An extended abstract of this paper appears in Advances in Cryptology — ASIACRYPT '06, Lecture Notes in Computer Science Vol. ??, 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[2]

[1] Nijmegen Institute for Computing and Information Sciences,
The Netherlands
d.galindo@cs.ru.nl
http://www.cs.ru.nl/~dgalindo/

[2] CWI Amsterdam
The Netherlands
kiltz@cwi.nl, j.herranz@cwi.nl
http://kiltz.net, http://www.cwi.nl/~herranz/

Abstract

It has been demonstrated by Bellare, Neven, and Namprempre (Eurocrypt 2004) that identity-based signature schemes can be constructed from any PKI-based signature scheme. 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 PKI-based signature schemes with the same properties? Our results show that this is possible for great 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 PKI-based schemes, we conclude that such identity-based signature schemes with additional properties can be constructed, enjoying some 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

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 aggregated signatures.

Initially, all these extensions were introduced for the standard PKI-based framework, 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 [47]. 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. In his paper, Shamir already proposed an identity-based signature scheme. In contrast, the problem of designing an efficient and secure identity-based encryption scheme remained open until [11, 46].

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 is folklore 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 this signing key to sign the message. The identity-based signature itself consists of this signature along with the certificate and the public verification key.

The above idea was formalized by Bellare, Neven, and Namprempre in [6], where they propose 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 [6]. In fact, in many papers the authors do not mention the generic approach from [6] and in spite of Shamir's work from more than two decades ago [47] it still seems to be a popular "opinion" among some researchers that the construction of identity-based signatures inherently relies on bilinear pairings.

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 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 [8] (with its well-known limitations [14]).

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 $\mathcal{PS}$ 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 $\mathcal{IB\_PS}$ 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 $\mathcal{PS}$ with the required additional properties, we obtain identity-based schemes $\mathcal{IB\_PS}$ 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 $\mathcal{IB\_PS}$ scheme secure in the standard model (i.e., without random oracles); (ii) the first $\mathcal{IB\_PS}$ scheme built without using bilinear pairings; and (iii) the first $\mathcal{IB\_PS}$ based on "general assumptions" (e.g. on the sole assumption 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 $\mathcal{IB\_PS}$ is the same as in [6] and a formal security statement can be proved following basically verbatim the proofs given in [6]. 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 and undeniable signatures and hence in these cases we will lay out our constructions in more 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 $\mathcal{PS}$, an additional public key different from that of the signer has to be used in the protocol. This includes ring, designated verifier, confirmer,

[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.

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 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 scheme and measure the efficiency of the resulting identity-based one. 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 [6] 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. Then we present our main results in Section 3: 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. We do not include the details of the constructions and the security analysis for each additional property. However, as a representative example, we give in Section 4 the details concerning the (identity-based) blind signature case. We stress that we have a formal proof for all other constructions.

2 Definitions

In this section we recall the well-known syntax and definition of (identity-based) signature schemes.

2.1 Standard Signatures

A standard signature scheme $\mathcal{S} = (\mathsf{S.KG}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ consists of the following three (probabilistic polynomial-time) algorithms. The **key generation** algorithm $\mathsf{S.KG}$ takes as input a security parameter k and returns a secret key SK and a matching public key PK. We use the notation $(SK, PK) \leftarrow \mathsf{S.KG}(1^k)$ to refer to one execution of this protocol. The **signing** algorithm $\mathsf{S.Sign}$ inputs a message m and a secret key SK. The output is a signature $sig_{SK}(m)$. We denote an execution of this protocol as $sig_{SK}(m) \leftarrow \mathsf{S.Sign}(SK, m)$. The **verification** algorithm $\mathsf{S.Vfy}$

takes as input a message m, a signature $sig = sig_{SK}(m)$ and a public key PK. The output is 1 if the signature is valid, or 0 otherwise. We use the notation $\{0,1\} \leftarrow \mathsf{S.Vfy}(PK, m, sig)$ to refer to one execution of this algorithm.

Security. We will consider security against adaptively-chosen message attacks. For a formal definition one considers a forger $\mathcal{F}$ trying to attack the scheme. This situation is modeled by the following interactive game that $\mathcal{F}$ plays against a challenger.

First the challenger runs the key generation protocol $(SK, PK) \leftarrow \mathsf{S.KG}(1^k)$ and gives PK to $\mathcal{F}$. The secret key SK is kept secret by the challenger. During its execution the forger $\mathcal{F}$ adaptively chooses messages m_i, then the challenger runs $sig_i \leftarrow \mathsf{S.Sign}(SK, m_i)$ and gives the resulting signatures to $\mathcal{F}$. Eventually the adversary $\mathcal{F}$ outputs a forgery consisting of a pair (m, sig). There are two kinds of unforgeability, depending on the outputs which are considered as a successful attack by $\mathcal{F}$. In the standard case, we say that $\mathcal{F}$ succeeds if sig is a valid forgery of message m (i.e. if $1 \leftarrow \mathsf{S.Vfy}(PK, m, sig)$) and if $m \neq m_i$ for all the messages m_i that $\mathcal{F}$ queried the signature for during the attack. We define the advantage of such a forger $\mathcal{F}$ as $\mathbf{Adv}^{\mathrm{forge}}_{\mathcal{S},\mathcal{F}}(k) = \Pr[\,\mathcal{F} \text{ succeeds}\,]$. For the notion of *strong* unforgeability we relax the second condition such that we require $(m, sig) \neq (m_i, sig_i)$ for all the tuples (m_i, sig_i) that $\mathcal{F}$ has obtained during the attack and define $\mathbf{Adv}^{\mathrm{sforge}}_{\mathcal{S},\mathcal{F}}(k) = \Pr[\,\mathcal{F} \text{ succeeds}\,]$. A scheme is called *(strongly) unforgeable* if the respective advantage is a negligible function in k.

2.2 Identity-Based Signatures

An identity-based signature scheme $\mathcal{IB\_S} = (\mathsf{IB\_S.KG}, \mathsf{IB\_S.Extr}, \mathsf{IB\_S.Sign}, \mathsf{IB\_S.Vfy})$ consists of the following four (probabilistic polynomial-time) algorithms [15]. The **setup** algorithm $\mathsf{IB\_S.KG}$ takes as input a security parameter k and returns, on the one hand, the system public parameters mpk and, on the other hand, the value master secret key msk, which is known only to the master entity. We note an execution of this protocol as $(mpk, msk) \leftarrow \mathsf{IB\_S.KG}(1^k)$. The **key extraction** algorithm $\mathsf{IB\_S.Extr}$ takes as inputs mpk, the master secret key msk and an identity $id \in \{0,1\}^*$, and returns a secret key $sk[id]$ for the user with this identity. We use notation $sk[id] \leftarrow \mathsf{IB\_S.Extr}(msk, id)$ to refer to one execution of this protocol. The **signing** algorithm $\mathsf{IB\_S.Sign}$ inputs a user secret key $sk[id]$, the public parameters mpk, an identity, and a message m. The output is a signature $sig = sig_{msk}(id, m)$. We denote an execution of this protocol as $sig \leftarrow \mathsf{IB\_S.Sign}(mpk, id, sk[id], m)$. Finally, the **verification** algorithm $\mathsf{IB\_S.Vfy}$ inputs mpk, a message m, an identity id and a signature sig; it outputs 1 if the signature is valid, and 0 otherwise. To refer to one execution of this protocol, we use notation $\{0,1\} \leftarrow \mathsf{IB\_S.Vfy}(mpk, id, m, sig)$.

Security. To define security of an identity-based signature scheme [15], one considers a forger $\mathcal{F}_{\mathrm{IB}}$ trying to attack the scheme. This situation is modelled by the following game, that $\mathcal{F}_{\mathrm{IB}}$ plays against a challenger.

Initially, the challenger runs the key generation protocol $(msk, mpk) \leftarrow \mathsf{IB\_S.KG}(1^k)$ and gives mpk to $\mathcal{F}_{\mathrm{IB}}$. The secret key msk is kept secret by the challenger. During its execution the forger $\mathcal{F}_{\mathrm{IB}}$ is allowed to make two different types of queries. The forger $\mathcal{F}_{\mathrm{IB}}$ may make a key extraction query for some identity id_i. Then the challenger first checks if it has already established a user secret key for id_i. If so, the old secret key is returned. Otherwise, it stores and returns a new user secret key by running $sk[id_i] \leftarrow \mathsf{IB\_S.Extr}(msk, id_i)$. Furthermore, the forger $\mathcal{F}_{\mathrm{IB}}$ is allowed to make signature queries with respect to pairs of identities and messages (id_i, m_i). The challenger first calls its internal key extraction oracle to obtain a (a new or stored) user secret ket $sk[id_i]$. Using this user secret key the challenger runs $sig_i \leftarrow \mathsf{IB\_S.Sign}(sk[id_i], m_i)$

and returns the resulting signature sig_i to $\mathcal{F}_{\mathrm{IB}}$. Eventually, the adversary $\mathcal{F}_{\mathrm{IB}}$ outputs a forgery (id, m, sig). We say that $\mathcal{F}_{\mathrm{IB}}$ succeeds if sig is a valid signature for id and message m (i.e., if $1 \leftarrow \mathsf{IB\_S.Vfy}(mpk, id, m, sig)$), if $id \neq id_i$ for all id_i that $\mathcal{F}_{\mathrm{IB}}$ has queried user secret keys for during the attack, and if $(id, m) \neq (id_i, m_i)$ for all the tuples (id_i, m_i) that $\mathcal{F}_{\mathrm{IB}}$ has queries signatures for during the attack.

We define the advantage of such a forger $\mathcal{F}_{\mathrm{IB}}$ as $\mathbf{Adv}^{\mathrm{forge}}_{\mathcal{IB\_S},\mathcal{F}_{\mathrm{IB}}}(k) = \Pr[\mathcal{F}_{\mathrm{IB}} \text{ succeeds}]$ and a scheme is called *unforgeable* if this advantage is a negligible function in k.

3 Generic Construction of Identity-based Signatures

In this section we first outline the BNN generic transformation [6] from two standard signature schemes $\mathcal{S}$, $\mathcal{S}'$ into an identity-based signature scheme. Subsequently we study the question whether, for different types of signature schemes $\mathcal{PS}$ with additional properties, we have a (similar) generic transformation that combines $\mathcal{S}$ with $\mathcal{PS}$ to obtain $\mathcal{IB\_PS}$, where $\mathcal{IB\_PS}$ is an identity-based signature scheme with the same additional property as $\mathcal{PS}$.

Let $\mathcal{S} = (\mathsf{S.KG}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ and $\mathcal{S}' = (\mathsf{S'.KG}, \mathsf{S'.Sign}, \mathsf{S'.Vfy})$ be two (possibly equal) standard signature schemes. The generic construction of an identity-based signature scheme $\mathcal{IB\_S} = (\mathsf{IB\_S.KG}, \mathsf{IB\_S.Extr}, \mathsf{IB\_S.Sign}, \mathsf{IB\_S.Vfy})$, proposed in [6], is defined as follows.

Key Generation $\mathsf{IB\_S.KG}(1^k)$: The key generation algorithm from the standard signature scheme $\mathcal{S}$ is run to obtain the master key-pair for the identity-based signature scheme $\mathcal{IB\_S}$: $(msk, mpk) \leftarrow \mathsf{S.KG}(1^k)$.

IBS Key extraction $\mathsf{IB\_S.Extr}(msk, id_i)$: The secret key of a user with identity id_i is defined as

$$sk[id_i] = (sig_{msk}(id_i||pk_i), pk_i, sk_i), \tag{1}$$

where (pk_i, sk_i) is a random key-pair obtained by running $\mathsf{S'.KG}(1^k)$ and $sig_{msk}(id_i||pk_i) \leftarrow \mathsf{S.Sign}(msk, id_i||pk_i)$. Here the signature $sig_{msk}(id_i||pk_i)$ can be viewed as a "certificate" on the validity of pk_i.

Identity-Based Sign $\mathsf{IB\_S.Sign}(mpk, id_i, sk[id_i], m)$: Given a user secret key for identity id_i (cf. Eqn. (1)) an identity-based signature for identity id_i and message m is defined as

$$sig(id_i, m) = (sig_{msk}(id_i||pk_i), pk_i, sig_{sk_i}(m)), \tag{2}$$

where $sig_{sk_i}(m) = \mathsf{S'.Sign}(sk_i, m)$ can be computed by the possessor of the user secret key $sk[id_i]$ since sk_i is contained in $sk[id_i]$. Signature $sig_{msk}(id_i||pk_i)$ included in Eqn. (2) certifies the validity of pk_i.

Verification $\mathsf{IB\_S.Vfy}(mpk, sig)$: For verification of the identity-based signature the user checks if the first signature from Eqn. (2) is valid with respect to mpk and the "message" $id||pk_i$ (using the verification protocol $\mathsf{S.Vfy}$); and if the second signature is valid with respect to pk_i and the message m (using the verification protocol $\mathsf{S'.Vfy}$).

Bellare, Namprempre, and Neven [6] prove the following result:

Theorem 3.1 If $\mathcal{S}$ and $\mathcal{S}'$ are both secure standard signature schemes then $\mathcal{IB\_S}$ is a secure identity-based signature scheme.

Let $\mathcal{PS}$ be a signature scheme with the property $\mathcal{P}$. We extend the above construction to an IBS with additional properties $\mathcal{IB\_PS}$ in a straightforward way: as with signing/verification, all functionality provided by $\mathcal{PS}$ is "lifted" to the identity-based case. That means that (analog

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 §3.1	$\star$	$\bigstar$	$\bigstar$	$\bigstar$
BS §3.2	$\star/\bigstar^{a}$	$\bigstar$	$\bigstar$	$\bigstar$
US §3.3	$\star$	$\bigstar$	$\bigstar$	–
FSS §3.4	$\bigstar$	$\bigstar$	$\bigstar$	$\bigstar$
SKIS §3.5	$\star$	$\bigstar$	$\bigstar$	$\bigstar$
PS §3.6	$\star$	$\bigstar$	$\bigstar$	$\bigstar$
OOS §3.7	$\star$	$\bigstar$	$\bigstar$	$\bigstar$
Threshold §3.8	$\star$	$\bigstar$	$\bigstar$	–

[a] against concurrent adversaries.

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

to IB_S.Sign and IB_S.Vfy) any protocol additionally provided by $\mathcal{PS}$ is executed using the corresponding secret/public key pair (sk_i, pk_i) from the user secret key Eqn. (1). We will refer to the latter construction as the "generic construction of identity-based signatures with additional properties" or simply "generic construction".

In the rest of this section we will demonstrate that this 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 (US); forward-secure signatures (FSS); strongly key insulated signatures (SKIS); online/offline signatures (OOS); threshold signatures (TS); and aggregate signatures (AS). For most properties the generic construction can be applied without many difficulties and therefore we decided to only outline the functionality and to summarize the known results for the IBS with the additional property. For (partially) blind, undeniable, and aggregate signatures our constructions derive from the generic construction and therefore we provide additional details. Due to lack of space we are forced 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 4. 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 $\mathcal{IB}\_\mathcal{PS}$ of new identity-based signature schemes are implied by our general constructions.

3.1 Verifiably Encrypted Signatures

Verifiably encrypted signature (VES) schemes can be seen as a special extension of the standard signature primitive. VES schemes enable a user Alice to create a signature encrypted using an adjudicator's public key (the VES signature), and enable public verification if 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 [12], one in the standard model in [40]. It was further noted in [40] 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 [29] where also a concrete . security model was proposed. In contrast to [29], here we only consider a weaker (but still reasonable) model where the adjudicator has a fixed public key, i.e. it is not identity-based.

Compared to a standard signature a VES scheme has three additional algorithms: VES signing/verification (with respect to an adjudicators public key), and adjudication. Here the adjudication algorithm inputs an adjudicators secret key and transforms a VES into a standard signature. For our generic construction VES signing and verification can be lifted to the identity-based case in the same way as in the generic construction, i.e. in an IB-VES one replaces $sig_{sk_i}(m)$ in Eqn. (2) with its VES counterpart obtained by running the VES signing algorithm on sk_i, m, and the adjudicator's public key. IB-VES verification checks the certificate and the VES using the standard VES verification algorithm. Since we only consider a standard (non identity-based) adjudicator we note that there is no need to make the adjudication process identity-based. More formally we can prove the following theorem:

Theorem 3.2 If $\mathcal{S}$ is a secure standard signature scheme and $\mathcal{PS}$ is a secure verifiably encrypted signature scheme then the generic construction gives a secure identity-based verifiably encrypted signature scheme.

An pairing-based IB-VES scheme secure in the random oracle model was given in [29]. We note that the IB-VES scheme from [19] does not have a formal security proof. Using our generic construction we get an IB-VES scheme based on any trapdoor one-way function [40], and a more efficient one using [12].

3.2 (Partially) Blind Signatures

In blind signature (BS) schemes [16] 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 [33, 44]. 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 [1] and the security of such schemes was formalized in [2].

The first identity-based blind signature (IB-BS) schemes were proposed in [54, 53]. They employ bilinear pairings, but their security is not formally analyzed. Subsequent schemes were proposed in [21] but security is only provided in a weaker model (i.e. against sequential adversaries). We take the case of blind signatures to exemplify how our generic construction of identity-based signature schemes with additional properties works: in Section 4 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 4 for details).

Theorem 3.3 If $\mathcal{S}$ is a *strongly secure* standard signature scheme and $\mathcal{PS}$ is a secure (partially) blind signature scheme then a secure identity-based (partially) blind signature scheme $\mathcal{IB\_PS}$ 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 the first IB-BS scheme provably secure (in the standard model), against concurrent adversaries (by using the results from [13, 43, 26]), we obtain IB-BS schemes which do not employ bilinear pairings [7], and we obtain IB-BS schemes from any one-way trapdoor permutation [33, 26].

3.3 Undeniable Signatures

Undeniable signatures [18] (US) are signature schemes in which testing for (in)validity of a signature requires interaction with the signer. Undeniable signatures are used in applications where signed documents carry some private information about the signer and where it is considered to be an important privacy factor to limit the ability of verification.

Following [23], an undeniable signature scheme $\mathcal{US}$ consists of four algorithms $\mathcal{US} = (\mathsf{US.KG}, \mathsf{US.Sign}, \mathsf{US.Conf}, \mathsf{US.Disav})$, where $\mathsf{US.Conf}$ is a confirmation and $\mathsf{US.Disav}$ is a disavowal protocol, both being interactive algorithms run between a prover and a verifier. 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 participated in the confirmation and disavowal protocols. Intuitively this is captured by requiring the confirmation and disavowal protocols to be "zero-knowledge", such that no information is leaked besides (in)validity. With simulatability one wants to ensure that the strings representing signatures can not be recognized (i.e., distinguished from a random string) by an attacker. This security property is fulfilled if there exists a signature simulator algorithm $\mathsf{US.Sim}$, that on input of a public key and a message, outputs a simulated signature $\mathtt{sig}(m)$ which looks like a "real undeniable signature" to anyone who only knows public information and has access to confirmation/disavowal oracles.

Extending the previous definition to the identity-based setting, an identity-based undeniable signature (IB-US) scheme consists of a tuple of five algorithms $\mathcal{IB\_US} = (\mathsf{IB\_US.KG}, \mathsf{IB\_US.Extr}, \mathsf{IB\_US.Sign}, \mathsf{IB\_US.Conf}, \mathsf{IB\_US.Disav})$ where $\mathsf{IB\_US.Conf}$ and $\mathsf{IB\_US.Disav}$ are interactive algorithms run between a prover and a verifier. 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 $\mathsf{IB\_US.Sim}$. On input of the system public parameters mpk, an identity id and a message m, $\mathsf{IB\_US.Sim}$ outputs a simulated signature $\mathtt{sig}(id, m)$, which is indistinguishable from a real signature for someone having access to confirmation/disavowal oracles for the identity id.

We now sketch our generic construction of identity-based undeniable signatures. In contrast to the generic construction (cf. Eqn. (2)) we define the identity-based undeniable signature $\mathsf{IB\_US.Sign}(sk[id_i], m)$ as $sig_{sk_i}(m)$ (i.e., the certificate $sig_{msk}(id_i||pk_i)$ and pk_i are not included in the signature). In the interactive identity-based confirmation and disavowal protocols, the signer sends his certificate $(sig_{msk}(id_i||pk_i), pk_i)$ to the verifier such that the verifier can be convinced about the link between the signature and $id_i||pk_i$. Then prover (using sk_i) and verifier (using pk_i) engage in the standard US confirmation/disavowal protocol.[2]

It remains to describe the identity-based simulation algorithm $\mathsf{IB\_US.Sim}$ in terms of the underlying algorithm $\mathsf{US.Sim}$. We define the output of $\mathsf{IB\_US.Sim}(mpk, id, m)$ as $\mathsf{US.Sim}(pk'_i, m)$, where $(pk'_i, sk'_i) \leftarrow \mathsf{US.KG}(1^k)$ is a fresh key pair generated by the simulator. Note that the simulator $\mathsf{IB\_US.Sim}$ does not input the user secret key $sk[id]$ and therefore the public key pk_i from the user secret key for id_i (cf. Eqn. (1)) is information theoretically hidden from it. However, an adversary against simulatability may learn this public key pk_i from an execution of the confirmation/disavowal protocol. It turns out that to ensure that our generic IB-US

[2] At this point it may be interesting to see why the generic construction would not be simulatable and therefore not secure. In our generic construction the signature also contains $(sig_{msk}(id_i||pk_i), pk_i)$. Now, for building an identity-based signature simulator, one should be able to simulate the signatures $sig_{msk}(id_i||pk_i)$ based on the master public-key only, which is infeasible since the signature scheme $\mathcal{S}$ is assumed to be unforgeable.

construction satisfies the simulatability property it is sufficient to require the scheme $\mathcal{US}$ to be anonymous in the sense of [27]. A scheme $\mathcal{US}$ is said to be *anonymous* if (roughly) 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 $\mathsf{US.Sign}(sk_0, m)$ and $\mathsf{US.Sign}(sk_1, m)$. More formally, we can prove the following theorem:

Theorem 3.4 If $\mathcal{S}$ is a secure standard signature scheme and $\mathcal{US}$ is a secure anonymous undeniable signature scheme then $\mathcal{IB\_US}$ as outlined above is a secure identity-based undeniable signature scheme.

As far as we know, only one IB-US has been previously presented in [39]. This scheme uses bilinear pairings and it is proved secure in the random oracle model. We stress that the security model in [39] seems to be incomplete, as the authors do not consider simulatability.

In [27], an anonymous PKI-based 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 [37]; it does not employ bilinear pairings, but the disavowal protocol is quite inefficient. Using these anonymous US schemes [27, 37], we can obtain secure IB-US schemes in the random oracle model and also in the standard model, based on different computational assumptions, which do not employ bilinear pairings.

3.4 Forward-Secure Signatures

In a forward-secure signature (FSS) scheme the verification key is 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.

History on FSS: FSS schemes were studied for the first time in [5], 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 [35]. 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) 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 $sk[id]_j$ is updated at regular intervals. Roughly speaking, the initial signing key $sk[id]_0$ 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 defense against key exposure used in identity-based cryptography, in which the master entity issues new private keys $sk[id||j]$ to the user with identity id at every time period j. The latter approach heavily relies on the master entity and increases the (costly) communication between the entity and the users.

Theorem 3.5 If $\mathcal{S}$ is a secure standard signature scheme and $\mathcal{FS}$ is a secure forward-secure signature scheme then the generic construction gives a secure identity-based forward-secure signature scheme $\mathcal{IB\_FS}$.

As a consequence of this theorem IB-FSS can be constructed from any one-way function [35].

3.5 (Strongly) Key Insulated Signatures

The concept of (strongly) key insulated signatures (SKIS) was introduced in [24] and is quite similar to the one of FSS. Without going into details we remark that the generic construction of identity-based SKIS is secure provided the underlying SKIS is secure. SKIS signatures can be built from any one-way function [24], which implies our generic construction yields identity-based SKIS schemes from any one-way function. Previously, an identity-based SKIS using bilinear pairings and random oracles has been proposed in [55].

3.6 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 [42]. The first formal analysis of the security of PKI-based proxy signatures was done in [9] where is 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). In general, one looks for more efficient constructions of (identity-based) proxy signature schemes than this generic constructions. Our generic construction to obtain an IB-PS from any PKI-based PS works in general, provided the public key of the proxy signer is not strictly needed in the delegation phase of the considered PKI-based PS (which is the case in general, where the public key is only used as an identifier of the proxy, and so it can be replaced with the identity of the proxy in the constructed IB-PS). Summing up, we obtain the following result.

Theorem 3.6 If $\mathcal{S}$ is a secure standard signature scheme and $\mathcal{PS}$ is a secure proxy signature scheme then the generic construction gives a secure identity-based proxy signature scheme $\mathcal{IB\_PS}$.

History on IB-PS: The first IB-PS appeared in [53], but they lacked of a formal security analysis, since the first formal security model for IB-PS (which was adapted from the one in [9]) came later, in [51]. All these existing proposals of IB-PS employ bilinear pairings, and their security is proved in the random oracle model. With our certificate-based approach, we can easily obtain IB-PS which do not employ bilinear pairings and whose security can be proved in the standard model. Furthermore, based on [9] we obtain an IB-PS scheme based on any one-way function.

3.7 Online/Offline Signatures

In online/offline signatures signing is split into two phases: the offline and online phase. The idea is to shift the major computational overhead to the offline phase, whereas the online phase requires only a very low computational overhead.

Online/offline signatures were introduced in [25]. They presented a general method for converting any signature scheme into an online/offline signature scheme which was later improved in [48]. Using our generic construction we can make identity-based signing online/offline.

Theorem 3.7 If $\mathcal{S}$ is a secure standard signature scheme and $\mathcal{PS}$ is a secure online/offline signature scheme then the generic construction gives a secure online/offline signature scheme $\mathcal{IB\_PS}$.

We are only aware of one identity-based online/offline signature scheme [52] in the literature that is in the random oracle model and uses bilinear pairings. Applying the known generic construction [25] to our construction we get identity-based online/offline signature scheme based on one-way functions.

3.8 Threshold Signatures

Threshold signatures (TS) are used whenever the ability to sign must be decentralized. The idea is to share the signing power (the master secret key) among a number of different players, in such a way that signing is possible only when a sufficiently large enough number of honest players cooperate together. A PKI-based non-interactive threshold signature schemes in the standard model and without pairings has recently been proposed in [22].

Identity-Based Threshold Signatures (IB-TS) were introduced in [4], to be used in a context where the signing key $sk[id]$ is shared by a collective of signers with a common identity id. Given any j-th share $sk[id]_j$ of the signing key it is possible to (non-interactively) create a j-th signature share $sig(id, m)_j$, so that a full signature $sig(id, m)$ is obtained by combining a sufficiently large fraction of correctly generated signature shares from different (honest) players. More IB-TS schemes were proposed in [20].

In the following, an IB-TS construction based on our generic construction is outlined. The components are a signature scheme $\mathcal{S}$ and a threshold signature scheme $\mathcal{TS}$. Let $(msk, mpk) \leftarrow \mathsf{S.KG}(1^k)$ be the master entity keys. For each identity id, the master entity executes the key generation algorithm for $\mathcal{TS}$, obtaining a verification key pk and a set of shares $\{sk_1, \ldots, sk_n\}$ of the matching secret key. Then, $sk[id]_j$ (i.e. the j-th share of the signing key $sk[id]$) is defined as $sk[id]_j = (sig_{msk}(id||pk), pk, sk_j)$. In a similar fashion, $sig(id, m)_j$ (i.e. the j-th signature share for a message m by the j-th player holding the identity id) is defined as $(sig_{msk}(id||pk), pk, sig(m)_j)$, where $sig(m)_j$ denotes the signature share on message m obtained by the j-th player when applying the signing protocol of the PKI-based threshold scheme $\mathcal{TS}$. The full signature $sig(id, m)$ is computed by combining signature shares $sig(id, m)_j$ (using the combining algorithm of $\mathcal{TS}$ with inputs shares $sig(m)_j$).

Note that, if the signing phase of the PKI-based threshold signature scheme $\mathcal{TS}$ is non-interactive, we obtain a non-interactive identity-based threshold signature scheme (i.e., comparable to that in [4]).

Theorem 3.8 If $\mathcal{S}$ is a secure standard signature scheme and $\mathcal{PS}$ is a secure threshold signature scheme then the generic construction gives a secure identity-based threshold signature scheme $\mathcal{IB\_PS}$.

As a consequence of this theorem and the work [22], IB-TS schemes can be obtained from RSA or discrete-log based signatures, without resorting to random oracles.

3.9 Aggregate Signatures

The idea of an aggregate signature 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. In the PKI-based scenario, an execution of such an aggregation mechanism can be represented as

$$Ag\_Sig \longleftarrow \text{Aggregate } (\; \{(pk_i, m_i, sig_{sk_i}(m_i)\}_{1 \leq i \leq n} \;).$$

The main goal in the design of such protocols is that the length of Ag_Sig be 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 Ag_Sig.

The idea of aggregate signatures was introduced in [12], where a scheme with constant-length aggregate signatures is presented and analyzed, based on the signature scheme of [10]. In the identity-based framework, the only proposal which achieves constant-length aggregation is that of [28]; however, this scheme only works in a more restrictive scenario where some interaction or sequentiality is needed among the signers of the messages which later will be aggregated (in the same direction as [41, 40] 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 [30], 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.

Using the approach of this work, we can achieve exactly the same level of partial aggregation for identity-based signatures. In effect, let us consider our generic construction, and let us assume that the employed PKI-based signature scheme $\mathcal{S}$ allows constant-length aggregation. The the input of the aggregation algorithm would be $\{(id_i, sig_{msk}(id_i||pk_i), pk_i, m_i, sig_{sk_i}(m_i)\}_{1\leq i\leq n}$, where $sig_{msk}(id_i||pk_i)$ and $sig_{sk_i}(m_i)$ are signatures resulting from scheme $\mathcal{S}$, and can therefore be aggregated into a PKI-based aggregate signature Ag_Sig, of constant-length. Then the final identity-based aggregate signature would be

$$IB\_Ag\_Sig \;=\; (Ag\_Sig, pk_1, \ldots, pk_n).$$

This aggregate signature, along with the n messages and the n identities, is sufficient to verify the correctness of the n signatures. Therefore, similar to [30], the length of the identity-based aggregate signature IB_Ag_Sig is linear with respect to the number of different signers (and not with respect to the number of messages).

3.10 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 [45, 54]; designated verifier signatures [31, 49]; confirmer signatures [17]; chameleon signatures [36, 3]; and nominative signatures [50].

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 [38], homomorphic signatures [32], etc.) as well. We also note that our generic construction can be extended to the case of hierarchical identity-based signatures (HIBS) using certificate-chains [34]. Furthermore, combinations of different additional properties are possible, e.g. it is possible to give a generic construction of identity-based threshold undeniable signatures based on the existence threshold undeniable signatures.

4 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.

4.1 Blind Signature Schemes

Blind signature schemes were introduced in [16] 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 $\mathcal{BS} = (\mathsf{BS.KG}, \mathsf{BS.Sign}, \mathsf{BS.Vfy})$ consists of the following (partially interactive) algorithms.

The **key generation** algorithm $\mathsf{BS.KG}$ takes as input a security parameter k and returns a secret key sk and a matching public key pk. We use notation $(sk, pk) \leftarrow \mathsf{BS.KG}(1^k)$ to refer to one execution of this protocol. The **blind signing** algorithm $\mathsf{BS.Sign}$ 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 signature $sig = sig_{sk}(m)$. We use notation $(Out_U, Out_S) \leftarrow \mathsf{BS.Sign}(Inp_U, Inp_S)$ to refer to one execution of this interactive protocol. Finally, the **verification** algorithm $\mathsf{BS.Vfy}$ is the same verification protocol as in standard signature schemes. To refer to one execution of this protocol, we use notation $\{0, 1\} \leftarrow \mathsf{BS.Vfy}(m, sig)$.

Blindness. Intuitively, the blindness property captures the notion of a signer who tries to obtain some information about the messages he is signing for some user. Formally, this notion is defined by the following game that an adversary (signer) $\mathcal{B}$ plays against a challenger (who plays the role of a user).

First the adversary $\mathcal{B}$ runs the key generation protocol $(sk, pk) \leftarrow \mathsf{BS.KG}(1^k)$. Then the adversary $\mathcal{B}$ 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 one bit $b \in \{0, 1\}$ and then the interactive signing protocol is executed two times (possibly in a concurrent way), resulting in $(Out_{U,b}, Out_{S,b}) \leftarrow \mathsf{BS.Sign}(Inp_{U,b}, Inp_{S,b})$ and $(Out_{U,1-b}, Out_{S,1-b}) \leftarrow \mathsf{BS.Sign}(Inp_{U,1-b}, Inp_{S,1-b})$, where adversary $\mathcal{B}$ 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 $\mathcal{B}$ 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 $\mathcal{B}$ succeeds if $b' = b$ and define its advantage in the above game as $\mathbf{Adv}^{\mathrm{blind}}_{\mathcal{BS},\mathcal{B}}(k) = |\Pr[\, b' = b\,] - 1/2|$. A scheme $\mathcal{BS}$ has the blindness property if, for all PPT adversaries $\mathcal{B}$, $\mathbf{Adv}^{\mathrm{blind}}_{\mathcal{BS},\mathcal{B}}(k)$ is a negligible function (with respect to the security parameter k). If $\mathbf{Adv}^{\mathrm{blind}}_{\mathcal{BS},\mathcal{B}}(k)$ is exactly 0, for any (possibly computationally unbounded) adversary $\mathcal{B}$, 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., [33, 44]), we consider the one from [33], which is given by the following game that an adversary $\mathcal{F}$ (user or forger) plays against a challenger (signer).

First the challenger runs the key generation protocol $(pk, sk) \leftarrow \mathsf{BS.KG}(1^k)$ and gives pk to $\mathcal{F}$, whereas the secret key sk is kept secret by the challenger. During its execution the forger $\mathcal{F}$ adaptively chooses messages m_j, then the interactive signing protocol $(Out_U, Out_S) \leftarrow \mathsf{BS.Sign}(Inp_U, Inp_S)$ is executed (possibly in a concurrent way), where the adversary $\mathcal{F}$ 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 the adversary $\mathcal{F}$ outputs a list of ℓ' tuples $\{(m_i, sig_i)\}_{1 \le i \le \ell'}$.

We say that $\mathcal{F}$ *succeeds* if $\ell < \ell'$ and $1 \leftarrow \mathsf{BS.Vfy}(pk, m_i, sig_i)$, for all $i = 1, \ldots, \ell'$.

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

4.2 Identity-Based Blind Signature Schemes

Analogously, an identity-based blind signature scheme $\mathcal{IB\_BS} = (\mathsf{IB\_BS.KG}, \mathsf{IB\_BS.Extr}, \mathsf{IB\_BS.Sign}, \mathsf{IB\_BS.Vfy})$ consists of the following algorithms.

The **setup** algorithm $\mathsf{IB\_BS.KG}$ takes as input a security parameter k and returns, on the one hand, the master public key mpk and, on the other hand, the value master secret key msk, which is known only to the master entity. We note an execution of this protocol as $(msk, mpk) \leftarrow \mathsf{IB\_BS.KG}(1^k)$. The **key extraction** algorithm $\mathsf{IB\_BS.Extr}$ takes as inputs mpk, the master secret key msk and an identity $id \in \{0,1\}^*$, and returns a secret key $sk[id]$ for the user with this identity. We use notation $sk[id] \leftarrow \mathsf{IB\_BS.Extr}(msk, id)$ to refer to one execution of this protocol. The **blind signing** algorithm $\mathsf{IB\_BS.Sign}$ is an interactive protocol between a user U and a signer with identity id. The common input for them is mpk. 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 $sk[id]$. 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 $sig = sig_{msk}(id, m)$. We use notation $(Out_U, Out_{id}) \leftarrow \mathsf{IB\_BS.Sign}(mpk, Inp_U, Inp_{id})$ to refer to one execution of this interactive protocol. Finally, the **verification** algorithm $\mathsf{IB\_BS.Vfy}$ takes as input mpk, a message m, an identity id and a signature sig; it outputs 1 if the signature is valid with respect to the public key mpk and the identity id, and 0 otherwise. To refer to one execution of this protocol, we use notation $\{0,1\} \leftarrow \mathsf{IB\_BS.Vfy}(mpk, id, m, sig)$.

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 $(msk, mpk) \leftarrow \mathsf{IB\_BS.KG}(1^k)$, the key extraction protocol $sk[id] \leftarrow \mathsf{IB\_BS.Extr}(msk, id)$, and the interactive signing protocol $(Out_U, Out_{id}) \leftarrow \mathsf{IB\_BS.Sign}(mpk, Inp_U, Inp_{id})$, where $Inp_U = (id, m)$ and $Inp_{id} = sk[id]$, the following property must be satisfied:

$$Out_{id} = '\text{completed}' \implies \Big(\, 1 \leftarrow \mathsf{IB\_BS.Vfy}(mpk, id, m, Out_U)\,\Big).$$

Blindness. Blindness of an identity-based blind signature scheme is defined by a game played between a challenger and an adversary. This adversary $\mathcal{B}_{\text{IB}}$ 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 $(msk, mpk) \leftarrow \mathsf{IB\_BS.KG}(1^k)$ and gives mpk to $\mathcal{B}_{\text{IB}}$. The master secret key msk is kept secret by the challenger. The adversary $\mathcal{B}_{\text{IB}}$ is allowed to query for secret keys of identities id_i of his choice. The challenger runs $sk[id_i] \leftarrow \mathsf{IB\_BS.Extr}(msk, id_i)$ and gives the resulting secret key $sk[id_i]$ to $\mathcal{B}_{\text{IB}}$. If the same identity is asked again, the same value $sk[id_i]$ must be returned by the challenger. At some point, the adversary $\mathcal{B}_{\text{IB}}$ 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 $(Out_{U,b}, Out_{id^*,b}) \leftarrow \mathsf{IB\_BS.Sign}(Inp_{U,b}, Inp_{id^*,b})$ and $(Out_{U,1-b}, Out_{id^*,1-b}) \leftarrow \mathsf{IB\_BS.Sign}(Inp_{U,1-b}, Inp_{id^*,1-b})$, where adversary $\mathcal{B}_{\text{IB}}$ plays the role of the signer 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 $\mathcal{B}_{\text{IB}}$ outputs its guess b'.

We say that such an adversary $\mathcal{B}$ succeeds if $b' = b$ and define its advantage in the above game as $\mathbf{Adv}^{\text{ib-blind}}_{\mathcal{IB\_BS},\mathcal{B}_{\text{IB}}}(k) = |\Pr[\,b' = b\,] - 1/2|$. A scheme $\mathcal{IB\_BS}$ has the blindness property if, for all PPT adversaries $\mathcal{B}_{\text{IB}}$, $\mathbf{Adv}^{\text{ib-blind}}_{\mathcal{IB\_BS},\mathcal{B}_{\text{IB}}}(k)$ is a negligible function (with respect to the security parameter k). If $\mathbf{Adv}^{\text{ib-blind}}_{\mathcal{B}_{\text{IB}}}(k)$ is exactly 0, for any (possibly computationally unbounded) adversary $\mathcal{B}_{\text{IB}}$, 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 [33] for standard PKI-based blind signatures. A forger $\mathcal{F}_{\text{IB}}$ 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 $(msk, mpk) \leftarrow \mathsf{IB\_BS.KG}(1^k)$ and gives mpk to $\mathcal{F}_{\text{IB}}$. The master secret key msk is kept secret by the challenger. Then the forger $\mathcal{F}_{\text{IB}}$ can make two kinds of queries to the challenger. On the one hand, $\mathcal{F}_{\text{IB}}$ can ask for the secret key of an identity id_i of his choice; the challenger runs $sk[id_i] \leftarrow \mathsf{IB\_BS.Extr}(msk, id_i)$ and gives the resulting user secret key $sk[id_i]$ to $\mathcal{F}_{\text{IB}}$. If an identity id_i is asked twice, the challenger must returns the same secret key $sk[id_i]$. On the other hand, the forger $\mathcal{F}_{\text{IB}}$ can ask for the execution of the blind signing protocol: $\mathcal{F}_{\text{IB}}$ chooses pairs (id_j, m_j), then the challenger first runs $sk[id_j] \leftarrow \mathsf{IB\_BS.Extr}(msk, id_j)$ to get the secret key $sk[id_j]$ for this identity. After that, the interactive signing protocol $(Out_U, Out_{id}) \leftarrow \mathsf{IB\_BS.Sign}(mpk, Inp_U, Inp_{id})$ is executed (possibly in a concurrent way), where the adversary $\mathcal{F}_{\text{IB}}$ 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 $sk[id_j]$. Let ℓ be the number of such queries that finish with $Out_{id_j} =$'completed'. Eventually, the adversary $\mathcal{F}_{\text{IB}}$ finally outputs a list of ℓ' tuples $\{(id_i, m_i, sig_i)\}_{1 \le i \le \ell'}$. We say that $\mathcal{F}_{\text{IB}}$ *succeeds* if:

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

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

4.3 Constructing Identity-Based Blind Signature Schemes

Let $\mathcal{S} = (\mathsf{S.KG}, \mathsf{S.Sign}, \mathsf{S.Vfy})$ be a standard signature scheme and let $\mathcal{BS} = (\mathsf{BS.KG}, \mathsf{BS.Sign}, \mathsf{BS.Vfy})$ be a blind signature scheme. We construct an identity-based blind signature scheme $\mathcal{IB\_BS} = (\mathsf{IB\_BS.KG}, \mathsf{IB\_BS.Sign}, \mathsf{IB\_BS.Extr}, \mathsf{IB\_BS.Vfy})$ as follows.

Setup $\mathsf{IB\_BS.KG}(1^k)$: on input a security parameter k, the key generation protocol $\mathsf{S.KG}$ of $\mathcal{S}$ is executed, resulting in $(SK, PK) \leftarrow \mathsf{S.KG}(1^k)$. The master public key is defined as $mpk = PK$, whereas the master secret key stored by the master entity is $msk = SK$.

Key extraction $\mathsf{IB\_BS.Extr}(msk, id_i)$: when the user secret key $sk[id_i]$ for some identity id_i is requested, the master entity first checks if it already has established a user secret key for id_i. If so, the old secret key is returned. Otherwise it generates and stores a new user secret key as follows: it runs the key generation protocol of the blind signature scheme $\mathcal{BS}$, resulting in $(sk_i, pk_i) \leftarrow \mathsf{BS.KG}(1^k)$. Then it uses signature scheme $\mathcal{S}$ to sign the "message" $id_i \parallel pk_i$, that is, it executes $sig_{msk}(id_i \parallel pk_i) \leftarrow \mathsf{S.Sign}(msk, id_i \parallel pk_i)$. The resulting secret key, which is sent to the owner of the identity, is $sk[id_i] = (sk_i, pk_i, sig_{msk}(id_i \parallel pk_i))$. The recipient can verify the obtained secret key by executing $\{0,1\} \leftarrow \mathsf{S.Vfy}(mpk, id_i \parallel pk_i, sig_{msk}(id_i||pk_i))$; if the output is 1, then the secret key is accepted.

Blind signature $\mathsf{IB\_BS.Sign}$: the interactive protocol between a user U and a signer with identity id_i consists of the following steps (recall that mpk is a common input for user and signer, the input of the user is (id_i, m) and the input of the signer is $sk[id_i]$).

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

2. If the signer does not want to sign, the protocol finishes with $Out_U =$'fail' and $Out_{id_i} =$'not completed'. Otherwise, the signer sends $(pk_i, sig_{msk}(id_i||pk_i))$ back to the user.

3. The user runs $\{0,1\} \leftarrow \mathsf{S.Vfy}(mpk, id_i||pk_i, sig_{msk}(id_i||pk_i))$. If the output is 0, then the protocol finishes with $Out_U =$'fail' and $Out_{id_i} =$'not completed'.

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

Verification $\mathsf{IB\_BS.Vfy}(mpk, id_i, m, sig(id_i, m_i))$: given as input a message m, an identity id_i and an identity-based signature $sig(id_i, m_i)$ that is parsed as $(sig_{msk}(id_i||pk_i), pk_i, sig_{sk_i}(m))$, the verification protocol works as follows. The two verification protocols, of schemes $\mathcal{S}$ and $\mathcal{BS}$, are executed: $\{0,1\} \leftarrow \mathsf{S.Vfy}(mpk, id_i||pk_i, sig_{msk}(id_i||pk_i))$ and $\{0,1\} \leftarrow \mathsf{BS.Vfy}(pk_i, m, sig_{sk_i}(m))$. If both outputs are 1, then the final output of this protocol is also 1. Otherwise, the output is 0.

4.4 Security Analysis

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

Theorem 4.1 Assume the signature scheme $\mathcal{S}$ is strongly unforgeable and the blind signature scheme $\mathcal{BS}$ is blind. Then the identity-based blind signature scheme $\mathcal{IB\_BS}$ constructed in Section 4.3 is blind.

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

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

We now construct $\mathcal{F}$ and $\mathcal{B}$.

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

Secret key queries. Adversary $\mathcal{B}_{\text{IB}}$ is allowed to make secret key queries for identities id_i of its choice. To answer this query, we run the key generation protocol of the blind signature scheme $\mathcal{BS}$ to obtain $(sk_i, pk_i) \leftarrow \mathsf{BS.KG}(1^k)$. Then we send the query $m_i = id_i \parallel pk_i$ to the signing oracle associated to the forger $\mathcal{F}$, and obtain as answer a valid signature sig_i with respect to scheme $\mathcal{S}$ and public key $PK = mpk$. Then we send to $\mathcal{B}_{\text{IB}}$ the consistent answer $sk[id_i] = (sk_i, pk_i, sig_i)$. We store all this information in some table. If the same identity is asked twice by $\mathcal{B}_{\text{IB}}$, then the same secret key is given as answer.

Challenge. At some point, $\mathcal{B}_{\text{IB}}$ will output some challenge identity id_* and two messages m_0, m_1. Without loss of generality we can assume that $\mathcal{B}_{\text{IB}}$ had already asked for the secret key of this identity (otherwise, we generate it now and send it to $\mathcal{B}_{\text{IB}}$), obtaining $sk[id_*] = (sk_*, pk_*, sig_*)$. Then we start constructing an adversary $\mathcal{B}$ against the blindness of the blind signature scheme $\mathcal{BS}$, 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 $\mathcal{B}_{\text{IB}}$, where $\mathcal{B}_{\text{IB}}$ acts as a signer and we act as the user. For both executions, we first send $(id_*, '\text{blindsignature?}')$ to $\mathcal{B}_{\text{IB}}$. As answers, we will obtain $(pk_*^{(0)}, sig_*^{(0)})$ and $(pk_*^{(1)}, sig_*^{(1)})$ from $\mathcal{B}_{\text{IB}}$, where $sig_*^{(j)}$ is a valid signature on $id_* \parallel pk_*^{(j)}$, for both $j = 0, 1$.

If $(pk_*^{(j)}, sig_*^{(j)}) \neq (pk_*, sig_*)$ for either $j = 0$ of $j = 1$, then $\mathcal{F}$ outputs $sig_*^{(j)}$ 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 $\mathcal{F}$ 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 $(pk_*^{(j)}, sig_*^{(j)}) = (pk_*, sig_*)$ for both $j = 0, 1$ and the two first steps in the two executions of the interactive signing protocol are identical. Then we run the two execution of the blind signing protocol of scheme $\mathcal{BS}$, playing the role of the signer: we obtain from $\mathcal{B}_{\text{IB}}$ the information that we must send to the challenger (user) of $\mathcal{BS}$, and this challenger sends back to us the information that we must provide to $\mathcal{B}_{\text{IB}}$. This challenger of $\mathcal{BS}$ is the one who chooses the bit $b \in \{0, 1\}$.

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

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

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

We stress that the signature scheme $\mathcal{S}$ really has to be *strongly* unforgeable. Otherwise an signer can break blindness by using different versions of $sk[id_i]$ in different signing sessions and later use this information to trace the user.

Theorem 4.2 Assume the standard signature scheme $\mathcal{S}$ is unforgeable and the blind signature scheme $\mathcal{BS}$ is unforgeable. Then the identity-based blind signature scheme $\mathcal{IB\_BS}$ from Section 4.3 is unforgeable.

Proof: The proof of Theorem 4.2 is similar to the one of Theorem 4.1. We prove that if there exists a successful adversary $\mathcal{F}_{\mathrm{IB}}$ against the unforgeability of the scheme $\mathcal{IB\_BS}$, then there exists either a successful forger $\mathcal{F}$ against the unforgeability of the signature scheme $\mathcal{S}$ or a successful adversary $\mathcal{F}'$ against the unforgeability of the blind signature scheme $\mathcal{BS}$. In particular, we show that

$$\mathbf{Adv}^{\mathrm{forge}}_{\mathcal{IB\_BS},\mathcal{F}_{\mathrm{IB}}}(k) \leq q \cdot (\mathbf{Adv}^{\mathrm{forge}}_{\mathcal{BS},\mathcal{F}'}(k) + \mathbf{Adv}^{\mathrm{forge}}_{\mathcal{S},\mathcal{F}}(k)),$$

where q is an upper bound for the total number of different identities appearing in $\mathcal{F}_{\mathrm{IB}}$'s queries during the security experiment.

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

Setup. Forger $\mathcal{F}$ receives as initial input some public key PK for the signature scheme $\mathcal{S}$. Then we initialize adversary $\mathcal{F}'$ by providing it with $mpk = PK$. Then the adversary $\mathcal{F}_{\mathrm{IB}}$ 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 $\mathcal{F}_{\mathrm{IB}}$'s queries). We also start constructing an adversary $\mathcal{F}$ against the unforgeability of the blind signature scheme $\mathcal{BS}$, 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 $\mathcal{F}_{\mathrm{IB}}$, where the indices refer to the order of appearance (id_1 is the first identity that appears in some $\mathcal{F}_{\mathrm{IB}}$'s queries, and so on), we act as follows:

- If $i \neq i_*$, then we run the key generation protocol of the blind signature scheme $\mathcal{BS}$ to obtain $(sk_i, pk_i) \leftarrow \mathsf{BS.KG}(1^k)$. Then we send the query $m_i = id_i \parallel pk_i$ to the signing oracle associated to the forger $\mathcal{F}$, and we obtain as answer a valid signature sig_i with respect to the scheme $\mathcal{S}$ and public key $mpk = PK$.
- For i_*-th identity, we send the query $m_{i_*} = id_{i_*} \parallel pk_*$ to the signing oracle associated to the forger $\mathcal{F}$, and we obtain as answer a valid signature sig_{i_*}.

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

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 $sk[id_j]$ 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 $\mathcal{F}_{\mathrm{IB}}$ the values (pk_*, sig_{i_*}).

For the rest of the protocol execution, we receive messages from $\mathcal{F}_{\rm IB}$, we forward them to the blind signing oracle associated with the adversary $\mathcal{F}'$ 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 $\mathcal{F}_{\rm IB}$.

Let ℓ be the number of such blind signature queries that are successfully completed. With probability $\mathbf{Adv}^{\rm forge}_{\mathcal{IB\_BS},\mathcal{F}_{\rm IB}}(k)$, the adversary $\mathcal{F}_{\rm IB}$ succeeds and outputs a valid forgery, a list of ℓ' tuples $\{(id_i, m_i, sig_i(id_i, m_i))\}_{1 \leq i \leq \ell'}$, with $\ell < \ell'$. Since it is not possible that the identities in this output list have been queried by $\mathcal{F}_{\rm IB}$ to obtain the corresponding user secret keys, and on the other hand the valid signature $sig_i(id_i, m_i)$ contains by definition a valid signature sig_i of the message $id_i \parallel pk_i$, under the signature scheme $\mathcal{S}$ and public key $mpk = PK$, 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 $\mathcal{F}_{\rm IB}$, then id_i has not appeared during the attack and so we have not asked for a signature on $id_i \parallel pk_i$ to the signing oracle associated with forger $\mathcal{F}$. This means that the signature $sig_i(id_i, m_i)$ is a valid forgery 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 $\mathcal{F}_{\rm IB}$ during its attack. Since $\ell < \ell'$, we 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 during our attack $\mathcal{F}'$ against the blind signature scheme $\mathcal{BS}$ $\ell(id)$ executions of the blind signature protocol, with public key pk_*, and we can easily obtain $\ell'(id)$ valid signatures under public key pk_* from the list output by $\mathcal{F}_{\rm IB}$, 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 $\mathcal{F}_{\rm IB}$ immediately implies a successful forgery of either the signature scheme $\mathcal{S}$ or the blind signature scheme $\mathcal{BS}$. This completes the proof. ▌

We remark that by defining two independent adversaries it is easy to improve the security reduction to $\mathbf{Adv}^{\rm forge}_{\mathcal{IB\_BS},\mathcal{F}_{\rm IB}}(k) \leq q \cdot \mathbf{Adv}^{\rm forge}_{\mathcal{BS},\mathcal{F}'}(k) + \mathbf{Adv}^{\rm forge}_{\mathcal{S},\mathcal{F}}(k)$ but 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.

Acknowledgements

We thank the anonymous referees for their comments that helped improving the presentation of our results.

The work of the second author has been carried out during the tenure of an ERCIM Fellowship (`http://www.ercim.org/activity/fellows/`). The third author was supported by the research program Sentinels (`http://www.sentinels.nl`). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

References

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

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

[3] G. Ateniese and B. de Medeiros. Identity-based chameleon hash and applications. In *Financial Cryptography'04*, pages 164–180, 2004. 12

[4] J. Baek and Y. Zheng. Identity-based threshold signature scheme from the bilinear pairings. In *ITCC (1)*, pages 124–128, 2004. 11

[5] Mihir Bellare and Sara K. Miner. A forward-secure digital signature scheme. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 431–448, Santa Barbara, CA, USA, August 15–19, 1999. Springer-Verlag, Berlin, Germany. 9

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

[7] 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, 2003. 7

[8] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *Advances in Cryptology – EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111, Perugia, Italy, May 9–12, 1994. Springer-Verlag, Berlin, Germany. 1

[9] 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/`. 10

[10] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. *Journal of Cryptology*, 17 (4):297–319, 2004. 11

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

[12] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432, Warsaw, Poland, May 4–8, 2003. Springer-Verlag, Berlin, Germany. 6, 7, 11

[13] Jan Camenisch, Maciej Koprowski, and Bogdan Warinschi. Efficient blind signatures without random oracles. In Carlo Blundo and Stelvio Cimato, editors, *SCN 04: 4th International Conference on Security in Communication Networks*, volume 3352 of *Lecture Notes in Computer Science*, pages 134–148, Amalfi, Italy, September 8–10, 2004. Springer-Verlag, Berlin, Germany. 7

[14] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998. ACM Press. 1

[15] Jae Choon Cha and Jung Hee Cheon. An identity-based signature from gap Diffie-Hellman groups. In Yvo Desmedt, editor, *PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 18–30, Miami, USA, January 6–8, 2003. Springer-Verlag, Berlin, Germany. 4

[16] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology – CRYPTO'82*, pages 199–203, Santa Barbara, CA, USA, 1983. Plenum Press, New York, USA. 7, 13

[17] David Chaum. Designated confirmer signatures. In Alfredo De Santis, editor, *Advances in Cryptology – EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 86–91, Perugia, Italy, May 9–12, 1994. Springer-Verlag, Berlin, Germany. 12

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

[19] 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

[20] 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. 11

[21] 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

[22] Ivan Damgard, Nelly Fazio, and Antonio Nicolosi. Non-interactive zero-knowledge from homomorphic encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 41–59, New York, NY, USA, March 4–7, 2006. Springer-Verlag, Berlin, Germany. 11

[23] Ivan Damgard and Torben P. Pedersen. New convertible undeniable signature schemes. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 372–386, Saragossa, Spain, May 12–16, 1996. Springer-Verlag, Berlin, Germany. 8

[24] Yevgeniy Dodis, Jonathan Katz, Shouhuai Xu, and Moti Yung. Key-insulated public key cryptosystems. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 65–82, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer-Verlag, Berlin, Germany. 9, 10

[25] Shimon Even, Oded Goldreich, and Silvio Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9(1):35–67, 1996. 10

[26] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 60–77, Santa Barbara, CA, USA, August 20–24, 2006. Springer-Verlag, Berlin, Germany. 7

[27] Steven D. Galbraith and Wenbo Mao. Invisibility and anonymity of undeniable and confirmer signatures. In Marc Joye, editor, *Topics in Cryptology – CT-RSA 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 80–97, San Francisco, CA, USA, April 13–17, 2003. Springer-Verlag, Berlin, Germany. 9

[28] Craig Gentry and Zulfikar Ramzan. Identity-based aggregate signatures. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006: 9th International Conference on Theory and Practice of Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 257–273, New York, NY, USA, April 24–26, 2006. Springer-Verlag, Berlin, Germany. 12

[29] 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

[30] J. Herranz. Deterministic identity-based signatures for partial aggregation. *The Computer Journal*, 49 (3):322–330, 2006. 12

[31] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 143–154, Saragossa, Spain, May 12–16, 1996. Springer-Verlag, Berlin, Germany. 12

[32] Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic signature schemes. In Bart Preneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 244–262, San Jose, CA, USA, February 18–22, 2002. Springer-Verlag, Berlin, Germany. 12

[33] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 150–164, Santa Barbara, CA, USA, August 17–21, 1997. Springer-Verlag, Berlin, Germany. 7, 13, 15

[34] 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: 32nd International Colloquium on Automata, Languages and Programming*, volume 3580 of *Lecture Notes in Computer Science*, pages 434–445, Lisbon, Portugal, July 11–15, 2005. Springer-Verlag, Berlin, Germany. 12

[35] H. Krawczyk. Simple forward-secure signatures from any signature scheme. In *ACM Conference on Computer and Communications Security*, pages 108–115, 2000. 9

[36] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *ISOC Network and Distributed System Security Symposium – NDSS 2000*, San Diego, California, USA, February 2–4, 2000. The Internet Society. 12

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

[38] L. Lamport. Constructing digital signatures from a oneway function. Technical report, SRI International, October 1979. 12

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

[40] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters. Sequential aggregate signatures and multisignatures without random oracles. In *Eurocrypt'06*, 2006. To appear, available at `http://eprint.iacr.org/2006/096`. 6, 7, 12

[41] Anna Lysyanskaya, Silvio Micali, Leonid Reyzin, and Hovav Shacham. Sequential aggregate signatures from trapdoor permutations. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 74–90, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany. 12

[42] 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. 10

[43] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 80–99, New York, NY, USA, March 4–7, 2006. Springer-Verlag, Berlin, Germany. 7

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

[45] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565, Gold Coast, Australia, December 9–13, 2001. Springer-Verlag, Berlin, Germany. 12

[46] 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

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

[48] Adi Shamir and Yael Tauman. Improved online/offline signature schemes. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 355–367, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany. 10

[49] W. Susilo, F. Zhang, and Y. Mu. Identity-based strong designated verifier signature schemes. In *ACISP'04*, pages 313–324, 2004. 12

[50] W. Susilo, F. Zhang, and Y. Mu. On the security of nominative signatures. In *ACISP'05*, pages 329–335, 2005. 12

[51] J. Xu, Z. Zhang, and D. Feng. ID-based proxy signature using bilinear pairings. In *ISPA'05 International Workshop IADS*, pages 359–367, 2005. 10

[52] 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. 10

[53] F. Zhang and K. Kim. Efficient ID-based blind signature and proxy signature from bilinear pairings. In *ACISP'03*, pages 312–323, 2003. 7, 10

[54] Fangguo Zhang and Kwangjo Kim. ID-based blind signature and ring signature from pairings. In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 533–547, Queenstown, New Zealand, December 1–5, 2002. Springer-Verlag, Berlin, Germany. 7, 12

[55] Y. Zhou, Z. Cao, and Z. Chai. Identity based key insulated signature. In *ISPEC'06*, pages 226–234, 2006. 10