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

David Galindo[1], Javier Herranz[2] and Eike Kiltz[2]

[1] Institute for Computing and Information Sciences,
Radboud University, Nijmegen, The Netherlands
[2] Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands
d.galindo@cs.ru.nl, j.herranz@cwi.nl, kiltz@cwi.nl

**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.

## 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 [29]. 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 [6, 28].

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 [3], 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 [3]. In fact, in many papers the authors do not mention the generic approach from [3] and in spite of Shamir's work from more than two decades ago [29] 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 [5] (with its well-known limitations [9]).

### 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 (BS); verifiably encrypted signatures (VES); undeniable signatures (US); forward-secure signatures (FSS); strong key insulated signatures (SKIS); online/offline signatures (OOS); threshold signatures (TS); and aggregate signatures (AS).[3]

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 [3] and a formal security statement can be proved following basically verbatim the proofs given in [3]. 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.

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

[3] 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.

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 [3] 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.

## 2 Definitions

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.

The standard security notion for signature schemes in unforgeability against adaptively-chosen message attacks, which can be found in [19, 17].

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 [10]. 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 $\{0, 1\} \leftarrow \mathsf{IB\_S.Vfy}(mpk, id, m, sig)$.

The standard security notion for identity-based signature schemes is unforgeability against adaptively-chosen identity and message attacks, which can be found in [3, 17].

## 3 Generic Construction of Identity-based Signatures

We first outline the BNN generic transformation [3] from two standard signature schemes $\mathcal{S}$, $\mathcal{S}'$ into an identity-based signature scheme.

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 [3], 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

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

where $(pk_i, sk_i)$ is a random key-pair obtained by running $\mathsf{S'.KG}(1^k)$ and $sig_i \leftarrow \mathsf{S.Sign}(msk, id_i||pk_i)$. Here the signature $sig_i$ 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 $id_i$ an id-based signature for identity $id_i$ and message $m$ is defined as

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

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 $sig_i$ included in Eqn. (2) certifies the validity of $pk_i$.

Verification $\mathsf{IB\_S.Vfy}(mpk, sig)$: The user checks if the first signature from Eqn. (2) is valid with respect to $mpk$ and "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 [3] prove the following result:

**Theorem 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 to $\mathsf{IB\_S.Sign}$ and $\mathsf{IB\_S.Vfy}$) 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. Due to the lack of space we only provide details for identity-based VES, US, AS, and BS schemes. For the details on the remaining results we refer to the full verion of this paper [17]. Table 1 summarizes the practical impact of our results, i.e. it is shown which types $\mathcal{IB\_PS}$ of new identity-based signature schemes are implied by our general constructions. The existence of the identity-based signature schemes can be derived by the existence of the respective standard signature scheme [17].

| Signature type | Existence of identity-based signature schemes | | | |
|---|---|---|---|---|
| | at all ? | w/o random oracles? | w/o pairings? | general assumptions? |
| VES §3.1 | $\star$ | $\bigstar$ | $\bigstar$ | $\bigstar$ |
| BS §4 | $\star$/$\bigstar$[4] | $\bigstar$ | $\bigstar$ | $\bigstar$ |
| US §3.2 | $\star$ | $\bigstar$ | $\bigstar$ | $-$ |
| FSS [17] | $\bigstar$ | $\bigstar$ | $\bigstar$ | $\bigstar$ |
| SKIS [17] | $\star$ | $\bigstar$ | $\bigstar$ | $\bigstar$ |
| PS [17] | $\star$ | $\bigstar$ | $\bigstar$ | $\bigstar$ |
| OOS [17] | $\star$ | $\bigstar$ | $\bigstar$ | $\bigstar$ |
| Threshold [17] | $\star$ | $\bigstar$ | $\bigstar$ | $-$ |

**Table 1.** A summary of the practical implications of our results. Here "$\star$" means that a scheme was known before (with a formal proof), a "$\bigstar$" means that our construction gives the first such scheme, and a "$-$" means that no such scheme is known.

### 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 [7], one in the standard model in [25]. It was further noted in [25] 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 [20] where also a concrete security model was proposed. In contrast to [20], 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. More formally we can prove the following theorem:

**Theorem 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.*

Using our generic construction we get an IB-VES scheme based on any trapdoor one-way function [25], and a more efficient one using [7].

### 3.2 Undeniable Signatures

Undeniable signatures [12] (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 [14], 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.

It remains to describe the identity-based simulation algorithm $\mathsf{IB\_US.Sim}$ in terms of the 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 construction satisfies the simulatability property it is sufficient to require the scheme $\mathcal{US}$ to be anonymous in the sense of [16]. 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.** *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 [24]. This scheme uses bilinear pairings and it is proved secure in the random oracle model. We stress that the security model in [24] seems to be incomplete, as the authors do not consider simulatability.

In [16], 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 [23]; it does not employ bilinear pairings, but the disavowal protocol is quite inefficient. Using these anonymous US schemes [16, 23], 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.

### 3.3 Aggregate Signatures

The idea of an aggregate signature scheme [7] is to combine $n$ signatures on $n$ different messages, signed by $n$ (possibly different) signers, in order to obtain a single aggregate signature $AgSig$ which provides the same certainty than the $n$ initial signatures. The main goal in the design of such protocols is that the length of $AgSig$ 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 $AgSig$.

In the identity-based framework, the only proposal which achieves constant-length aggregation is that of [18]; 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 [25] for the PKI-based scenario). With respect to non-interactive aggregate signatures in the identity-based setting, the most efficient proposal is from [21], that 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 $\{(id_i, sig_{msk}(id_i||pk_i), pk_i, m_i, sig_i\}_{1\leq i\leq n}$, where $sig_i$ and $sig_{sk_i}(m_i)$ are signatures resulting from scheme $\mathcal{S}$, and can therefore be aggregated into a PKI-based aggregate signature $AgSig$, of constant-length. Then the final identity-based aggregate signature would be $IBAgSig = (Ag\_Sig, pk_1, \ldots, pk_n)$. This aggregate signature, along with the $n$ messages and the $n$ identities, is sufficient to verify the correctness of the $n$ signatures. Therefore the length of the identity-based aggregate signature $IBAgSig$ is linear with respect to the number of different signers.

### 3.4 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; designated verifier signatures; confirmer signatures; chameleon signatures; and nominative signatures.

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, homomorphic signatures, etc.) as well.

## 4 Generic Construction of ID-Based Blind Signatures

In this section we consider in more detail the generic construction in the case of blind signature schemes. In blind signature (BS) schemes [11] 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 [22, 27]. 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 [31, 30]. They employ bilinear pairings, but their security is not formally analyzed. Subsequent schemes were proposed in [13] but security is only provided in a weaker model (i.e. against sequential adversaries).

The main result of this section can be stated as follows.

**Theorem 4.** *If $\mathscr{S}$ is a* strongly secure *standard signature scheme and $\mathscr{PS}$ is a secure (partially) blind signature scheme then a secure identity-based (partially) blind signature scheme $\mathscr{IB\_PS}$ 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 [8, 26, 15]), we obtain IB-BS schemes which do not employ bilinear pairings [4], and we obtain IB-BS schemes from any one-way trapdoor permutation [22, 15].

We now formally prove Theorem 4. First we recall the basic definitions of PKI-based and identity-based blind signature schemes, then we explain and analyze our construction and prove its blindness. Due to lack of space, we included all details (definitions and analysis) related to the unforgeability property in the full version of this paper [17].

### 4.1 Blind Signature Schemes

Blind signature schemes were introduced in [11] 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 $b \in \{0, 1\}$ at random 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 $k$).

### 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. Due to lack of space, we focus only on the blindness property.

Blindness. Blindness of an identity-based blind signature scheme is defined by a game played between a challenger and an adversary. This adversary $\mathcal{B}_{\rm IB}$ 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 $\mathcal{B}_{\rm IB}$. The master secret key $msk$ is kept secret by the challenger. The adversary $\mathcal{B}_{\rm IB}$ 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 $\mathcal{B}_{\rm IB}$. If the same identity is asked again, the same value $sk[id_i]$ must be returned by the challenger. At some point, the adversary $\mathcal{B}_{\rm IB}$ chooses an identity $id^*$ and two messages $m_0, m_1$, and sends these values to the challenger. The challenger chooses $b \in \{0, 1\}$ at random 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 $\mathcal{B}_{\rm IB}$ 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 $\mathcal{B}_{\rm IB}$ 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 $\mathbf{Adv}^{\text{ib-blind}}_{\mathcal{IB\_BS},\mathcal{B}_{\rm IB}}(k) = |\Pr[b' = b] - 1/2|$. A scheme $\mathcal{IB\_BS}$ has the blindness property if, for all PPT adversaries $\mathcal{B}_{\rm IB}$, $\mathbf{Adv}^{\text{ib-blind}}_{\mathcal{IB\_BS},\mathcal{B}_{\rm IB}}(k)$ is a negligible function (with respect to the security parameter $k$).

### 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 ID-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 $m$ 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 in parallel: $\{0,1\} \leftarrow \mathsf{S.Vfy}(mpk, id_i||pk_i, sig_{msk}(id_i||pk_i))$ 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 blindness property, assuming that the schemes $\mathcal{S}$ and $\mathcal{BS}$ employed as primitives are secure. The detailed analysis of the unforgeability property can be found in [17].

**Theorem 5.** *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.* Assume there exists a successful adversary $\mathcal{B}_{\mathrm{IB}}$ against the blindness of the scheme $\mathcal{IB\_BS}$. We show that 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 $\mathcal{BS}$. 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 $\mathcal{B}_{\mathrm{IB}}$ by providing it with $mpk = PK$.

**Secret key queries.** Adversary $\mathcal{B}_{\mathrm{IB}}$ is allowed to make secret key queries for identities $id_i$ of its choice. To answer a 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 of 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 $\mathcal{B}_{\mathrm{IB}}$ the consistent answer $sk[id_i] = (sk_i, pk_i, sig_i)$. We store all this information in a table. If the same identity is asked twice by $\mathcal{B}_{\mathrm{IB}}$, then the same secret key is given as answer.

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

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

The first two steps in the two executions of the interactive signing protocol of $\mathcal{IB\_BS}$ run between $\mathcal{B}_{\mathrm{IB}}$ and us are identical. Hence distinguishing between the two executions of $\mathsf{IB\_BS.Sign}$ is equivalent to distinguishing between the two executions of $\mathsf{BS.Sign}$. This completes the proof. $\square$

We stress that the signature scheme $\mathcal{S}$ really has to be *strongly* unforgeable; otherwise a 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 6.** *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.*

The proof of Theorem 6 can be found in [17]. Theorems 5 and 6 imply Theorem 4.

## 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. M. Abe and E. Fujisaki. How to date blind signatures. *ASIACRYPT'96*, volume 1163 of *LNCS*, pages 244–251, 1996.
2. M. Abe and T. Okamoto. Provably secure partially blind signatures. *CRYPTO 2000*, volume 1880 of *LNCS*, pages 271–286, 2000.
3. M. Bellare, C. Namprempre and G. Neven. Security proofs for identity-based identification and signature schemes. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 268–286, 2004.
4. M. Bellare, C. Namprempre, D. Pointcheval and M. Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, 2003.
5. M. Bellare and P. Rogaway. Optimal asymmetric encryption. *EUROCRYPT'94*, volume 950 of *LNCS*, pages 92–111, 1994.
6. D. Boneh and M.K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.
7. D. Boneh, C. Gentry, B. Lynn and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432, 2003.

8. J. Camenisch, M. Koprowski and B. Warinschi. Efficient blind signatures without random oracles. *SCN 04*, volume 3352 of *LNCS*, pages 134–148, 2004.
9. R. Canetti, O. Goldreich and S. Halevi. The random oracle methodology, revisited. *30th ACM STOC*, pages 209–218, 1998.
10. J.C. Cha and J.H. Cheon. An identity-based signature from gap Diffie-Hellman groups. *PKC 2003*, volume 2567 of *LNCS*, pages 18–30, 2003.
11. D. Chaum. Blind signatures for untraceable payments. *CRYPTO'82*, pages 199–203, USA, 1983.
12. D. Chaum and H. Van Antwerpen. Undeniable signatures. *CRYPTO'89*, volume 435 of *LNCS*, pages 212–216, 1990.
13. S.M. Chow, L.K. Hui, S.M. Yiu and K.P. Chow. Two improved partially blind signature schemes from bilinear pairings. *ACISP 2005*, pages 316–325, 2005.
14. I. Damgard and T.P. Pedersen. New convertible undeniable signature schemes. *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 372–386, 1996.
15. M. Fischlin. Round-optimal composable blind signatures in the common reference string model. *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77, 2006.
16. S.D. Galbraith and W. Mao. Invisibility and anonymity of undeniable and confirmer signatures. *CT-RSA 2003*, volume 2612 of *LNCS*, pages 80–97, 2003.
17. D. Galindo, J. Herranz. and E. Kiltz. On the generic construction of identity-based signatures with additional properties. Cryptology ePrint Archive, Report 2006/296, 2006. Full version of this paper, `http://eprint.iacr.org/`.
18. C. Gentry and Z. Ramzan. Identity-based aggregate signatures. *PKC 2006*, volume 3958 of *LNCS*, pages 257–273, 2006.
19. S. Goldwasser, S. Micali and R.L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.
20. C. Gu and Y. Zhu. An id-based verifiable encrypted signature scheme based on Hess's scheme. *CISC'05*, pages 42–52, 2005.
21. J. Herranz. Deterministic identity-based signatures for partial aggregation. *The Computer Journal*, 49 (3):322–330, 2006.
22. A. Juels, M. Luby and R. Ostrovsky. Security of blind digital signatures (extended abstract). *CRYPTO'97*, volume 1294 of *LNCS*, pages 150–164, 1997.
23. F. Laguillaumie and D. Vergnaud. Short undeniable signatures without random oracles: the missing link. *Indocrypt'05*, volume 3797 of *LNCS*, pages 283–296, 2005.
24. B. Libert and J.J. Quisquater. Identity based undeniable signatures. *CT-RSA 2004*, volume 2964 of *LNCS*, pages 112–125, 2004.
25. S. Lu, R. Ostrovsky, A. Sahai, H. Shacham and B. Waters. Sequential aggregate signatures and multisignatures without random oracles. *EUROCRYPT'06*, 2006.
26. T. Okamoto. Efficient blind and partially blind signatures without random oracles. *TCC 2006*, volume 3876 of *LNCS*, pages 80–99, 2006.
27. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
28. R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing over elliptic curve (in Japanese). *SCIS 2001*, Jan 2001.
29. A. Shamir. Identity-based cryptosystems and signature schemes. *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53, 1985.
30. F. Zhang and K. Kim. Efficient ID-based blind signature and proxy signature from bilinear pairings. In *ACISP'03*, pages 312–323, 2003.
31. F. Zhang and K. Kim. ID-based blind signature and ring signature from pairings. *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 533–547, 2002.