Multi-key Hierarchical Identity-Based Signatures*

Hoon Wei Lim and Kenneth G. Paterson

Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
{h.lim, kenny.paterson}@rhul.ac.uk

Abstract. We motivate and investigate a new cryptographic primitive that we call *multi-key hierarchical identity-based signatures* (multi-key HIBS). Using this primitive, a user is able to prove possession of a set of identity-based private keys associated with nodes at arbitrary levels of a hierarchy when signing a message. Our primitive is related to, but distinct from, the notions of identity-based multi-signatures and aggregate signatures. We develop a security model for multi-key HIBS. We then present and prove secure an efficient multi-key HIBS scheme that is based on the Gentry-Silverberg hierarchical identity-based signature scheme.

1 Introduction

Research in identity-based cryptography (IBC) [32] has proliferated in recent years since the discovery of novel pairing-based key agreement protocols due to Sakai *et al.* [30] and Joux [23], and the seminal work of Boneh and Franklin [8] giving the first secure and practical identity-based encryption (IBE) scheme.

Several proposals for identity-based signature (IBS) schemes, such as [12, 21, 26], followed quickly after Boneh and Franklin's publication [8]. More recent proposals for IBS can be found in [4, 5, 17, 27], for example. However, IBS schemes are arguably less interesting than IBE schemes. They suffer from the key escrow property that is inherent in IBC, which makes non-repudiation of these signatures more difficult to achieve. Moreover, there is a generic construction that creates an IBS scheme from two instantiations of any (normal) public key signature scheme [5]. This is related to the fact that, with a normal,

* The first author's research was supported by the EPSRC under grant EP/D051878/1. The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the authors' views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

certificate-based signature, all information necessary for verification can be included along with the basic signature to create a self-contained package. Nevertheless, many real world applications of identity-based signatures have been proposed in the literature, such as lightweight email authentication [1], and biometric authentication [11]. Attractive variants of identity-based signatures, for example identity-based multi-signatures (IBMS) [6, 18], identity-based aggregate signatures (IBAS) [18] and identity-based threshold signatures (IBTS) [2] have also been proposed. These applications and primitives offer other advantages over the use of certificate-based signatures, such as significant savings in communication bandwidth. Moreover, IBS schemes can be deployed in identity-based infrastructures so as to provide a complement to IBE schemes whilst making use of the same infrastructural components and code base.

In hierarchical identity-based cryptography (HIBC) [7, 10, 19, 22], multiple levels of private key generators (PKGs) and users form a tree-like structure which mimics the existing hierarchical PKI model. A user at any level of the tree can encrypt or sign a message targeting any intended recipient at any level, using only a set of shared cryptographic system parameters published by the root PKG. A hierarchical identity-based signature (HIBS, or HIDS in [19]) scheme is then the analogue of an IBS in the hierarchical setting. One interesting application of the hierarchical approach is that it can be used for delegation in a natural way: an act of delegation can be carried out by the issuance of a private key by a node in the tree to its child. A delegatee's delegated credential can be checked by performing only one signature verification, regardless of the length of the delegation chain [14].

In this paper, we introduce, motivate and develop a new cryptographic primitive called *multi-key hierarchical identity-based signatures* (multi-key HIBS). The essence of this new primitive is as follows. We operate in the setting of HIBC, but we assume each user owns multiple identifiers and thus possesses a set of corresponding private or signing keys. These identifiers may be located at arbitrary positions in the hierarchy. When a user generates a signature on a message, he uses a subset of his private keys for signing. Informally, then, a multi-key HIBS scheme is used to produce a single signature on a selected message using a set of signing keys. We discuss two potential applications for multi-key HIBS next, and then explain the relationship between our new primitive and related cryptographic concepts.

Motivating Examples. Our first example is related to access control in open distributed environments, such as grid computing systems [15].

There has been a recent trend in research on access control in which user authentication and access control are achieved in a unified way using cryptographic techniques, for example, policy-based encryption [3, 33], attribute-based encryption [20, 28, 29] and role signatures [13]. Particularly, the use of roles as identifiers (in the context of HIBC), or role identifiers and their associated signatures for access control in [13] has inspired the work presented in this paper.

Role-based access control (RBAC) [31] is well-known for being more scalable than access control based on user identities. Users are granted membership into roles based on their competencies and responsibilities in their organisation. The operations that a user is permitted to perform are based on the user's role. Role signatures, as proposed in [13], are designed to address the problems of inter-domain principal mapping and authentication of user credentials simultaneously.

It is observed in [13] that a HIBS scheme can be used naturally for RBAC within a distributed computing environment, such as a grid system, which has a hierarchical structure. A verification key can be defined by a role identifier defined within a hierarchical namespace. User authentication and access control is unified, and credential verification is rendered trivial. An authorisation service is required to only verify a single signature, produced from the signing key associated to the role identifier, to both confirm that the user is an authenticated member of an organisation and occupies a particular role within that organisation.

Suppose, for example, that Alice has been given the roles r_1= `lecturer`, r_2= `professor` and r_3= `IEEE member`. Let Alice be an employee of university X, which in turn is part of the Open University, OU. Then, a role identifier, defined in a hierarchical namespace, can be of the form "OU, X, r_i", where $1 \leq i \leq 3$. In order to abide by the principle of least privilege, Alice should be given the options of selecting which role(s) she wishes to use when accessing some resources.

Now assume that Alice wants to access some restricted digital documents stored in the Open Library using roles r_2 and r_3. In principle, Alice must then sign an access request using private keys S_2 and S_3 which correspond to role identifiers "OU, X, r_2" and "OU, X, r_3", respectively. Our multi-key HIBS primitive allows the access request to be signed using the pair of private keys S_2 and S_3, while verification of the signed request uses the pair of associated role identifiers.

Our second example is related to mobile ad hoc network (MANET) applications. Here, conserving computation and bandwidth are at a premium, so the use of identity-based techniques is attractive [24]. However, it is commonly assumed in the MANET setting that there are no trusted entities that can play the role of PKGs (or trusted authorities). Moreover, nodes may be compromised or unavailable. Therefore, it is desirable to distribute the functions of PKGs across multiple network nodes. In the most general setting, the distributed identity-based key infrastructure that results may even be hierarchical in nature. Hence a user in the network may receive multiple private keys from different nodes, yet wish to efficiently demonstrate possession of some or all of these keys when creating a signature. For example, this may be required as part of an identity-based key exchange protocol used to create a session key shared between two network nodes. In this situation, a multi-key HIBS scheme is just what is needed.

Related Concepts. Our concept of multi-key HIBS is closely related to the concepts of identity-based multi-signatures (IBMS) [6, 18] and identity-based aggregate signatures (IBAS) [18].

In multi-signatures, a set of users all sign the same message and collectively produce a single signature. An IBMS scheme may be either non-interactive or interactive; this property is related to whether or not individual signatures can be combined to produce a multi-signature by an outside agency. An example of the former type of IBMS scheme can be found in [18], and of the latter in [6]. In an aggregate signature scheme [9], each user can sign a different message; subsequently a set of signatures can be turned into a single signature via an aggregation process which may be executed by any entity. The first efficient constructions of IBAS were obtained in [18], but under the restriction that signers coordinate some state before signing. A non-interactive IBMS scheme can be obtained from any IBAS scheme.

It is easy to see that the basic functionality needed in our motivating examples can be obtained by just using an IBMS scheme (of either type), and hence also by using an IBAS scheme. However, this assumes that hierarchical IBMS or IBAS schemes with the required degree of flexibility can be obtained; currently available schemes operate only in the basic IBC setting having a single level in the hierarchy. Moreover, the fact that all signing keys are possessed by a single party in the multi-key HIBS setting opens up the possibility of making significant efficiency gains (particularly in the signing algorithm) by using a scheme designed from scratch. Indeed, we show here that such gains can be realised in practice. This efficiency gain highlights one of the main differences between our new concept of multi-key HIBS and the existing notions of IBMS/IBAS.

A second main difference arises when we consider the strength of security model that we introduce here for multi-key HIBS. Our model allows the adversary to obtain all but one of the signing keys involved in its output forgery and gives the adversary full access to a signing oracle. This is comparable to the multi-signer security model introduced for interactive IBMS in [6], which seems to be the strongest security model for IBMS introduced to date. However, our model is much stronger than the model for non-interactive IBMS given in [18]: the restrictions on adversarial behaviour in [18] are such that no identity may be involved in two different queries to the signing oracle (on different messages). This seems very limited in comparison to our model.[1]

Yet a third difference arises from the hierarchical and flexible nature of our multi-key HIBS definition and its concrete instantiations: in contrast to existing definitions and realisations of IBMS/IBAS, we work in a fully hierarchical setting, and allow the identifiers involved to be located at arbitrary positions in the hierarchy. A special case of our instantiations yields an efficient multi-key IBS (one-level) scheme.

Our new primitive is also loosely related to the concept of IBTS [2]. Generally speaking, a (t, n)-IBTS scheme allows a set of t (the threshold) out of n parties to first compute individual signature shares; these shares are then combined into one single signature. However, IBTS schemes are interactive, and generally much less efficient than our multi-key HIBS schemes. Another related idea is Boneh and Franklin's technique [8] of distributing an identity-based master secret over multiple PKGs using Shamir secret sharing. Boneh and Franklin observed that

[1] Note that the paper [18] does not explicitly include a security model for IBMS, though it does contain a concrete IBMS scheme and claim of security for that scheme. However, one can infer the IBMS model intended by the authors of [18] by specialising their state-oriented IBAS model to the stateless IBMS setting.

the master secret in their IBE scheme can be distributed in a t-out-of-n fashion by giving each of the n PKGs one share of the master secret. A user can then construct his private key by obtaining shares of his private key from each of the t chosen PKGs and then combining these using the appropriate Lagrange coefficients. Our new primitive can be seen as being related to a special case of this idea in which $t = n$. However, we deal with signatures and not encryption, we work in the hierarchical setting and not the basic identity-based setting, and our multi-key HIBS schemes allow signatures to be constructed in a more flexible way from private keys corresponding to nodes at arbitrary positions in the hierarchy.

Our Results. We provide a formal definition of multi-key HIBS and an appropriate security model for this primitive. We then present a concrete and provably secure multi-key HIBS scheme. Our scheme is provably secure in the Random Oracle Model (ROM), assuming the hardness of the computational Diffie-Hellman problem in groups equipped with a pairing. We consider the strongest possible security setting, by allowing the adversary arbitrary access to signing and key extraction oracles (subject to obvious limitations needed to prevent the adversary from trivially winning the security games).

We then show how the complexity of the verification algorithm of our scheme can be reduced in special cases that depend on the relative positions of nodes associated with the signing keys used to generate a signature. We also demonstrate that, in situations where multi-key HIBS are applicable, our concrete scheme has a much more efficient signing procedure than the currently best available IBMS schemes [6, 18]. Since these IBMS schemes are only one-level (i.e. non-hierarchical), we make the comparison in the one-level case, even though our concrete scheme is far more flexible.

In the next section, we provide definitions for our new multi-key HIBS primitive. In Section 3, we propose a concrete multi-key HIBS scheme. Its security analysis is given in Section 4. Section 5 explains how to optimise our multi-key HIBS scheme in specific cases. Then, in Section 6, we compare the performance of our concrete multi-key HIBS scheme with the best IBMS schemes available in the literature. In Section 7, we present our conclusions and some open problems suggested by our work.

2 Definitions

2.1 Pairings and Associated Problems

Our scheme makes use of pairings. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups where $|\mathbb{G}| = |\mathbb{G}_T| = q$, a large prime. Then an admissible pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ has the following properties:

- *Bilinear*: Given $P, Q, R \in \mathbb{G}$, we have

$$e(P, Q + R) = e(P, Q) \cdot e(P, R) \text{ and } e(P + Q, R) = e(P, R) \cdot e(Q, R).$$

Hence, for any $a, b \in \mathbb{Z}_q^*$,

$$e(aP, bQ) = e(abP, Q) = e(P, abQ) = e(aP, Q)^b = e(P, Q)^{ab}.$$

– *Non-degenerate*: There exists $P \in \mathbb{G}$ such that $e(P, P) \neq 1$.
– *Computable*: If $P, Q \in \mathbb{G}$, then $e(P, Q)$ can be efficiently computed.

Further details on pairings and their implementations using elliptic curves and the Weil, Tate or related pairings can be found in [16]. Our scheme is easily adapted to cope with pairings $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, but we focus here on the simpler case for ease of presentation.

Let λ denote the security parameter. We say that a randomised algorithm $\mathcal{G}$ is a Bilinear Diffie-Hellman (BDH) parameter generator if: (i) $\mathcal{G}$ takes $\lambda > 0$ as input; (ii) $\mathcal{G}$ runs in time polynomial in λ; and (iii) $\mathcal{G}$ outputs the description of two groups $\mathbb{G}$ and $\mathbb{G}_T$ of the same prime order q and the description of an admissible pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. The security of the schemes presented in this paper is based on the assumed hardness of the computational Diffie-Hellman (CDH) problem in groups $\mathbb{G}$ produced by such a generator. This problem is defined as follows:

Definition 1 *(CDH in a group equipped with a pairing) Given $\mathbb{G}$, $\mathbb{G}_T$ of the same prime order q, an admissible pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and $\langle P, aP, bP \rangle \in \mathbb{G}$ for some random $P \in \mathbb{G}$ and randomly chosen $a, b \in \mathbb{Z}_q^*$, the CDH problem in $\mathbb{G}$ is to compute $abP \in \mathbb{G}$.*

An algorithm $\mathcal{A}$ is said to have advantage ϵ in solving the CDH problem in $\mathbb{G}$ if

$$\Pr\left[\mathcal{A}(P, aP, bP) = abP \geq \epsilon\right]$$

where the probability is over the random choice of P in $\mathbb{G}$, the random scalars a and b in $\mathbb{Z}_q^*$, and the random bits used by $\mathcal{A}$. (Here we suppress the additional inputs to $\mathcal{A}$.)

2.2 Multi-key HIBS Scheme

Hierarchical identity-based cryptography involves nodes arranged in a tree structure, with each node having an identifier. The identifier of an entity is the concatenation of the node identifiers in the path from the root to the node associated with that entity. Assuming the root Private Key Generator (PKG) is located at level 0, then the identifier of an entity at level t is the concatenation of node identifiers $id_1, \ldots, id_t$ in which each $id_i \in \{0, 1\}^*$. We denote the concatenation of node identifiers $id_1, \ldots, id_t$ by identifier ID_t. The entity with identifier ID_t has an ancestor at level i with identifier $\mathrm{ID}_i = id_1, \ldots, id_i$ for $1 \leq i < t$; this entity's parent is the node with identifier ID_{t-1} and its children are all the nodes with identifiers of the form $\mathrm{ID}_{t+1} = id_1, \ldots, id_t, id_{t+1}$. We use $(P_t,\, S_t)$ to represent the public/private key pair of the entity with identifier ID_t.

A multi-key hierarchical identity-based signature (multi-key HIBS) scheme can be regarded as an extended HIBS scheme. It produces signatures in the hierarchical identity-based setting using sets of signing keys. A multi-key HIBS scheme is specified by the following algorithms:

Root Setup**:** This algorithm is performed by the root PKG. It generates the system parameters and a master secret on input a security parameter λ. The system parameters, which include a description of the message space $\mathcal{M}$ and the signature space $\mathcal{S}$, will be made publicly available to all entities (PKGs or users). However, the master secret is known only to the root PKG.

Lower-level Setup**:** All entities at lower levels must obtain the system parameters generated by the root PKG. This algorithm allows a lower-level PKG to establish a secret value to be used to issue private keys to its children.

Extract**:** This algorithm is performed by a PKG (root or lower-level PKG) with identifier ID_t to compute a private key S_{t+1} for any of its children using the system parameters and its private key (and any other secret information).

Sign**:** Given a set $\mathsf{SK} = \{S^j_{t_j} : 1 \le j \le n\}$ of signing (private) keys, a message $M \in \mathcal{M}$, and the system parameters, this algorithm outputs a signature $\sigma \in \mathcal{S}$. Here t_j denotes the level of the j-th signing key in the set SK.

Verify**:** Given a signature $\sigma \in \mathcal{S}$, a set $\mathsf{ID} = \{\mathrm{ID}^j_{t_j} : 1 \le j \le n\}$ of identifiers, a message $M \in \mathcal{M}$, and the system parameters, this algorithm outputs `valid` or `invalid`.

We have the obvious consistency requirement: if σ is output by Sign on input a set SK of private keys and message M, then Verify outputs `valid` when given input σ, the set ID of identifiers corresponding to SK, and M.

We remark that the first three algorithms specified above are identical to those of a HIBS scheme. Moreover, a multi-key HIBS scheme, when used with a single signing key, is essentially just a normal HIBS scheme. We also remark that while we have chosen to work with sets of identifiers and private keys, the definitions (as well as the security model and concrete construction to follow) can easily be adapted to deal with ordered lists of identifiers and keys.

2.3 Security Model

The security model for a multi-key HIBS scheme is based on the following game between a challenger and an adversary that extends the normal HIBS security game [19]:

1. The challenger runs the Root Setup algorithm of the multi-key HIBS scheme. The resulting system parameters are given to the adversary. The master secret, however, is kept secret by the challenger.
2. The adversary adaptively issues queries to the challenger. Each query can be one of the following:

– Extract: The adversary can ask for the private key associated with any identifier $\mathrm{ID}^j_{t_j} = id^j_1, \ldots, id^j_{t_j}$. The challenger responds by running the EXTRACT algorithm to generate a private key $S^j_{t_j}$, which is then returned to the adversary.
– Sign: The adversary can ask for the signature associated with a set ID of identifiers on a message M of its choice. The challenger responds by running the SIGN algorithm using as input a set of signing keys SK corresponding to the set of identifiers ID, message M, and the system parameters. The resulting signature σ is returned to the adversary.

3. The adversary outputs a string σ^*, a set of target identifiers ID^*, and a message M^*. The adversary wins the game if the following are all true:
   – $\mathrm{VERIFY}(\sigma^*, \mathsf{ID}^*, M^*) = \mathtt{valid}$;
   – There exists an identifier $\mathrm{ID}' \in \mathsf{ID}^*$ for which the adversary has not made an Extract query on ID' or any of its ancestors;
   – The adversary has not made a Sign query on input ID^*, M^*.

The advantage of an adversary $\mathcal{A}$ in the above game is defined to be

$$\mathrm{Adv}_{\mathcal{A}} = \Pr[\mathcal{A}\ wins]$$

where the probability is taken over all coin tosses made by the challenger and the adversary.

Note that the usual security model for HIBS is recovered in the case where all queries involve a single identifier.

3 Construction

We now present a concrete multi-key HIBS scheme, which is adapted from the Gentry-Silverberg HIBS scheme [19]. We will show in Section 5 how the cost of the verification algorithm of this scheme can be reduced in specific situations.

ROOT SETUP: The root PKG:

1. runs $\mathcal{G}$ on input λ to generate $\mathbb{G}$ and $\mathbb{G}_T$ of prime order q and an admissible pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$;
2. chooses a generator $P_0 \in \mathbb{G}$;
3. picks a random value $s_0 \in \mathbb{Z}_q^*$ and sets $Q_0 = s_0 P_0$;
4. selects cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}$ and $H_2 : \{0,1\}^* \to \mathbb{G}$.

The root PKG's master secret is s_0 and the system parameters are $\langle \mathbb{G}, \mathbb{G}_T, e, q, P_0, Q_0, H_1, H_2 \rangle$. The message space is $\mathcal{M} = \{0,1\}^*$ and the signature space is $\mathcal{S} = \bigcup_{t \geq 0} \mathbb{G}^{t+1}$.

LOWER-LEVEL SETUP: A lower-level entity (lower-level PKG or user) at level $t \geq 1$ picks a random secret $s_t \in \mathbb{Z}_q^*$.

EXTRACT: For an entity with identifier $\mathrm{ID}_t = id_1, \ldots, id_t$, the entity's parent:

1. computes $P_t = H_1(\mathrm{ID}_t) \in \mathbb{G}$;

2. sets $S_t = \sum_{i=1}^{t} s_{i-1}P_i = S_{t-1} + s_{t-1}P_t$;
3. defines $Q_i = s_i P_0$ for $1 \le i \le t-1$.

The private key $\langle S_t, Q_1, \ldots, Q_{t-1} \rangle$ is given to the entity by its parent. Note that up to this point, our scheme is identical to the Gentry-Silverberg HIBS scheme.

Sign: Given any $n \ge 1$ and a set $\mathsf{SK} = \{\langle S^j_{t_j}, Q^j_1, \ldots, Q^j_{t_j-1} \rangle : 1 \le j \le n\}$ of n private keys associated with a set $\mathsf{ID} = \{\mathrm{ID}^j_{t_j} : 1 \le j \le n\}$ of identifiers, and a message M, the signer:

1. chooses a secret value $s_\varphi \in \mathbb{Z}_q^*$;
2. computes $P_M = H_2(\mathrm{ID}^1_{t_1}, \ldots, \mathrm{ID}^n_{t_n}, M)$ (where we assume the identifiers are first placed in lexicographic order if necessary);
3. calculates

$$\varphi = \sum_{j=1}^{n} S^j_{t_j} + s_\varphi P_M \quad \text{and} \quad Q_\varphi = s_\varphi P_0.$$

The algorithm outputs the signature $\sigma = \langle \varphi, \mathsf{Q}, Q_\varphi \rangle$, where $\mathsf{Q} = \{Q^j_i : 1 \le i \le t_j - 1, 1 \le j \le n\}$.

Verify: Given $\sigma = \langle \varphi, \mathsf{Q}, Q_\varphi \rangle$, a set of identifiers $\mathsf{ID} = \{\mathrm{ID}^1_{t_1}, \ldots, \mathrm{ID}^n_{t_n}\}$ and a message M, the verifier:

1. computes $P^j_i = H_1(\mathrm{ID}^j_i)$ for $1 \le i \le t_j$ and $1 \le j \le n$;
2. computes $P_M = H_2(\mathrm{ID}^1_{t_1}, \ldots, \mathrm{ID}^n_{t_n}, M)$ (first arranging the identifiers lexicographically if they are not already in this order);
3. checks if $e(P_0, \varphi)$ is equal to

$$\left(\prod_{j=1}^{n} \prod_{i=1}^{t_j} e(Q^j_{i-1}, P^j_i) \right) \cdot e(Q_\varphi, P_M),$$

outputting `valid` if this equation holds, and `invalid` otherwise.

It is not hard to verify that this scheme is consistent. We remark that two different hash functions H_1 and H_2 are used in the construction to make the security proof easier to follow; a similar scheme using only one hash function is easily constructed. However, in such a scheme we need to take care to distinguish the different types of strings input to the hash. Even with two hash functions, we need to assume that any input to H_2 can be uniquely parsed as a list of identifiers concatenated with a (possibly empty) message. This prevents trivial re-encoding attacks against the scheme. This requirement can be met in a number of ways. For example, we can assume that there is a special separating symbol which marks the end of the identifier portion and which does not appear in any identifier, or we can encode each identity bit id as $0id$ and each message bit m as $1m$. We note that this parsing requirement is not explicitly specified for the HIBS scheme of [19], meaning that (trivially preventable) re-encoding attacks *are* possible against that scheme.

4 Security Analysis

In this section, we establish the security of the concrete multi-key HIBS scheme of Section 3. For simplicity of presentation, we focus here on the case in which all identifiers lie at level 1 in the hierarchy (in which case they have the root PKG as their parent). Thus our proof is actually for the multi-key IBS scheme that arises as a special case of our more general hierarchical scheme. We slightly extend the techniques of [8] to directly relate this scheme's security to the hardness of the CDH problem in groups $\mathbb{G}$ equipped with a pairing. After the proof, we sketch how our approach might be extended to handle the more general case by using additional techniques from [19].

Theorem 2 *Suppose that $\mathcal{A}$ is a forger against our multi-key IBS scheme that has success probability ϵ. Let hash functions H_1 and H_2 be modeled as random oracles. Then there is an algorithm $\mathcal{B}$ which solves the CDH problem in groups $\mathbb{G}$ equipped with a pairing, with advantage at least*

$$\epsilon/(\mathbf{e} \cdot q_{H_1} \cdot q_{H_2})$$

and which has running time $\mathcal{O}(time(\mathcal{A}))$. Here, q_{H_1} is the maximum number of H_1 queries made by $\mathcal{A}$ during its attack, q_{H_2} is the maximum number of H_2 queries made by $\mathcal{A}$, and $\mathbf{e}$ denotes the base of natural logarithms.

Proof: Algorithm $\mathcal{B}$ is given as input an admissible pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, where $|\mathbb{G}| = |\mathbb{G}_T| = q$, and an instance $\langle P, aP, bP \rangle$ of the CDH problem in $\mathbb{G}$, generated by $\mathcal{G}$. It will interact with algorithm $\mathcal{A}$, as follows, in an attempt to compute abP.

Setup: Algorithm $\mathcal{B}$ sets the system parameters of the root PKG to be $\langle \mathbb{G}, \mathbb{G}_T, e, P_0 = P, Q_0 = aP, H_1, H_2 \rangle$, so that the master secret is the unknown value a. The system parameters are then forwarded to $\mathcal{A}$. Here, H_1 and H_2 are random oracles controlled by $\mathcal{B}$. Additionally, $\mathcal{B}$ randomly selects $c \in \{1, \ldots, q_{H_1}\}$, where q_{H_1} is the maximum number of queries to H_1 made by $\mathcal{A}$.

H_1 queries: $\mathcal{A}$ can query H_1 on any input $\mathrm{ID} \in \{0,1\}^*$. In responding to these queries, $\mathcal{B}$ maintains a list Λ_{H_1} containing tuples of the form $\langle \mathrm{ID}^i, r_i, R_i \rangle$. The list Λ_{H_1} is initially empty. If ID already appears in the list Λ_{H_1}, in position i say, then $\mathcal{B}$ responds with R_i. Otherwise, $\mathcal{B}$ responds as follows:

1. If this is the c-th distinct query to H_1, then set $\mathrm{ID}' = \mathrm{ID}$;
2. Select r at random from $\mathbb{Z}_q^*$;
3. If $\mathrm{ID} = \mathrm{ID}'$, then add $\langle \mathrm{ID}, r, R = bP + rP \rangle$ to the list Λ_{H_1}.
4. Otherwise, add $\langle \mathrm{ID}, r, R = rP \rangle$ to the list Λ_{H_1}.
5. Output R as the response to the H_1 query.

Notice that the output R is always chosen uniformly in $\mathbb{G}$, as required.

H_2 queries: $\mathcal{A}$ can query H_2 on any list of identifiers $\mathrm{ID}^1, \ldots, \mathrm{ID}^n$ and any message M. To respond to these queries, $\mathcal{B}$ maintains a list Λ_{H_2} of tuples $\langle(\mathrm{ID}_i^1, \ldots, \mathrm{ID}_i^n), M_i, z_i, f_i, d_i, T_i\rangle$. The list is initially empty, and $\mathcal{B}$ uses the list to reply consistently with values T_i to $\mathcal{A}$'s queries. $\mathcal{B}$ first queries H_1 on each identifier ID^j and sets $\mathsf{ID} = \{\mathrm{ID}^1, \ldots, \mathrm{ID}^n\}$. Then $\mathcal{B}$ responds as follows:

1. If $\mathrm{ID}' \in \mathsf{ID}$, then generate a coin $f \in \{0, 1\}$ where $\Pr[f = 0] = \delta$ for some δ to be determined later;
   (a) if $f = 0$, then set $d = 0$;
   (b) otherwise select d at random from $\mathbb{Z}_q^*$;
2. If $\mathrm{ID}' \notin \mathsf{ID}$, then set $f = 0$ and $d = 0$;
3. Set $T = zP - d(bP)$ where z is selected at random from $\mathbb{Z}_q^*$;
4. Record $\langle(\mathrm{ID}^1, \ldots, \mathrm{ID}^n), M, z, f, d, T\rangle$ in Λ_{H_2};
5. Return T to $\mathcal{A}$ as the output of H_2.

Again, the output T is always chosen uniformly in $\mathbb{G}$.

$\mathsf{Extract}$ queries: When algorithm $\mathcal{A}$ requests a private key associated with identifier ID, algorithm $\mathcal{B}$ responds as follows:

1. Recover the associated tuple $\langle\mathrm{ID}, r, R\rangle$ from Λ_{H_1} (first making a query to H_1 if necessary);
2. If $\mathrm{ID} = \mathrm{ID}'$, then $\mathcal{B}$ aborts;
3. Otherwise $\mathcal{B}$ returns $r(aP)$ to $\mathcal{A}$.

It is easy to see that $\mathcal{A}$ receives a valid private key for identifier ID, provided $\mathcal{B}$ does not abort.

Sign queries: Algorithm $\mathcal{A}$ can request a signature for any set of identifiers $\mathsf{ID} = \{\mathrm{ID}^1, \ldots, \mathrm{ID}^n\}$ and any message M. We can assume the list $\mathrm{ID}^1, \ldots, \mathrm{ID}^n$ is ordered lexicographically.

In responding to $\mathcal{A}$'s request, $\mathcal{B}$ performs the following steps:

1. Recover $\langle(\mathrm{ID}^1, \ldots, \mathrm{ID}^n), M, z, f, d, T\rangle$ from Λ_{H_2} (if $\mathcal{A}$ has not already queried H_2 on the appropriate input, then $\mathcal{B}$ first makes the relevant H_2 query himself);
2. If $\mathrm{ID}' \notin \mathsf{ID}$, then:
   (a) Select s_φ at random from $\mathbb{Z}_q^*$;
   (b) Run $\mathsf{Extract}$ on input ID^j to recover a private key S^j, for each $1 \leq j \leq n$;
   (c) Compute $\varphi = \sum_{j=1}^n S^j + s_\varphi T$ and $Q_\varphi = s_\varphi P$;
   (d) Return $\langle\varphi, Q_\varphi\rangle$ to $\mathcal{A}$.
3. If $\mathrm{ID}' \in \mathsf{ID}$ and $f = 0$, then abort;
4. Otherwise (when $\mathrm{ID}' \in \mathsf{ID}$ and $f = 1$, so that $T = zP - d(bP)$):
   (a) Let $\mathrm{ID}^{j'} = \mathrm{ID}'$, and obtain entry $\langle\mathrm{ID}', r', R'\rangle$ from Λ_{H_1};
   (b) Set $v = d^{-1} \bmod q$ and, for each $j \neq j'$, run $\mathsf{Extract}$ on input ID^j to recover a private key S^j.
   (c) Set

$$\varphi = \sum_{j=1, j\neq j'}^{n} S^j + r'(aP) + vz(aP), \quad Q_\varphi = vaP;$$

(d) Return $\langle \varphi, Q_\varphi \rangle$ to $\mathcal{A}$.

We remark that in step 4 above, $\mathcal{B}$ is not able to compute the private key corresponding to identifier ID' (which should equal $abP + ar'P$), but is still able to compute a valid signature. This can be shown from the definitions of T and R', which were made with the aim of causing certain cancelations to occur in the signature computation: if $\mathcal{B}$'s output is to be a valid signature, we should have $\varphi = \sum_{j=1}^n S^j + s_\varphi H_2(\mathrm{ID}^1, \ldots, \mathrm{ID}^n, M)$ with $s_\varphi = va$. Indeed,

$$\begin{aligned}
&\textstyle\sum_{j=1}^n S^j + s_\varphi H_2(\mathrm{ID}^1, \ldots, \mathrm{ID}^n, M) \\
&= \textstyle\sum_{j=1, j\neq j'}^n S^j + (abP + ar'P) + va(zP - d(bP)) \\
&= \textstyle\sum_{j=1, j\neq j'}^n S^j + r'(aP) + abP - vdabP + vz(aP) \\
&= \varphi
\end{aligned}$$

Forgery: Eventually, algorithm $\mathcal{A}$ outputs a set ID^* of n target identifiers, a message M^* and a pair $\sigma^* = \langle \varphi^*, Q_\varphi^* \rangle$. If σ^* is to be a valid signature then it should satisfy the verification equation:

$$e(P, \varphi^*) = e(Q_0, \sum_{j=1}^n H_1(\mathrm{ID}^{*j})) \cdot e(Q_\varphi^*, P_{M*})$$

where the identifiers in ID^* are, in lexicographic order, $\mathrm{ID}^{*1}, \ldots, \mathrm{ID}^{*n}$, and where

$$P_{M*} = H_2(\mathrm{ID}^{*1}, \ldots, \mathrm{ID}^{*n}, M^*).$$

This ends our description of $\mathcal{B}$'s interaction with $\mathcal{A}$. If $\mathcal{B}$ does not abort, then $\mathcal{A}$'s view is identical to that in a real attack. Hence the probability that $\mathcal{A}$ outputs a valid signature is at least ϵ, provided $\mathcal{B}$ does not abort.

Now let $\mathcal{E}_1$ be the event that $\mathcal{A}$ has queried H_1 on every identifier ID^{*j}. It is obvious from inspecting the above verification equation for $\mathcal{A}$'s output that if $\mathcal{E}_1$ does not occur, then $\mathcal{A}$'s success probability is negligible. Because ϵ is non-negligible (assuming $\mathcal{B}$ has not aborted), it follows that $\mathcal{E}_1$ does occur. Now the probability that both $\mathrm{ID}' \in \mathsf{ID}^*$ and that ID' is not the subject of an $\mathsf{Extract}$ query is at least $1/q_{H_1}$. This follows because we selected $c \in \{1, \ldots, q_{H_1}\}$ uniformly at random and set ID' to be the identifier in the c-th query to H_1, and because at least one identity appearing in ID^* is not the subject of an $\mathsf{Extract}$ query.

Let $\mathcal{E}_2$ be the event that $\mathcal{A}$ has made an H_2 query on input $\mathrm{ID}^{*1}, \ldots, \mathrm{ID}^{*n}$ and M^*. If $\mathcal{E}_2$ does not occur, then $\mathcal{A}$'s success probability is again negligible. For otherwise, P_{M*} is uniformly distributed over $\mathbb{G}$ and the probability that the above verification equation holds is $1/q$, which is negligible in λ, the security parameter. Because ϵ is non-negligible, it follows that $\mathcal{E}_2$ does occur, so there is an entry $\langle (\mathsf{ID}^{*1}, \ldots, \mathsf{ID}^{*n}), M^*, z^*, f^*, d^*, T^* \rangle$ in Λ_{H_2}.

We now explain how $\mathcal{B}$ analyzes $\mathcal{A}$'s output. If $f^* = 1$ or if $\mathrm{ID}' \notin \mathsf{ID}^*$, then $\mathcal{B}$ aborts. Otherwise, we can assume that $f^* = 0$ and $\mathrm{ID}' = \mathrm{ID}_t^{*j'}$ for some $\mathrm{ID}_t^{*j'} \in \mathsf{ID}^*$. Then, from the simulation of H_2, we have $d^* = 0$ and $P_{M*} =$

$H_2(\mathrm{ID}^{*1}, \ldots, \mathrm{ID}^{*n}, M^*) = z^*P$. Then, by rearranging the verification equation for $\mathcal{A}$'s output, we obtain:

$$\begin{aligned} e(P, \varphi^*) &= e(Q_0, \textstyle\sum_{j=1}^n H_1(\mathrm{ID}^{*j})) \cdot e(Q^*_\varphi, P_{M*}) \\ &= e(aP, \textstyle\sum_{j=1}^n H_1(\mathrm{ID}^{*j})) \cdot e(Q^*_\varphi, z^*P) \\ &= e(P, \textstyle\sum_{j=1}^n aH_1(\mathrm{ID}^{*j})) \cdot e(P, z^*Q^*_\varphi) \\ &= e(P, \textstyle\sum_{j=1}^n aH_1(\mathrm{ID}^{*j}) + z^*Q^*_\varphi). \end{aligned}$$

From the non-degeneracy of the pairing, we may now deduce

$$\varphi^* = \sum_{j=1}^{n} aH_1(\mathrm{ID}^{*j}) + z^*Q^*_\varphi.$$

But, for $j \neq j'$, we have $aH_1(\mathrm{ID}^{*j}) = r_j(aP)$ for some value r_j available in list Λ_{H_1}, while $aH_1(\mathrm{ID}^{*j'}) = r'(aP) + abP$. It is now easy to see that $\mathcal{B}$ can recover abP from $\mathcal{A}$'s output and its knowledge of the r_j values and z^*, provided $\mathcal{A}$'s output is a valid signature. Hence, if $\mathcal{B}$ does not abort, then $\mathcal{B}$ can extract the value abP with probability at least ϵ.

So it remains to calculate the probability that $\mathcal{B}$ does not abort. $\mathcal{B}$ is forced to abort if $\mathcal{A}$ makes an $\mathsf{Extract}$ query on ID', if $\mathcal{A}$ makes a Sign query on a set ID containing ID' for which $f = 0$, or if the set ID^* does not contain ID', or if $f^* = 1$. Hence $\mathcal{B}$ does *not* abort if all of the following conditions are met:

- $\mathcal{A}$ does not make an $\mathsf{Extract}$ query on ID';
- $f = 1$ in every Sign query involving ID' made by $\mathcal{A}$;
- $\mathrm{ID}' \in \mathsf{ID}^*$ and $f^* = 0$.

We have already seen that the probability that $\mathcal{A}$ does not make an $\mathsf{Extract}$ query on ID' and $\mathrm{ID}' \in \mathsf{ID}^*$ is $1/q_{H_1}$. Setting $\delta = \Pr[f = 0] = 1/q_{H_2}$, where q_{H_2} denotes a bound on the number of queries made to H_2, it is then easy to see that $\mathcal{B}$'s success probability is at least

$$\epsilon \cdot \frac{1}{q_{H_1}} \left(1 - \frac{1}{q_{H_2}}\right)^{q_S} \cdot \frac{1}{q_{H_2}}$$

where q_S is the number of Sign queries involving ID' made by $\mathcal{A}$. In turn, since $q_S \leq q_{H_2}$ and, for large q_{H_2}, we have $(1 - 1/q_{H_2})^{q_{H_2}} \approx 1/\mathbf{e}$ where $\mathbf{e}$ is the base of natural logarithms, we obtain (neglecting negligible terms) that $\mathcal{B}$'s success probability is at least

$$\epsilon/(\mathbf{e} \cdot q_{H_1} \cdot q_{H_2})$$

as required. This completes the proof. □

We now sketch how this proof might be extended to cope with the more complicated situation of our multi-key HIBS (rather than IBS). The main idea is to borrow the simulation techniques used in proving the security of the Gentry-Silverberg HIBE scheme, [19, Lemma 2]. There, it is shown how to simulate H_1

and $\mathsf{Extract}$ queries for a HIBE scheme that has the same key generation procedures as our multi-key HIBS scheme, in such a way as to embed a value abP into the private key held by some proportion of the entities, whilst allowing $\mathsf{Extract}$ queries to be answered for all other entities (in the more challenging hierarchical setting). When combined with our approach to handling Sign queries, this simulation technique should yield a security reduction for our multi-key HIBS scheme. However, we stress that we have so far only obtained a security proof for some special cases in the hierarchical setting using the sketched techniques. Constructing a proof for the general case remains an open problem, and, unfortunately, as with the proofs in [19], we expect that any reduction obtained using this approach will not be very tight.

5 Reducing the Cost of Verification

In this section, we study how the complexity of the verification algorithm for our concrete multi-key HIBS scheme can be reduced in special cases.

The main situation we consider is where all the identifiers involved in signing are at the same level in the hierarchy and have a common parent at the level above. This includes as a special case the situation where all identifiers are at the first level in the hierarchy. In this case, the Sign and Verify algorithms of multi-key HIBS scheme can be modified as follows.

Sign**:** Given any $n \geq 1$, and a set SK of n signing keys corresponding to a set of identifiers $\{\mathrm{ID}_t^1, \ldots, \mathrm{ID}_t^n\}$ having a common parent, and a message M, the signer:

1. chooses a secret value $s_\varphi \in \mathbb{Z}_q^*$;
2. computes $P_M = H_2(\mathrm{ID}_t^1, \ldots, \mathrm{ID}_t^n, M)$ (where we assume the identifiers are first placed in lexicographic order if necessary);
3. calculates
$$\varphi = \sum_{j=1}^{n} S_t^j + s_\varphi P_M \quad \text{and} \quad Q_\varphi = s_\varphi P_0.$$

The algorithm outputs the signature $\sigma = \langle \varphi, Q_1, \ldots, Q_{t-1}, Q_\varphi \rangle$. Here, we assume (because of the common parent) that each signing key in SK involves the same list $Q_1, \ldots, Q_{t-1}$ of Q-values.

Verify**:** Given $\sigma = \langle \varphi, Q_1, \ldots, Q_{t-1}, Q_\varphi \rangle$, a set of identifiers $\mathsf{ID} = \{\mathrm{ID}_t^1, \ldots, \mathrm{ID}_t^n\}$ having a common parent, and a message M, the verifier:

1. computes $P_i = H_1(\mathrm{ID}_i)$ for $1 \leq i \leq t-1$ and $P_t^j = H_1(\mathrm{ID}_t^j)$ for $1 \leq j \leq n$;
2. computes $P_M = H_2(\mathrm{ID}_t^1, \ldots, \mathrm{ID}_t^n, M)$ (first arranging the identifiers lexicographically if they are not already in this order);
3. checks if $e(P_0, \varphi)$ is equal to
$$e\left(Q_{t-1}, \sum_{j=1}^{n} P_t^j\right) \cdot e(Q_\varphi, P_M) \cdot \left(\prod_{i=1}^{t-1} e(Q_{i-1}, P_i)\right)^n,$$
outputting `valid` if this equation holds, and `invalid` otherwise.

Note that the above verification algorithm requires only $t+2$ pairing computations as compared to the $n \cdot t + 2$ that would be needed with the unoptimised verification algorithm of Section 3.

For $n = 1$, the above SIGN and VERIFY algorithms are essentially those of the Gentry-Silverberg HIBS scheme [19]. In the special case where we take $t = 1$ throughout in the above scheme, we obtain a multi-key IBS (i.e. one-level) scheme in which only 3 pairing computations are needed during verification.

In more general situations, we can obtain a more efficient verification algorithm whenever the nodes associated with the signing keys used to generate a signature have some common ancestors and common Q-values. For then certain pairing computations in the verification equation can be eliminated. Recall that this equation involves a term $\prod_{j=1}^{n} \prod_{i=1}^{t_j} e(Q_{i-1}^j, P_i^j)$. To illustrate the possibility of eliminating pairing computations, suppose for example that all n nodes have a single common ancestor at some level k (and therefore at all levels above level k as well). Then the n terms $e(Q_{i-1}^j, P_i^j)$, $1 \le j \le n$, are in fact equal for each value $i \le k$, and so we can replace these n ostensibly different pairings at level i with a power of a single pairing, for each $i \le k$. Similar remarks also apply when some (but not necessarily all) nodes share a common ancestor at some level in the hierarchy.

6 Efficiency Comparison

To gauge the efficiency gain offered by our multi-key HIBS primitive in comparison to IBMS schemes, we compare the multi-key IBS scheme that arises from our multi-key HIBS scheme in the one-level case (with an optimised verification equation) with the RSA-based Bellare-Neven IBMS scheme [6] and the pairing-based Gentry-Ramzan IBMS scheme [18]. The computational costs for these schemes are shown in Table 1.

From Table 1, it is evident that our multi-key IBS scheme is much more efficient than both the Bellare-Neven and the Gentry-Ramzan IBMS schemes in terms of signing cost at an equivalent security level. Suppose our scheme and the Gentry-Ramzan IBMS scheme are instantiated using pairings defined on an elliptic curve at the 80-bit security level, while the Bellare-Neven scheme is instantiated using a 1024-bit RSA modulus (also offering roughly 80 bits of security). Then the main signing cost in our scheme is 2 elliptic curve point multiplications, whereas the Gentry-Ramzan IBMS scheme needs $2n$. The signing cost of the Bellare-Neven scheme is much greater for reasonable values of n, and this is due to its interactive nature. This is as expected, since our scheme exploits the fact that signing keys can be "aggregated" before generating a signature.

On the other hand, the verification cost for the Gentry-Ramzan scheme and our scheme is the same. The verification cost for the Bellare-Neven scheme appears to be the least among the three schemes (since a modular exponentiation is expected to be faster than a pairing computation).

Our signatures have the same length as in the Gentry-Ramzan scheme and are generally shorter than signatures in the Bellare-Neven scheme, as they com-

Table 1. Computational costs for the Bellare-Neven IBMS scheme, the Gentry-Ramzan IBMS scheme and our multi-key IBS (one-level) scheme. Here, ADD denotes the number of elliptic curve point additions, eMUL the number of elliptic curve point multiplications, PAI the number of pairing computations, HASH the number of hash operations, mMUL the number of modular multiplications, and EXP the number of modular exponentiations.

	ADD	eMUL	PAI	HASH	mMUL	EXP
Bellare-Neven IBMS						
signing	-	-	-	$n(n+1)$	n^2+n-1	$2n$
verification	-	-	-	$n-1$	n	2
Gentry-Ramzan IBMS						
signing	$3n-2$	$2n$	0	n	-	-
verification	$n-1$	0	3	$n+1$	-	-
Multi-key IBS						
signing	n	2	0	1	-	-
verification	$n-1$	0	3	$n+1$	-	-

prise of only two group elements (at around 320 bits in total), instead of the 1184 bits quoted in [6]. Note that bandwidth savings can be more critical than computational efficiency gains in many contexts.

7 Conclusions and Open Problems

There exist practical applications which require users to demonstrate possession of more than a single private signing key. We developed a new primitive called multi-key hierarchical identity-based signatures (multi-key HIBS) which can achieve this more efficiently than with existing approaches. We showed that our multi-key HIBS primitive can make use of a security model that is at least as strong as, if not even stronger than, existing related security models for multi-signature schemes and aggregate signature schemes. We also showed how the new primitive could be efficiently instantiated in the Random Oracle Model.

Currently, our multi-key HIBS primitive is restricted to demonstrating the possession of all the private keys in a given set. This should be compared to threshold cryptographic schemes in which signing parties, for example, effectively demonstrate knowledge of a subset of size k of a set of private keys of size n. It will be interesting to generalise our multi-key HIBS concept to the threshold setting. This could involve multiple hierarchies with different roots, or potentially, a single hierarchy, but working with a threshold of private keys within that hierarchy.

Naturally, there exists a version of our multi-key HIBS primitive in the normal public-key setting. It seems easy to design a one-level non-identity-based multi-key scheme by adapting the BGLS aggregate signature scheme [9], for example. Nevertheless, it will be interesting to explore security models and concrete schemes that support hierarchical signing keys in that setting. It seems likely that it will be possible to adapt the Bellare-Neven IBMS scheme to produce a more efficient RSA-based multi-key IBS. However, finding an efficient, RSA-based, multi-key HIBS may be a more challenging task.

It may also be interesting to try to construct a multi-key HIBS scheme secure in the standard model, perhaps by adapting the work of Lu *et al.* [25].

References

1. B. Adida, D. Chau, S. Hohenberger, and R.L. Rivest. Lightweight signatures for email. In R. De Prisco and M. Yung, editors, *Proceedings of the 5th International Conference on Security and Cryptography for Networks (SCN 2006)*, pages 288–302. Springer-Verlag LNCS 4116, September 2006.
2. J. Baek and Y. Zheng. Identity-based threshold signature scheme from the bilinear pairings. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2004) Volume 1*, pages 124–128. IEEE Computer Society Press, 2004.
3. W. Bagga and R. Molva. Policy-based cryptography and applications. In A.S. Patrick and M. Yung, editors, *Proceedings of the 9th International Conference on Financial Cryptography and Data Security (FC 2005)*, pages 72–87. Springer-Verlag LNCS 3570, February 2005.
4. P.S.L.M. Barreto, B. Libert, N. McCullagh, and J. Quisquater. Efficient and provably-secure identity-based signatures and signcryption from Bilinear maps. In B.K. Roy, editor, *Advances in Cryptology - Proceedings of ASIACRYPT 2005*, pages 515–532. Springer-Verlag LNCS 3788, December 2005.
5. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - Proceedings of EUROCRYPT 2004*, pages 268–286. Springer-Verlag LNCS 3027, 2004.
6. M. Bellare and G. Neven. Identity-based multi-signatures from RSA. In M. Abe, editor, *Proceedings of the RSA Conference: Topics in Cryptology - the Cryptographers' Track (CT-RSA 2007)*, pages 145–162. Springer-Verlag LNCS 4377, February 2007.
7. D. Boneh, X. Boyen, and E. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2005*, pages 440–456. Springer-Verlag LNCS 3494, May 2005.
8. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *Advances in Cryptology - Proceedings of CRYPTO 2001*, pages 213–229. Springer-Verlag LNCS 2139, August 2001.
9. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In E. Biham, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2003*, pages 416–432. Springer-Verlag LNCS 2656, May 2003.
10. X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In C. Dwork, editor, *Advances in Cryptology - Proceedings of CRYPTO 2006*, pages 290–307. Springer-Verlag LNCS 4117, August 2006.

11. A. Burnett, F. Byrne, T. Dowling, and A. Duffy. A biometric identity based signature scheme. *International Journal of Network Security*, 5(3):317–326, November 2007.
12. J.C. Cha and J.H. Cheon. An identity-based signature from Gap Diffie-Hellman groups. In Y.G. Desmedt, editor, *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography - PKC 2003*, pages 18–30. Springer-Verlag LNCS 2567, January 2003.
13. J. Crampton and H.W. Lim. *Role Signatures for Access Control in Grid Computing.* Royal Holloway, University of London, Technical Report RHUL-MA-2007-2, May 2007.
14. J. Crampton, H.W. Lim, K.G. Paterson, and G. Price. A certificate-free grid security infrastructure supporting password-based user authentication. In *Proceedings of the 6th Annual PKI R&D Workshop 2007.* NIST Interagency Report, 2007.
15. I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 15(3):200–222, 2001.
16. S.D. Galbraith, K.G. Paterson, and N.P. Smart. *Pairings for Cryptographers.* Cryptology ePrint Archive, Report 2006/165, May 2006. Available at `http://eprint.iacr.org/2006/165`.
17. D. Galindo, J. Herranz, and E. Kiltz. On the generic construction of identity-based signatures with additional properties. In X. Lai and K. Chen, editors, *Advances in Cryptology - Proceedings of ASIACRYPT 2006*, pages 178–193. Springer-Verlag LNCS 4284, December 2006.
18. C. Gentry and Z. Ramzan. Identity-based aggregate signatures. In M. Yung, editor, *Proceedings of the 9th International Workshop on Theory and Practice in Public Key Cryptography - PKC 2006*, pages 257–273. Springer-Verlag LNCS 3958, April 2006.
19. C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In Y. Zheng, editor, *Advances in Cryptology - Proceedings of ASIACRYPT 2002*, pages 548–566. Springer-Verlag LNCS 2501, December 2002.
20. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In R.N. Wright, S.D.C. di Vimercati, and V. Shmatikov, editors, *Proceedings of the 13th ACM Computer and Communications Security Conference (CCS 2006)*, pages 89–98. ACM Press, October 2006.
21. F. Hess. Efficient identity based signature schemes based on pairings. In K. Nyberg and H. Heys, editors, *Proceedings of the 9th International Workshop on Selected Areas in Cryptography (SAC 2002)*, pages 310–324. Springer-Verlag LNCS 2593, August 2003.
22. J. Horwitz and B. Lynn. Towards hierarchical identity-based encryption. In L.R. Knudsen, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2002*, pages 466–481. Springer-Verlag LNCS 2332, May 2002.
23. A. Joux. A one round protocol for tripartite Diffie-Hellman. In W. Bosma, editor, *Proceedings of 4th Algorithmic Number Theory Symposium (ANTS-IV)*, pages 385–394. Springer-Verlag LNCS 1838, July 2000.
24. A. Khalili, J. Katz, and W.A. Arbaugh. Toward secure key distribution in truly ad-hoc networks. In *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03)*, pages 342–346. IEEE Computer Society Press, January 2003.
25. S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters. Sequential aggregate signatures and multisignatures without random oracles. In S. Vaudenay, edi-

tor, *Advances in Cryptology - Proceedings of EUROCRYPT 2006*, pages 465–485. Springer-Verlag LNCS 4004, May 2006.
26. K.G. Paterson. ID-based signatures from pairings on elliptic curves. *Electronics Letters*, 38(18):1025–1026, 2002.
27. K.G. Paterson and J.C.N. Schuldt. Efficient identity-based signatures secure in the standard model. In L. Batten and R. Safavi-Naini, editors, *Proceedings of the 11th Australasian Conference on Information Security and Privacy (ACISP 2006)*, pages 207–222. Springer-Verlag LNCS 4058, July 2006.
28. M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure attribute-based systems. In R.N. Wright, S.D.C. di Vimercati, and V. Shmatikov, editors, *Proceedings of the 13th ACM Computer and Communications Security Conference (CCS 2006)*, pages 99–112. ACM Press, October 2006.
29. A. Sahai and B. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2005*, pages 457–473. Springer-Verlag LNCS 3494, May 2005.
30. R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Proceedings of the 2000 Symposium on Cryptography and Information Security (SCIS 2000)*, January 2000.
31. R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996.
32. A. Shamir. Identity-based cryptosystems and signature schemes. In G.R. Blakley and D. Chaum, editors, *Advances in Cryptology - Proceedings of CRYPTO '84*, pages 47–53. Springer-Verlag LNCS 196, August 1985.
33. N.P. Smart. Access control using pairing based cryptography. In M. Joye, editor, *Proceedings of the RSA Conference: Topics in Cryptology - the Cryptographers' Track (CT-RSA 2003)*, pages 111–121. Springer-Verlag LNCS 2612, April 2003.