{sk}$. Then it uses BLS to sign the message $id \,\|\, pk$; that is, it computes

User $U(PK, id, m)$:		Signer $S(USK[id])$:
		Parse $USK[id]$ as $(cert, pk, sk)$
$r \xleftarrow{\$} \mathbb{Z}_q^*$		
$R = H(m) \cdot g^r$	$\xrightarrow{\quad id, R \quad}$	
If $\hat{e}(cert, g) \neq \hat{e}(H(id \,\|\, pk), X)$ then fail	$\xleftarrow{\quad cert, pk, sig' \quad}$	$sig' = R^{sk}$
$sig \leftarrow sig' \cdot pk^{-r}$		
If $\hat{e}(sig, g) \neq \hat{e}(H(m), pk)$ then fail		
$\sigma \leftarrow sig \cdot cert$		
Return $SIG = (pk, \sigma)$		

Figure 8: Two-rounds identity based blind signing protocol.

$cert \leftarrow H(id \,\|\, pk)^x$. The resulting secret key, which is sent to the owner of the identity, is $USK[id] = (cert, pk, sk)$. The recipient can verify the correctness of the obtained secret key by checking if

$$\hat{e}(cert, g) = \hat{e}(H(id \,\|\, pk), X).$$

Blind signature $\mathsf{SIGN}_{\mathcal{BS}}$: PK is a common input for user and signer, the input of the user is (id, m) and the input of the signer is $USK[id] = (cert, pk, sk)$.

1. User U chooses at random $r \in \mathbb{Z}_q^*$ and computes $R = H(m) \cdot g^r$. Then he sends the query $(id, \text{`blindsignature?'}, R)$ to the signer.
2. If the signer does not want to sign, the protocol finishes with Out_U =‘fail’ and Out_{id} =‘not completed’. Otherwise, the signer computes $sig' = R^{sk}$, and sends the tuple $(cert, pk, sig')$ back to the user.

Once he has received the tuple $(cert, pk, sig')$, the user first verifies that

$$\hat{e}(cert, g) = \hat{e}(H(id \,\|\, pk), X).$$

If the output is 0, then the protocol finishes with Out_U =‘fail’ and Out_{id} =‘not completed’. Otherwise, the user computes $sig = sig' \cdot pk^{-r} = H(m)^{sk}(g^r)^{sk} \cdot pk^{-r} = H(m)^{sk}$, which is a valid BLS signature on the message m, with secret key sk. Actually, the user can verify that this signature is correct by checking if

$$\hat{e}(sig, g) = \hat{e}(H(m), pk).$$

Again, if the output is 0, then the protocol finishes with Out_U =‘fail’ and Out_{id} =‘not completed’. Otherwise, the user aggregates sig and $cert$ into $\sigma = sig \cdot cert$. The identity-based signature is the pair $SIG = (pk, \sigma) \in \mathbb{G}^2$. This blind signing protocol is also depicted in Figure 8.

Verification $\mathsf{VFY}_{\mathcal{BS}}(PK, id, m, SIG)$: given as input a message m, an identity id and an identity-based signature SIG that is parsed as (pk, σ), the verification protocol checks if

$$\hat{e}(\sigma, g) = \hat{e}(H(m), pk) \cdot \hat{e}(H(id \,\|\, pk), X).$$

The security of this identity-based blind signature scheme, in the random oracle model, directly follows from the security of BLS signatures and Boldyreva's blind signature scheme, and from Theorems 5.3 and 5.4. Regarding efficiency, this scheme is more efficient than any other identity-based blind signature scheme proposed in the literature (including the one in [61]), in terms of number of rounds, length of the signatures and computational cost of the protocols.

6 Conclusions

In this paper we explain how to construct generic identity-based signature schemes with additional properties, by starting from a standard signature scheme and a signature scheme with this property for the PKI-based scenario. Our method is inspired by the work from [57, 9], and works properly for many of these additional properties: verifiable encrypted signatures, (partially) blind signatures, undeniable signatures, forward-secure signatures, strongly key insulated signatures, proxy signatures, online/offline signatures, threshold signatures, aggregate signatures.

By using known results on standard signature schemes (with these additional properties, if necessary), we can deduce from our generic construction the existence of identity-based signatures with additional properties, that are provably secure in the standard model, do not need bilinear pairings, or can be based on general assumptions. This solves many open problems in the area of identity-based signatures. From a more practical point of view, our generic construction can eventually lead to very efficient schemes, when applied to the most efficient inherent signature schemes. An example of this fact can be seen in Section 5.5, where we obtain the most efficient identity-based blind signature scheme up to date (in the random oracle model), as a result of our generic construction. In general, our work provides a benchmark to measure the efficiency of existing and future schemes.

Certificateless cryptography was introduced in [4] as an alternative to both the PKI-based and the identity-based scenarios for the implementation of cryptographic protocols. The goal is to combine the best of the two worlds: digital certificates are not necessary, on the one hand, and the master entity cannot impersonate users because it does not know their complete secret keys, on the other hand. Starting from some generic construction of certificateless signatures (see, e.g., [7]), and using the techniques introduced in this paper, it is also possible to construct certificateless signature schemes with additional properties.

References

[1] IEEE P1363.3: standard for identity-based cryptographic techniques using pairings. 1

[2] Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In Kwangjo Kim and Tsutomu Matsumoto, editors, *ASIACRYPT'96*, volume 1163 of *LNCS*, pages 244–251, Kyongju, Korea, November 3–7, 1996. Springer-Verlag, Berlin, Germany. 7

[3] Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 271–286, Santa Barbara, CA, USA, August 20–24, 2000. Springer-Verlag, Berlin, Germany. 7

[4] Sattam S. Al-Riyami and Kenneth G. Paterson. Certificateless public key cryptography. In Chi-Sung Laih, editor, *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 452–473, Taipei, Taiwan, November 30 – December 4, 2003. Springer-Verlag, Berlin, Germany. 24

[5] G. Ateniese and B. de Medeiros. Identity-based chameleon hash and applications. In *Financial Cryptography'04*, pages 164–180, 2004. 15

[6] Joonsang Baek and Yuliang Zheng. Identity-based threshold signature scheme from the bilinear pairings. In *ITCC'04 (1)*, pages 124–128, 2004. 13

[7] Z. Zhang B.C. Hu, D.S. Wong and X. Deng. Certificateless signature: a new security model and an improved generic construction. *Designs, Codes and Cryptography*, 42 (2):109–126, 2007. 24

[8] Mihir Bellare and Sara K. Miner. A forward-secure digital signature scheme. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 431–448, Santa Barbara, CA, USA, August 15–19, 1999. Springer-Verlag, Berlin, Germany. 10

[9] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 268–286, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany. 1, 2, 3, 5, 24

[10] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, June 2003. 8

[11] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. 2

[12] A. Boldyreva, A. Palacio, and B. Warinschi. Secure proxy signature schemes for delegation of signing rights. Cryptology ePrint Archive, Report 2003/096, 2003. `http://eprint.iacr.org/`. 11

[13] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46, Miami, USA, January 6–8, 2003. Springer-Verlag, Berlin, Germany. 8, 19, 22

[14] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. *Journal of Cryptology*, 17 (4):297–319, 2004. 22

[15] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, April 2008. 12

[16] Dan Boneh and Matthew K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. 1

[17] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432, Warsaw, Poland, May 4–8, 2003. Springer-Verlag, Berlin, Germany. 6, 7, 14, 22

[18] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, September 2004. 8, 14

[19] Jan Camenisch, Maciej Koprowski, and Bogdan Warinschi. Efficient blind signatures without random oracles. In Carlo Blundo and Stelvio Cimato, editors, *SCN 04*, volume 3352 of *LNCS*, pages 134–148, Amalfi, Italy, September 8–10, 2004. Springer-Verlag, Berlin, Germany. 8

[20] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998. ACM Press. 2

[21] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 199–203, Santa Barbara, CA, USA, 1983. Plenum Press, New York, USA. 7, 16

[22] David Chaum. Designated confirmer signatures. In Alfredo De Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 86–91, Perugia, Italy, May 9–12, 1994. Springer-Verlag, Berlin, Germany. 15

[23] David Chaum and Hans Van Antwerpen. Undeniable signatures. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 212–216, Santa Barbara, CA, USA, August 20–24, 1990. Springer-Verlag, Berlin, Germany. 8

[24] X. Cheng, J. Liu, and X. Wang. Identity-based aggregate and verifiably encrypted signatures from bilinear pairing. In *Proceedings of ICCSA*, pages 1046–1054, 2005. 7

[25] X. Cheng, J. Liu, and X. Wang. An identity-based signature and its threshold version. In *19th International Conference on Advanced Information Networking and Applications (AINA'05)*, pages 973–977, 2005. 13

[26] S. S. M. Chow, L. C.K. Hui, S. M. Yiu, and K.P. Chow. Two improved partially blind signature schemes from bilinear pairings. In *Proceedings of ACISP 2005*, pages 316–325, 2005. 7

[27] Ivan Damgaard, Nelly Fazio, and Antonio Nicolosi. Non-interactive zero-knowledge from homomorphic encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 41–59, New York, NY, USA, March 4–7, 2006. Springer-Verlag, Berlin, Germany. 13

[28] Ivan Damgaard and Torben P. Pedersen. New convertible undeniable signature schemes. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 372–386, Zaragoza, Spain, May 12–16, 1996. Springer-Verlag, Berlin, Germany. 8

[29] Yevgeniy Dodis and Jonathan Katz. Chosen-ciphertext security of multiple encryption. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 188–209, Cambridge, MA, USA, February 10–12, 2005. Springer-Verlag, Berlin, Germany. 13

[30] Yevgeniy Dodis, Jonathan Katz, Shouhuai Xu, and Moti Yung. Key-insulated public key cryptosystems. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 65–82, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer-Verlag, Berlin, Germany. 11

[31] Shimon Even, Oded Goldreich, and Silvio Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9(1):35–67, 1996. 12

[32] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77, Santa Barbara, CA, USA, August 20–24, 2006. Springer-Verlag, Berlin, Germany. 8

[33] Steven D. Galbraith and Wenbo Mao. Invisibility and anonymity of undeniable and confirmer signatures. In Marc Joye, editor, *CT-RSA 2003*, volume 2612 of *LNCS*, pages 80–97, San Francisco, CA, USA, April 13–17, 2003. Springer-Verlag, Berlin, Germany. 8, 9

[34] Craig Gentry and Zulfikar Ramzan. Identity-based aggregate signatures. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006*, volume 3958 of *LNCS*, pages 257–273, New York, NY, USA, April 24–26, 2006. Springer-Verlag, Berlin, Germany. 14

[35] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 548–566, Queenstown, New Zealand, December 1–5, 2002. Springer-Verlag, Berlin, Germany. 4

[36] 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. 4

[37] C. Gu and Y. Zhu. An id-based verifiable encrypted signature scheme based on Hess's scheme. In *CISC'05*, pages 42–52, 2005. 6, 7

[38] J. Herranz. Deterministic identity-based signatures for partial aggregation. *The Computer Journal*, 49 (3):322–330, 2006. 14

[39] K. Chen J. Weng, S. Liu and X. Li. Identity-based key-insulated signature with secure key-updates. In *Inscrypt'06*, pages 13–26, 2006. 11

[40] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 143–154, Zaragoza, Spain, May 12–16, 1996. Springer-Verlag, Berlin, Germany. 15

[41] Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic signature schemes. In Bart Preneel, editor, *CT-RSA 2002*, volume 2271 of *LNCS*, pages 244–262, San Jose, CA, USA, February 18–22, 2002. Springer-Verlag, Berlin, Germany. 15

[42] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 150–164, Santa Barbara, CA, USA, August 17–21, 1997. Springer-Verlag, Berlin, Germany. 7, 8, 16, 18

[43] Eike Kiltz, Anton Mityagin, Saurabh Panjwani, and Barath Raghavan. Append-only signatures. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *ICALP 2005*, volume 3580 of *LNCS*, pages 434–445, Lisbon, Portugal, July 11–15, 2005. Springer-Verlag, Berlin, Germany. 15

[44] H. Krawczyk. Simple forward-secure signatures from any signature scheme. In *ACM Conference on Computer and Communications Security*, pages 108–115, 2000. 10

[45] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS 2000*, San Diego, California, USA, February 2–4, 2000. The Internet Society. 15

[46] F. Laguillaumie and D. Vergnaud. Short undeniable signatures without random oracles: the missing link. In *Indocrypt'05*, pages 283–296, 2005. 9

[47] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, October 1979. 15

[48] Benoît Libert and Jean-Jacques Quisquater. Identity based undeniable signatures. In Tatsuaki Okamoto, editor, *CT-RSA 2004*, volume 2964 of *LNCS*, pages 112–125, San Francisco, CA, USA, February 23–27, 2004. Springer-Verlag, Berlin, Germany. 9

[49] Steve Lu, Rafail Ostrovsky, Amit Sahai, Hovav Shacham, and Brent Waters. Sequential aggregate signatures and multisignatures without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 465–485, St. Petersburg, Russia, May 28 – June 1, 2006. Springer-Verlag, Berlin, Germany. 6, 7, 14

[50] Anna Lysyanskaya, Silvio Micali, Leonid Reyzin, and Hovav Shacham. Sequential aggregate signatures from trapdoor permutations. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 74–90, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany. 14

[51] M. Mambo, K. Usuda, and E. Okamoto. Proxy signatures: Delegation of the power to sign messages. *IEICE Trans. Fundamentals*, E79-A (9):1338–1353, 1996. 11

[52] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 80–99, New York, NY, USA, March 4–7, 2006. Springer-Verlag, Berlin, Germany. 8

[53] L. Trieu Phong and W. Ogata. Blind HIBE and its applications to identity-based blind signature and blind decryption. Cryptology ePrint Archive, Report 2008/327, 2008. `http://eprint.iacr.org/`. 6, 7

[54] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000. 7, 16

[55] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565, Gold Coast, Australia, December 9–13, 2001. Springer-Verlag, Berlin, Germany. 15

[56] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairings. In *Proceedings of the Symposium on Cryptography and Information Security — SCIS 2000*, pages ???–???, jan 2000. 1

[57] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53, Santa Barbara, CA, USA, August 19–23, 1985. Springer-Verlag, Berlin, Germany. 1, 5, 24

[58] Adi Shamir and Yael Tauman. Improved online/offline signature schemes. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 355–367, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany. 12

[59] W. Susilo, F. Zhang, and Y. Mu. Identity-based strong designated verifier signature schemes. In *ACISP'04*, pages 313–324, 2004. 15

[60] W. Susilo, F. Zhang, and Y. Mu. On the security of nominative signatures. In *ACISP'05*, pages 329–335, 2005. 15

[61] G. Wang W. Gao, X. Wang and F. Li. One-round id-based blind signature scheme without ROS assumption. In *Pairing'08*, pages 316–331, 2008. 7, 23

[62] J. Xu, Z. Zhang, and D. Feng. ID-based proxy signature using bilinear pairings. In *ISPA'05 International Workshop IADS*, pages 359–367, 2005. 11

[63] S. Xu, Y. Mu, and W. Susilo. Efficient authentication scheme for routing in mobile ad hoc networks. In *Proceedings of The First International Workshop on Security in Ubiquitous Computing Systems (SecUbiq 2005)*, pages 854–863, 2005. 12

[64] F. Zhang and K. Kim. Efficient ID-based blind signature and proxy signature from bilinear pairings. In *ACISP'03*, pages 312–323, 2003. 7, 11, 15

[65] Fangguo Zhang and Kwangjo Kim. ID-based blind signature and ring signature from pairings. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 533–547, Queenstown, New Zealand, December 1–5, 2002. Springer-Verlag, Berlin, Germany. 7

[66] Y. Zhou, Z. Cao, and Z. Chai. Identity based key insulated signature. In *ISPEC'06*, pages 226–234, 2006. 11