Identity-based signatures secure in the standard model

Kenneth G. Paterson and Jacob C. N. Schuldt

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

Abstract. We present the first (to our knowledge) identity-based signature scheme that is provably secure in the standard model. Our construction is obtained from a modification of Waters' recently proposed identity-based encryption scheme. The scheme is computationally efficient, the signatures are short, and the scheme's security rests on the hardness of the computational Diffie-Hellman problem in groups equiped with a pairing.

1 Introduction

Identity based encryption (IBE), introduced by Shamir [Sha84], enables the computation of a public key for an entity, given only some general scheme parameters and a string identifying the entity (e.g. an e-mail address, a telephone number, etc.). A private key generator (PKG) computes private keys from a master secret and distributes these to the entities participating in the scheme. This eliminates the need for certificates as used in a traditional public key infrastructure. Although Shamir proposed the idea of an IBE scheme in 1984, no construction that was both efficient and secure was found until recently, when the work of Boneh and Franklin [BF01] and Cocks [Coc01] was published. Since then, a large number of papers have been published in this area (see [Bar] for a list of some of these), including a number of identity-based signature (IBS) schemes [Pat02,Hes02,CC03,Yi03].

An interesting observation, attributed to Naor by Boneh and Franklin, is that any IBE can be used to construct a (non ID-based) signature scheme. This is done by keeping the master secret of the IBE scheme as the private key and publishing the scheme parameters of the IBE scheme as the public key. A signature on a message $\mathfrak{m}$ is then the private key of the identity $\mathfrak{u}_m = \mathfrak{m}$ and verification can be performed by selecting a random message $\mathfrak{m}_r$, encrypting $\mathfrak{m}_r$ with the public key of $\mathfrak{u}_m$, and verifying that decryption is possible using the given signature as a decryption key. If the used IBE scheme is IND-ID-CCA secure, the resulting signature scheme is existentially unforgeable under an adaptive chosen message attack. This technique was used by Boneh, Lynn and Shacham [BLS04] to construct short signatures from the IBE scheme of Boneh and Franklin [BF01], and likewise by Boneh and Boyen to obtain another short signature scheme [BB04c] from an IBE scheme due to the same authors [BB04a].

As noted by Gentry and Silverberg [GS02], IBS schemes can be constructed in a very similar way if a hierarchical IBE (HIBE) scheme is used in place of an IBE scheme. This will, in fact, lead to a hierarchical IBS scheme where signing identities are part of a hierarchy having one level less than the used HIBE scheme. When an identity $(\mathfrak{u}_1, \ldots, \mathfrak{u}_t)$ signs a message $\mathfrak{m}$, the identity $\mathfrak{u}_m = \mathfrak{m}$ is inserted as a child of $(\mathfrak{u}_1, \ldots, \mathfrak{u}_t)$ in the hierarchy. As in the above, a signature is the private key of $\mathfrak{u}_m$ and a verifier checks that decryption of a random message, encrypted with the public key for identity $(\mathfrak{u}_1, \ldots, \mathfrak{u}_t, \mathfrak{u}_m)$, is possible. Limiting the used HIBE to a 2-level scheme leads to an ordinary IBS.

Most of the schemes mentioned above are provably secure in the random oracle model [BR93]. However, it has been shown that when random oracles are instantiated with concrete hash functions, the resulting scheme may not be secure [CGH98,BBP04]. Recently, efforts have been made to construct schemes that are provably secure in the standard model to overcome this problem. Boneh and Boyen initially proposed an IBE scheme [BB04a] which could be proven secure using a "Selective ID" security model that is slightly weaker than the model original proposed by Boneh and Franklin. The same authors later proposed an IBE scheme [BB04b] that is secure in the full Boneh-Franklin security model and in the standard model, but which is inefficient. Finally, Waters [Wat05] succeeded in constructing a fairly efficient scheme which meets both requirements. Naccache [Nac05] and Sarkar-Chatterjee [SC05] independently proposed a technique for reducing the size requirements of Waters' scheme, making it more suitable for practical use. The signature scheme of Boneh and Boyen [BB04c] mentioned above was also proven secure in the standard model.

Our contribution As a natural extension of the efforts to provide secure schemes without the use of random oracles, we construct the first IBS scheme that is provably secure in the standard model. Our scheme is based on a hierarchical extension of Waters' scheme, and we use the above-described technique of converting a 2-level HIBE scheme into a IBS scheme for our construction. However, while the security of Waters' scheme relies on the hardness of the bilinear Diffie-Hellman problem, we prove our signature scheme to be secure under the weaker computational Diffie-Hellman assumption. This assumption seems much more natural than many of the hardness assumptions recently introduced to pairing based cryptography. In terms of signature size and computational cost, our new scheme is competitive with existing ID-based signature schemes (that are provably secure only in the random oracle model). The only drawback of our scheme is the relatively large size of its public parameters. However, we show how the technique of Naccache and Sarkar-Chatterjee can be applied to our scheme to reduce the size of the public parameters, at the cost of a looser security reduction. Lastly, we examine the aggregation properties of our scheme.

2 Identity-based signatures

An identity-based signatures scheme can be described as a collection of the following four algorithms:

Setup This algorithm is run by the master entity on input a security parameter, and generates the public parameters `params` of the scheme and a master secret. The master entity publishes `params` and keeps the master secret to itself.

Extract Given an identity $\mathfrak{u}$, the master secret and `params`, this algorithm generates the private key $d_{\mathfrak{u}}$ of $\mathfrak{u}$. The master entity will use this algorithm to generate private keys for all entities participating in the scheme and distribute the private keys to their respective owners through a secure channel.

Sign Given a message $\mathfrak{m}$, an identity $\mathfrak{u}$, a private key $d_{\mathfrak{u}}$ and `params`, this algorithm generates the signature σ of $\mathfrak{u}$ on $\mathfrak{m}$. The entity with identity $\mathfrak{u}$ will use this algorithm for signing.

Verify Given a signature σ, a message $\mathfrak{m}$, an identity $\mathfrak{u}$ and `params`, this algorithm outputs `accept` if σ is a valid signature on $\mathfrak{m}$ for identity $\mathfrak{u}$, and outputs `reject` otherwise.

2.1 Existential unforgeability

The security model of existential unforgeability under an adaptive chosen message attack, defined by Goldwasser, Micali and Rivest [GMR88], can be extended to the identity-based scenario in a natural way. We will define security for identity-based signature schemes by the following game between a challenger and an adversary:

Setup The challenger runs the algorithm **Setup** of the signature scheme and obtains both the public parameters `params` and the master secret. The adversary is given `params` but the master secret is kept by the challenger.

Queries The adversary adaptively makes a number of different queries to the challenger. Each query can be one of the following.

- Extract query. The adversary can ask for the private key of any identity $\mathtt{u}$. The challenger responds by running **Extract**$(\mathtt{params}, \mathtt{u})$ and forwards the private key $d_{\mathtt{u}}$ to the adversary.
- Sign query. The adversary can ask for the signature of any identity $\mathtt{u}$ on any message $\mathtt{m}$. The challenger responds by first running **Extract**$(\mathtt{params}, \mathtt{u})$ to obtain the private key $d_{\mathtt{u}}$ of $\mathtt{u}$, and then running **Sign**$(\mathtt{params}, d_{\mathtt{u}}, \mathtt{u}, \mathtt{m})$ to obtain a signature, which is forwarded to the adversary.

Forgery The adversary outputs a message $\mathtt{m}^*$, an identity $\mathtt{u}^*$ and a string σ^*. The adversary *succeeds* if the following hold true:

1. **Verify**$(\mathtt{params}, \mathtt{u}^*, \mathtt{m}^*, \sigma^*) = \mathtt{accept}$
2. The adversary has not made an extract query on $\mathtt{u}^*$.
3. The adversary has not made a sign query on $(\mathtt{u}^*, \mathtt{m}^*)$.

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

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

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

Definition 1. *An adversary $\mathcal{A}$ is said to be an (ϵ, t, q_e, q_s)-forger of an identity-based signature scheme if $\mathcal{A}$ has advantage at least ϵ in the above game, runs in time at most t, and makes at most q_e and q_s extract and sign queries, respectively. A scheme is said to be (ϵ, t, q_e, q_s)-secure if no (ϵ, t, q_e, q_s)-forger exists.*

The above game can easily be extended to cover strong unforgeability by changing the third requirement in the forgery stage to "σ^* was not output as a response to a sign query". However, our concrete scheme does not enjoy security in this stronger sense, as an adversary can easily modify an existing signature on a message into a new signature on the same message.

3 Complexity assumptions

The security of our signature scheme will be reduced to the hardness of the computational Diffie-Hellman (CDH) problem in the group in which the signature is constructed. We briefly review the definition of the CDH problem:

Definition 2. *Given a group $\mathbb{G}$ of prime order p with generator g and elements $g^a, g^b \in \mathbb{G}$ where a, b are selected uniformly at random from $\mathbb{Z}_p^*$, the CDH problem in $\mathbb{G}$ is to compute g^{ab}.*

Definition 3. *We say that the (ϵ, t)-CDH assumption holds in a group $\mathbb{G}$ if no algorithm running in time at most t can solve the CDH problem in $\mathbb{G}$ with probability at least ϵ.*

4 Construction

Our new identity-based signature scheme is based on an hierarchical extension of the identity-based encryption scheme presented by Waters [Wat05]. However, as shown in Section 5, the security of our scheme can be reduced to the hardness of the CDH problem, whereas Waters' scheme relies on the stronger Bilinear Diffie-Hellman assumption. Our construction is based on bilinear maps and we now briefly review some of the basic properties of such maps.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order p and let g be a generator of $\mathbb{G}$. The map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is said to be an admissible map if the following three conditions hold true:

- e is bilinear, i.e. $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_p$.
- e is non-degenerate, i.e. $e(g, g) \neq 1$.
- e is efficiently computable.

See [Gal05] for more details on the construction of such maps. In Section 7, we will sketch the modifications necessary to allow our scheme to operate in the more general setting where $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with $\mathbb{G}_1 \neq \mathbb{G}_2$.

In the following all identities and messages will be assumed to be bit strings of length n_u and n_m, respectively. To construct a more flexible scheme which allows identities and messages of arbitrary lengths, collision-resistant hash functions, $H_u : \{0,1\}^* \to \{0,1\}^{n_u}$ and $H_m : \{0,1\}^* \to \{0,1\}^{n_m}$, can be defined and used to create identities and messages of the desired length. We will use the notation $v \leftarrow_R S$ as a short hand for choosing a value v uniformly at random from the set S. Our new signature scheme is defined by the following algorithms:

Setup Choose groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p such that an admissible pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ can be constructed and pick a generator g of $\mathbb{G}$.

Now, pick a secret $\alpha \leftarrow_R \mathbb{Z}_p$, compute $g_1 = g^\alpha$ and pick $g_2 \leftarrow_R \mathbb{G}$. Furthermore, pick elements $u', m' \leftarrow_R \mathbb{G}$ and vectors $\boldsymbol{U} = (u_i)$, $\boldsymbol{M} = (m_i)$ of length n_u and n_m, respectively, whose entries are random elements from $\mathbb{G}$. The public parameters are $\texttt{params} = (\mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, u', \boldsymbol{U}, m', \boldsymbol{M})$ and the master secret is g_2^α.

Extract Let $\mathfrak{u}$ be a bit string of length n_u representing an identity and let $\mathfrak{u}[i]$ be the ith bit of $\mathfrak{u}$. Define $\mathcal{U} \subset \{1, \ldots, n_u\}$ to be the set of indicies i such that $\mathfrak{u}[i] = 1$.

To construct the private key, $d_\mathfrak{u}$, of the identity $\mathfrak{u}$, pick $r_u \leftarrow_R \mathbb{Z}_p$ and compute:

$$d_\mathfrak{u} = \left(g_2^\alpha \left(u' \prod_{i \in \mathcal{U}} u_i \right)^{r_u}, \ g^{r_u} \right).$$

Note that a user can easily re-randomize his private key after he has received it from the master entity.

Sign Let $\mathfrak{u}$ be the bit string of length n_u representing a signing identity and let $\mathfrak{m}$ be a bit string representing a message. As in the **Extract** algorithm, let $\mathcal{U}$ be the set of indicies i such that $\mathfrak{u}[i] = 1$, and likewise, let $\mathcal{M} \subset \{1, \ldots, n_m\}$ be the set of indicies j such that $\mathfrak{m}[j] = 1$, where $\mathfrak{m}[j]$ is the jth bit of $\mathfrak{m}$.

A signature of $\mathfrak{u}$ on $\mathfrak{m}$ is constructed by picking $r_m \leftarrow_R \mathbb{Z}_p$ and computing

$$\sigma = \left(g_2^\alpha \left(u' \prod_{i \in \mathcal{U}} u_i \right)^{r_u} \left(m' \prod_{j \in \mathcal{M}} m_j \right)^{r_m}, \ g^{r_u}, \ g^{r_m} \right) \in \mathbb{G}^3.$$

Verify Given a purported signature $\sigma = (V, R_u, R_m) \in \mathbb{G}^3$ of an identity $\mathfrak{u}$ on a message $\mathfrak{m}$, a verifier accepts σ if the following equality holds:

$$e(V, g) = e(g_2, g_1) e\Big(u' \prod_{i \in \mathcal{U}} u_i, R_u\Big) e\Big(m' \prod_{j \in \mathcal{M}} m_j, R_m\Big).$$

4.1 Correctness

If an entity with identity $\mathfrak{u}$ constructs a signature $\sigma = (V, R_u, R_m)$ on a message $\mathfrak{m}$ as described in the **Sign** algorithm above, it is easy to see that σ will be accepted by a verifier:

$$\begin{aligned} e(V, g) &= e\Bigg(g_2^{\alpha}\Big(u' \prod_{i \in \mathcal{U}} u_i\Big)^{r_u} \Big(m' \prod_{j \in \mathcal{M}} m_j\Big)^{r_m}, g\Bigg) \\ &= e(g_2, g)^{\alpha} e\Big(u' \prod_{i \in \mathcal{U}} u_i, g\Big)^{r_u} e\Big(m' \prod_{j \in \mathcal{M}} m_j, g\Big)^{r_m} \\ &= e(g_2, g_1) e\Big(u' \prod_{i \in \mathcal{U}} u_i, R_u\Big) e\Big(m' \prod_{j \in \mathcal{M}} m_j, R_m\Big) \end{aligned}$$

Thus the scheme is correct.

4.2 Efficiency

Our scheme has a private key size and a signature size of two and three group elements, respectively. Note, however, that the second value of a signature tuple, g^{r_u}, will remain the same for all signatures made by a given user. Hence, if many messages are signed by a single user and verified by a single verifier, the value g^{r_u} will only need to be included in one of the signatures. The public parameters of our scheme will consist of a description of the groups $\mathbb{G}, \mathbb{G}_T$ and the pairing e, and $n_u + n_m + 5$ group elements of $\mathbb{G}$. In a practical scheme, the size of the public parameters will be a performance concern and in Section 6 we will discuss how the number of group elements needed in `params` can be reduced.

To construct a signature, a signer will need to compute at most $n_m + 1$ multiplications in $\mathbb{G}$ ($n_m/2 + 1$ on average) and perform two exponentiations in $\mathbb{G}$. Verification requires at most $n_u + n_m$ multiplications in $\mathbb{G}$ ($(n_u + n_m)/2$ on average) and four pairing computations. However, the value $e(g_1, g_2)$ can be precomputed and cached, reducing the verification cost by one pairing. A further pairing can be eliminated if a verifier checks multiple signatures from a single signer.

Thus, our scheme is only slightly more expensive than existing IBS schemes (see for example the table in [Hes02]). However, these schemes are only proven secure in the random oracle model while our scheme, as the next section will show, can be proven secure in the standard model.

5 Proof of security

We will prove that our identity-based signature scheme is existentially unforgeable under a chosen message attack, in the standard model, given that the computational Diffie-Hellman problem is hard.

Theorem 1. *The identity-based signature scheme of Section 4 is (ϵ, t, q_e, q_s)-secure, assuming that the (ϵ', t')-CDH assumption holds in $\mathbb{G}$, where:*

$$\epsilon' = \frac{\epsilon}{16(q_e + q_s)q_s(n_u + 1)(n_m + 1)},$$
$$t' = t + O\big((q_e n_u + q_s(n_u + n_m))\rho + (q_e + q_s)\tau\big),$$

and ρ and τ are the time for a multiplication and an exponentiation in $\mathbb{G}$, respectively.

Proof We will assume that an (ϵ, t, q_e, q_s)-forger $\mathcal{A}$ for our scheme exists. From this forger, we will construct an algorithm $\mathcal{B}$ that solves CDH with probability at least ϵ' and in time at most t', contradicting the (ϵ', t')-CDH assumption. Our approach is based on that of [Wat05].

The algorithm $\mathcal{B}$ will be given a group $\mathbb{G}$, a generator g and the elements g^a and g^b. To be able to use $\mathcal{A}$ to compute g^{ab}, $\mathcal{B}$ must be able to simulate a challenger for $\mathcal{A}$. Such a simulation can be created in the following way:

Setup $\mathcal{B}$ sets $l_u = 2(q_e+q_s)$ and $l_m = 2q_s$, and randomly chooses two integers k_u and k_m, with $0 \leq k_u \leq n_u$ and $0 \leq k_m \leq n_m$. We will assume that $l_u(n_u + 1) < p$ and $l_m(n_m + 1) < p$ for the given values of q_e, q_s, n_u and n_m. The simulator then chooses an integer $x' \leftarrow_R \mathbb{Z}_{l_u}$ and a vector $\boldsymbol{X} = (x_i)$ of length n_u, with $x_i \leftarrow_R \mathbb{Z}_{l_u}$ for all i. Likewise, it randomly chooses another integer $z' \leftarrow_R \mathbb{Z}_{l_m}$ and a vector $\boldsymbol{Z} = (z_j)$ of length n_m, with $z_j \leftarrow_R \mathbb{Z}_{l_m}$ for all j. Lastly, $\mathcal{B}$ chooses two integers $y', w' \leftarrow_R \mathbb{Z}_p$ and two vectors, $\boldsymbol{Y} = (y_i)$ and $\boldsymbol{W} = (w_j)$, of length n_u and n_m, respectively, with $y_i, w_j \leftarrow_R \mathbb{Z}_p$ for all i and j.

To make the notation easier to follow, the following two pairs of functions are defined for an identity $\mathfrak{u}$ and a message $\mathfrak{m}$ respectively:

$$F(\mathfrak{u}) = x' + \sum_{i \in \mathcal{U}} x_i - l_u k_u \quad \text{and} \quad J(\mathfrak{u}) = y' + \sum_{i \in \mathcal{U}} y_i,$$
$$K(\mathfrak{m}) = z' + \sum_{j \in \mathcal{M}} z_j - l_m k_m \quad \text{and} \quad L(\mathfrak{m}) = w' + \sum_{j \in \mathcal{M}} w_j$$

Now, $\mathcal{B}$ constructs a set of public parameters for the IBE scheme by making the following assignments:

$$\begin{aligned} g_1 &= g^a, & g_2 &= g^b \\ u' &= g_2^{-l_u k_u + x'} g^{y'}, & u_i &= g_2^{x_i} g^{y_i} \;\; 1 \leq i \leq n_u \\ m' &= g_2^{-l_m k_m + z'} g^{w'}, & m_j &= g_2^{z_j} g^{w_j} \;\; 1 \leq j \leq n_m \end{aligned}$$

Note that these public parameters will have the same distribution as in the game between the challenger and $\mathcal{A}$. Furthermore, this assignment means that the master secret will be $g_2^{\alpha} = g_2^{a} = g^{ab}$ and that for any identity $\mathfrak{u}$ and message $\mathfrak{m}$, the equations

$$u' \prod_{i \in \mathcal{U}} u_i = g_2^{F(\mathfrak{u})} g^{J(\mathfrak{u})} \quad \text{and} \quad m' \prod_{j \in \mathcal{M}} m_j = g_2^{K(\mathfrak{m})} g^{L(\mathfrak{m})}$$

hold. All public parameters are passed to $\mathcal{A}$.

Queries When running the adversary, both extract and sign queries can occur. $\mathcal{B}$ answers these in the following way:

- Extract queries. Consider a query for the private key of an identity $\mathfrak{u}$. $\mathcal{B}$ does not know the master secret, but assuming $F(\mathfrak{u}) \neq 0 \mod p$, it can construct a private key by choosing $r_u \leftarrow_R \mathbb{Z}_p$ and computing:

$$d_\mathfrak{u} = (d_0, d_1) = \left(g_1^{-J(\mathfrak{u})/F(\mathfrak{u})} \left(u' \prod_{i \in \mathcal{U}} u_i \right)^{r_u}, \; g_1^{-1/F(\mathfrak{u})} g^{r_u} \right)$$

  Writing $\tilde{r}_u = r_u - a/F(v)$, it can be verified that defining $d_\mathfrak{u}$ in this manner yields a valid private key of $\mathfrak{u}$, since:

$$\begin{aligned} d_0 &= g_1^{-J(\mathfrak{u})/F(\mathfrak{u})} \left(u' \prod_{i \in \mathcal{U}} u_i \right)^{r_u} \\ &= g_1^{-J(\mathfrak{u})/F(\mathfrak{u})} (g_2^{F(\mathfrak{u})} g^{J(\mathfrak{u})})^{r_u} \\ &= g_2^a (g_2^{F(\mathfrak{u})} g^{J(\mathfrak{u})})^{-a/F(\mathfrak{u})} (g_2^{F(\mathfrak{u})} g^{J(\mathfrak{u})})^{r_u} \\ &= g_2^a (g_2^{F(\mathfrak{u})} g^{J(\mathfrak{u})})^{r_u - a/F(\mathfrak{u})} \\ &= g_2^a \left(u' \prod_{i \in \mathcal{U}} u_i \right)^{\tilde{r}_u} \end{aligned}$$

  and

$$d_1 = g_1^{-1/F(\mathfrak{u})} g^{r_u} = g^{r_u - a/F(\mathfrak{u})} = g^{\tilde{r}_u}.$$

  Hence, to the adversary, all private keys computed by $\mathcal{B}$ will be indistinguishable from the keys generated by a true challenger.

  If, on the other hand, $F(\mathfrak{u}) = 0 \mod p$, the above computation cannot be performed and the simulator will abort. To make the analysis of the simulation easier, we will force the simulator to abort whenever $F(\mathfrak{u}) = 0 \mod l_u$. Given the assumption $l_u(n_u+1) < p$ which implies $0 \leq l_u k_u < p$ and $0 \leq x' + \sum_{i \in \mathcal{U}} x_i < p$, it is easy to see that $F(\mathfrak{u}) = 0 \mod p$ implies that $F(\mathfrak{u}) = 0 \mod l_u$. Hence, $F(\mathfrak{u}) \neq 0 \mod l_u$ implies $F(\mathfrak{u}) \neq 0 \mod p$, so the former condition will be a sufficient requirement to ensure that a private key for $\mathfrak{u}$ can be constructed.
- Sign queries. Consider a query for a signature of $\mathfrak{u}$ on $\mathfrak{m}$ (it can be assumed, without loss of generality, that $\mathcal{A}$ has not made an extraction query on $\mathfrak{u}$). If $F(\mathfrak{u}) \neq 0 \mod l_u$, $\mathcal{B}$ can just construct a private key for $\mathfrak{u}$ as in an extract query, and then use the **Sign** algorithm to create a signature on $\mathfrak{m}$.

  If $F(\mathfrak{u}) = 0 \mod l_u$, $\mathcal{B}$ will try to construct a signature in a similar way to the construction of a private key in an extract query. Assume $K(\mathfrak{m}) \neq 0 \mod l_m$. Arguing as above, this implies $K(\mathfrak{m}) \neq 0 \mod p$, given the assumption $l_m(n_m + 1) < p$. The signature of $\mathfrak{u}$ on $\mathfrak{m}$ can now be constructed by picking $r_u, r_m \leftarrow_R \mathbb{Z}_p$ and computing

$$\begin{aligned} \sigma &= \left(\left(u' \prod_{i \in \mathcal{U}} u_i \right)^{r_u} g_1^{-L(\mathfrak{m})/K(\mathfrak{m})} \left(m' \prod_{j \in \mathcal{M}} m_i \right)^{r_m}, g^{r_u}, g_1^{-1/K(\mathfrak{m})} g^{r_m} \right) \\ &= \left(g_2^a \left(u' \prod_{i \in \mathcal{U}} u_i \right)^{r_u} \left(m' \prod_{j \in \mathcal{M}} m_j \right)^{\tilde{r}_m}, g^{r_u}, g^{\tilde{r}_m} \right), \end{aligned}$$

where $\tilde{r}_m = r_m - a/K(\mathfrak{m})$. This last equation shows that $\mathcal{B}$'s replies to $\mathcal{A}$'s sign queries are distributed as they would be in an interaction with a real challenger.

If $K(\mathfrak{m}) = 0 \mod l_m$, the simulator will simply abort.

Forgery If $\mathcal{B}$ does not abort as a consequence of one of the queries above, $\mathcal{A}$ will, with probability at least ϵ, return an identity $\mathfrak{u}^*$, a message $\mathfrak{m}^*$, and a valid forgery $\sigma^* = (V, R_u, R_m)$ of a signature of $\mathfrak{u}^*$ on $\mathfrak{m}^*$. If $F(\mathfrak{u}^*) \neq 0 \mod p$ or $K(\mathfrak{m}^*) \neq 0 \mod p$ then $\mathcal{B}$ will abort. If, on the other hand, $F(\mathfrak{u}^*) = 0 \mod p$ and $K(\mathfrak{m}^*) = 0 \mod p$, $\mathcal{B}$ computes and outputs

$$\begin{aligned}\frac{V}{R_u^{J(\mathfrak{u}^*)} R_m^{L(\mathfrak{m}^*)}} &= \frac{g_2^a \left(u' \prod_{i\in\mathcal{U}} u_i\right)^{r_u} \left(m' \prod_{j\in\mathcal{M}} m_j\right)^{r_m}}{g^{J(\mathfrak{u}^*) r_u} g^{L(\mathfrak{m}^*) r_m}} \\ &= g_2^a \\ &= g^{ab}\end{aligned}$$

which is the solution to the given CDH problem.

This completes the description of the simulation. It remains to analyse the probability of $\mathcal{B}$ not aborting. For the simulation to complete without aborting, we require that all extraction queries on an identity $\mathfrak{u}$ have $F(\mathfrak{u}) \neq 0 \mod l_u$, that all sign queries $(\mathfrak{u}, \mathfrak{m})$ will either have $F(\mathfrak{u}) \neq 0 \mod l_u$ or $K(\mathfrak{m}) \neq 0 \mod l_m$, and that $F(\mathfrak{u}^*) = 0 \mod l_u$ and $K(\mathfrak{m}^*) = 0 \mod l_m$. However, to make the analysis simpler, we will bound the probability of a subcase of this event. More specifically, we will divide the sign queries into two groups – queries involving $\mathfrak{u}^*$ and queries involving identities $\mathfrak{u} \neq \mathfrak{u}^*$ – and then consider the event that all identities $\mathfrak{u}$ have $F(\mathfrak{u}) \neq 0 \mod l_u$, ignoring that sign queries $(\mathfrak{u}, \mathfrak{m})$ can be answered if $F(\mathfrak{u}) = 0 \mod l_u$ and $K(\mathfrak{m}) \neq 0 \mod l_m$. Thus we will provide a lower bound on the probability that $\mathcal{B}$ aborts.

Let $\mathfrak{u}_1, \ldots, \mathfrak{u}_{q_I}$ be the identities appearing in either extract queries or in sign queries not involving the challenge identity and let $\mathfrak{m}_1, \ldots, \mathfrak{m}_{q_M}$ be the messages in the sign queries involving the challenge identity $\mathfrak{u}^*$. Clearly, we will have $q_I \leq q_e + q_s$ and $q_M \leq q_s$. Define the events A_i, A^*, B_j and B^* as

$$\begin{aligned} A_i &: F(\mathfrak{u}_i) \neq 0 \mod l_u \\ A^* &: F(\mathfrak{u}^*) = 0 \mod p \\ B_j &: K(\mathfrak{m}_j) \neq 0 \mod l_m \\ B^* &: K(\mathfrak{m}^*) = 0 \mod p \end{aligned}$$

From the analysis above, the probability of $\mathcal{B}$ not aborting is

$$\Pr[\neg\texttt{abort}] \geq \Pr[\bigwedge_{i=1}^{q_I} A_i \wedge A^* \wedge \bigwedge_{j=1}^{q_M} B_j \wedge B^*].$$

It is easy to see that the events $(\bigwedge_{i=1}^{q_I} A_i \wedge A^*)$ and $(\bigwedge_{j=1}^{q_M} B_j \wedge B^*)$ are independent. Essentially, this is because the functions F and K which define these events are selected independently and are hidden from the adversary's view of the simulation.

As seen above, the assumption $l_u(n_u+1) < p$ leads to the implication $F(\mathfrak{u}) = 0 \mod p \Rightarrow F(\mathfrak{u}) = 0 \mod l_u$. Furthermore, this assumption gives that if $F(\mathfrak{u}) = 0 \mod l_u$, there will be an unique choice of k_u with $0 \leq k_u \leq n_u$ such that $F(\mathfrak{u}) = 0 \mod p$. Since k_u and $x', \boldsymbol{X}$ are randomly chosen, we have

$$\begin{aligned}
\Pr[A^*] &= \Pr[F(\mathfrak{u}^*) = 0 \mod p \wedge F(\mathfrak{u}^*) = 0 \mod l_u] \\
&= \Pr[F(\mathfrak{u}^*) = 0 \mod l_u] \Pr[F(\mathfrak{u}^*) = 0 \mod p | F(\mathfrak{u}^*) = 0 \mod l_u] \\
&= \frac{1}{l_u} \frac{1}{n_u + 1}
\end{aligned}$$

We also have that

$$\begin{aligned}
\Pr[\bigwedge_{i=1}^{q_I} A_i | A^*] &= 1 - \Pr[\bigvee_{i=1}^{q_I} \neg A_i | A^*] \\
&\geq 1 - \sum_{i=1}^{q_I} \Pr[\neg A_i | A^*]
\end{aligned}$$

If F is evaluated on two different identities, $\mathfrak{u}_1$ and $\mathfrak{u}_2$, then the sums appearing in $F(\mathfrak{u}_1)$ and $F(\mathfrak{u}_2)$ will differ in at least one randomly chosen value, and the events $F(\mathfrak{u}_1) = 0 \mod l_u$ and $F(\mathfrak{u}_1) = 0 \mod l_u$ will be independent. As a special case, the events A_i and A^* are independent for any i, and $\Pr[\neg A_i | A^*] = 1/l_u$. Hence, we have

$$\begin{aligned}
\Pr[\bigwedge_{i=1}^{q_I} A_i \wedge A^*] &= \Pr[A^*] \Pr[\bigwedge_{i=1}^{q_I} A_i | A^*] \\
&= \frac{1}{l_u(n_u + 1)} \left(1 - \frac{q_I}{l_u}\right) \\
&\geq \frac{1}{l_u(n_u + 1)} \left(1 - \frac{q_e + q_s}{l_u}\right)
\end{aligned}$$

and setting $l_u = 2(q_e + q_s)$ as in the simulation gives

$$\Pr[\bigwedge_{i=1}^{q_I} A_i \wedge A^*] \geq \frac{1}{4(q_e + q_s)(n_u + 1)}.$$

A similar analysis for the sign queries gives the result

$$\Pr[\bigwedge_{j=1}^{q_M} B_j \wedge B^*] \geq \frac{1}{4q_s(n_m + 1)}$$

and we get that

$$\begin{aligned}
\Pr[\neg\texttt{abort}] &\geq \Pr[\bigwedge_{i=1}^{q_I} A_i \wedge A^*] \Pr[\bigwedge_{j=1}^{q_M} B_j \wedge B^*] \\
&\geq \frac{1}{16(q_e + q_s) q_s (n_u + 1)(n_m + 1)}
\end{aligned}$$

If the simulation does not abort, $\mathcal{A}$ will create a valid forgery with probability at least ϵ. The algorithm $\mathcal{B}$ can then compute g^{ab} from the forgery as shown above.

The time complexity of the algorithm $\mathcal{B}$ is dominated by the exponentiations and, for larger values of n_u and n_m, multiplications performed in the extract and sign queries. Since there are $O(n_u)$ and $O(n_u + n_m)$ multiplications and $O(1)$ and $O(1)$ exponentiations in the extract and sign stage respectively, the time complexity of $\mathcal{B}$ is $t + O\big((q_e n_u + q_s(n_u + n_m))\rho + (q_e + q_s)\tau\big)$. Thus, the theorem follows. □

6 Trading security for efficiency

In a practical scheme, the n_u-bit identities and the n_m-bit messages will most likely be outputs from collision resistant hash functions. This suggests that n_u and n_m should be at least 160, which means that the public parameters of our scheme will contain at least 325 group elements of $\mathbb{G}$. When $\mathbb{G}$ is chosen such that the CDH problem in $\mathbb{G}$ is considered to be hard, the public parameters will grow to a size which is not suitable for environments with limited storage capacity. However, Naccache [Nac05] and Sarkar-Chatterjee [SC05] independently suggested a modification to Waters' scheme to reduce the size of the public parameters. This modification is also applicable to our signature scheme.

6.1 The technique of Naccache and Sarkar-Chatterjee

In our signature scheme, when an entity signs a message $\mathfrak{m} \subset \{0,1\}^{n_m}$, he computes the product

$$m' \prod_{j \in \mathcal{M}} m_j$$

where $\mathcal{M} \subset \{1, \ldots, n_m\}$ is the set of indicies j such that the jth bit of $\mathfrak{m}$ is 1. The idea is to consider the message as a set of concatenated t-bit integers instead of a set of concatenated bits, i.e. $\mathfrak{m} = \mathfrak{m}[0]||\cdots||\mathfrak{m}[n'_m]$ where $n'_m = n_m/t$ and $\mathfrak{m}[j] \in \mathbb{Z}_{2^t}$, and then replace the above product with

$$m' \prod_{j=1}^{n'_m} m_j^{\mathfrak{m}[j]}.$$

This will reduce the size of $\boldsymbol{M}$ by a factor of $n_m/n'_m = t$ and the number of group elements included in the public parameters `params` will be reduced to $n_u + n_m/t + 5$.

Likewise, an identity $\mathfrak{u} \subset \{0,1\}^{n_u}$ can be considered as a concatenation of n'_u s-bit integers and by replacing the product of elements from $\boldsymbol{U}$ in a similar way as above, the size of $\boldsymbol{U}$ can be reduced by a factor of $n_u/n'_u = s$. Applying both modifications, the number of group elements in `params` can be reduced to $n_u/s + n_m/t + 5$.

6.2 Security of our modified scheme

The security analysis of our scheme, when using the above idea, is very similar to the analysis presented in Section 5. However, a few modifications to the construction of F, J, K, and L are required to ensure that these functions continue to have the properties needed in Section 5. We will assume that the same setup as in Section 5 is given and only focus on the changes needed to make the security analysis valid for our modified scheme. As in the above, we will assume that identities and messages consist of n'_u s-bit and n'_m t-bit integers respectively.

The first change is that the ranges within which the values k_u and k_m are chosen in the setup stage of the simulation, are expanded to $0 \leq k_u \leq 2^{s-1}n'_u$ and $0 \leq k_m \leq 2^{t-1}n'_m$. All other chosen values and assignments are the same as in the original setup stage of the simulation. We will assume that $l_u(2^{s-1}n'_u + 1) < p$ and $l_m(2^{t-1}n'_m + 1) < p$. The functions

F, J, K and L are then redefined as

$$F(\mathfrak{u}) = x' + \sum_{i=1}^{n'_u} \mathfrak{u}[i]x_i - l_u k_u \quad \text{and} \quad J(\mathfrak{u}) = y' + \sum_{i=1}^{n'_u} \mathfrak{u}[i]y_i,$$

$$K(\mathfrak{m}) = z' + \sum_{j=1}^{n'_m} \mathfrak{m}[j]z_j - l_m k_m \quad \text{and} \quad L(\mathfrak{m}) = w' + \sum_{j=1}^{n'_m} \mathfrak{m}[j]w_j$$

where $\mathfrak{u}[i]$ and $\mathfrak{m}[j]$ denote the s- and t-bit integers making up $\mathfrak{u}$ and $\mathfrak{m}$ respectively. It is easy to see that these modifications ensure that the following hold:

- $u' \prod_{i=1}^{n'_u} u_i^{\mathfrak{u}[i]} = g_2^{F(\mathfrak{u})} g^{J(\mathfrak{u})}$ and $m' \prod_{j=1}^{n'_m} m_j^{\mathfrak{m}[j]} = g_2^{K(\mathfrak{m})} g^{L(\mathfrak{m})}$
- $F(\mathfrak{u}) = 0 \mod p$ implies $F(\mathfrak{u}) = 0 \mod l_u$ and $K(\mathfrak{m}) = 0 \mod p$ implies $K(\mathfrak{m}) = 0 \mod l_m$
- If $F(\mathfrak{u}) = 0 \mod l_u$ then there is a unique choice of k_u with $0 \le k_u \le 2^{s-1}n'_u$ such that $F(\mathfrak{u}) = 0 \mod p$. Similarly, if $K(\mathfrak{m}) = 0 \mod l_m$ then there is a unique choice of k_m with $0 \le k_m \le 2^{t-1}n'_m$ such that $K(\mathfrak{m}) = 0 \mod p$.

With these properties, the other stages of the simulation for the modified scheme can be carried out just as described in the original simulation in Section 5.

The analysis of the success probability of the simulation is almost identical to the analysis in Section 5, since only the increased range of the values k_u and k_m (i.e. the last property listed above) affects the treated probabilities. This changes the probability of the events A^* and B^* when defined with the modified F and K functions and we get

$$\Pr[A^*] = \frac{1}{l_u(2^{s-1}n'_u + 1)} \quad \text{and} \quad \Pr[B^*] = \frac{1}{l_m(2^{t-1}n'_m + 1)}.$$

Since the probabilities of the events $(\bigwedge_{i=1}^{q_I} A_i | A^*)$ and $(\bigwedge_{j=1}^{q_M} B_j | B^*)$ do not change with the modifications to F and K, the success probability of the simulation is

$$\Pr[\neg\texttt{abort}] \ge \frac{1}{16(q_e + q_s)q_s(2^{s-1}n'_u + 1)(2^{t-1}n'_m + 1)}.$$

This is approximately a factor of $2^{s-1}2^{t-1}/(st)$ lower than the success probability of the simulation in Section 5.

The time complexity of the simulation remain as $t + O\big((q_e n'_u + q_s(n'_u + n'_m))\rho + (q_e + q_s)\tau\big)$ where t is the time taken by the adversary, ρ is the time for a multiplication in $\mathbb{G}$ and τ is the time for an exponentiation in $\mathbb{G}$.

6.3 Tradeoffs between size, computation and security

The above result means that we can reduce the number of elements in the public parameters to $n_u/s + n_m/t + 5$, but this will be at the cost of a loss in security of $s + t - 2 - \log_2(st)$ bits compared to the original scheme. For small values of s and t, it may be acceptable simply to trade the loss of security for the increased efficiency, which is the approach suggested by Naccache.

However, it is possible to avoid the loss of security by increasing the computational cost of the scheme. The idea, which was suggested by Sarkar-Chatterjee, is to increase the size of

$\mathbb{G}$ to increase the security level provided by the CDH problem in $\mathbb{G}$ to compensate for the loss of security in the security proof caused by a given choice of s and t values. By choosing $|\mathbb{G}|$ carefully, it is possible to maintain a given security level for any choice of s and t. However, there are several factors to take into account when this approach is taken.

First of all, the level of security provided by the CDH problem in $\mathbb{G}$ will need to be estimated. Currently, the best known way of solving the CDH problem is by solving the discrete logarithm problem (DLP). The DLP in $\mathbb{G}$ can easily be reduced to the DLP in $\mathbb{G}_T$ and hence, the best known algorithms for solving the DLP in both $\mathbb{G}$ and $\mathbb{G}_T$ will need to be considered when choosing these groups. Secondly, the availability of a suitable class of elliptic curves that enables the construction of a pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ with $\mathbb{G}$ and $\mathbb{G}_T$ satisfying the above requirements, will need to be considered. We note that the authors of [SC05] do not consider this issue in their analysis. Finally, the increase in size of $\mathbb{G}$ and $\mathbb{G}_T$ will lead to larger space requirements for a single group element and will increase the complexity of the arithmetic in these groups.

All of these issues are important for evaluating the efficiency and security of the scheme when the size of the public parameters is reduced. However, we do not include the detailed analysis here.

7 Scheme construction using general curves

Currently, the only known way of constructing a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is by using a Weil or Tate pairing on an elliptic curve. Furthermore, if $\mathbb{G}_1 = \mathbb{G}_2$, as we assume in our construction, we will be limited to using supersingular elliptic curves, a very limited class of curves. However, our scheme can easily be generalised to work with a bilinear map of the form $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, allowing the use of a wider class of elliptic curves. This flexibility is important for implementation and for the selection of parameters meeting a particular concrete security level. In the following we will highlight the changes needed to generalise our scheme and sketch how a security proof for the modified scheme can be constructed.

Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be an admissible map where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are of prime order p and let g and h be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. By admissible here, we mean that e is bilinear, non-degenerate and efficiently computable. For our security proof to work, we also require that an efficiently computable isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ exists and that $\psi(h) = g$. This isomorphism is only needed in the security proof and will not be used in the scheme itself. The generalised setting will also change the problem underlying the security reduction from the CDH problem in $\mathbb{G}$ to the more general co-CDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$, which is defined as follows:

Definition 4. *Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of prime order p and with generators $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$. Given g, h and h^s where s is selected uniformly at random in $\mathbb{Z}_p$, the co-CDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ is to compute g^s.*

We assume that the admissible map used in our scheme is defined by a pairing on the elliptic curve E over the finite field $\mathbb{F}_q$ for some prime power q. Then we can also assume that $\mathbb{G}_1 \subset E(\mathbb{F}_q)$, $\mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$ and $\mathbb{G}_T \subset \mathbb{F}_{q^k}$ where k is the embedding degree for E and p. Hence, the size of the description of an element in $\mathbb{G}_2$ will be larger than the description of an element in $\mathbb{G}_1$ and the efficiency of our scheme, in terms of the size of a signature and the public parameters, will be dependent on the group in which we place the different group elements of the scheme. As argued above, the size of the public parameters may be a concern and we will therefore choose the placement of elements such that the size of `params` is minimised.

When initialising our scheme, we pick a secret $\alpha \leftarrow_R \mathbb{Z}_p$. In the generalised setting we then compute $g_1 = h^\alpha \in \mathbb{G}_2$ and pick $g_2, u', m' \leftarrow_R \mathbb{G}_1$, $\boldsymbol{U} \leftarrow_R \mathbb{G}_1^{n_u}$ and $\boldsymbol{M} \leftarrow_R \mathbb{G}_1^{n_m}$. The group elements of the public parameters are $(g, h, g_1, g_2, u', \boldsymbol{U}, m', \boldsymbol{M}) \in \mathbb{G}_1 \times \mathbb{G}_2^2 \times \mathbb{G}_1^{2+n_u+n_m}$ and the master secret is $g^\alpha \in \mathbb{G}_1$. Replacing all occurrences of g with the generator h in the algorithms **Extract** and **Sign** in the original scheme leads to private keys and signatures being elements of $\mathbb{G}_1 \times \mathbb{G}_2$ and $\mathbb{G}_1 \times \mathbb{G}_2^2$, respectively, and enables verification by the original **Verify** algorithm.

To prove this scheme secure, we will modify the algorithm $\mathcal{B}$ to solve an instance of the co-CDH problem given an adversary of the scheme. The challenge to $\mathcal{B}$ consists of computing $g^s \in \mathbb{G}_1$ given $g \in \mathbb{G}_1$ and $h, h^s \in \mathbb{G}_2$ where $s \leftarrow_R \mathbb{Z}_p$. $\mathcal{B}$ sets $g_1 = h^s$, $g_2 = g$ and computes u', $\boldsymbol{U}$, m' and $\boldsymbol{M}$ as in the proof of Theorem 1, replacing the original g with the generator of $\mathbb{G}_1$. Note that the functions $F(\mathfrak{u})$, $J(\mathfrak{u})$, $K(\mathfrak{m})$ and $L(\mathfrak{m})$ can be defined just as in Section 5. By picking $r_u, r_m \leftarrow_R \mathbb{Z}_p$, $\mathcal{B}$ can answer extract and sign queries by computing

$$d_{\mathfrak{u}} = \left(\psi(g_1)^{-J(\mathfrak{u})/F(\mathfrak{u})} \left(u' \prod_{i \in \mathcal{U}} u_i \right)^{r_u}, \; g_1^{-1/F(\mathfrak{u})} h^{r_u} \right)$$

and

$$\sigma = \left(\left(u' \prod_{i \in \mathcal{U}} u_i \right)^{r_u} \psi(g_1)^{-L(\mathfrak{m})/K(\mathfrak{m})} \left(m' \prod_{j \in \mathcal{M}} m_i \right)^{r_m}, \; h^{r_u}, g_1^{-1/K(\mathfrak{m})} h^{r_m} \right),$$

assuming $F(\mathfrak{u}) \neq 0$ and $K(\mathfrak{m}) \neq 0$, respectively. Given a valid forgery $\sigma^* = (V, R_u, R_m)$ produced by the adversary, $\mathcal{B}$ can compute the solution to the co-CDH problem by

$$g^s = \frac{V}{\psi(R_u)^{J(\mathfrak{u}^*)} \psi(R_m)^{L(\mathfrak{m}^*)}},$$

assuming $F(\mathfrak{u}^*) = 0$ and $K(\mathfrak{m}^*) = 0$. The probability analysis of the simulation in Section 5 is not affected by the described modifications and so is valid for the above simulation as well. Thus, we have established that the generalised scheme is secure, given that the co-CDH problem on $(\mathbb{G}_1, \mathbb{G}_2)$ is hard.

In an alternative approach, we can minimise the signature size by placing $g_2, u', \boldsymbol{U}, m'$ and $\boldsymbol{M}$ in $\mathbb{G}_2$ and carefully choosing the appropriate generator to be used in the computations. However, regardless of how we choose to place the elements, the techniques of Section 6 can be applied to reduce the number of public parameters.

8 Aggregation properties

In this section we briefly consider the aggregation properties of the concrete scheme of Section 4. An aggregate signature scheme, defined by Boneh, Gentry, Lynn and Shacham [BGLS03], provides a method for combining t signatures on t messages from t users into a single short aggregate signature. This aggregation can be performed by any entity, even entities untrusted by all users, and the resulting aggregate signature will convince any verifier that the t users did indeed sign the t original messages. Boneh, Gentry, Lynn and Shacham proposed an aggregate (non identity-based) signature scheme provably secure in the random oracle model.

Since the first definition and construction of an aggregate signature scheme were proposed, efforts have been made to construct an identity-based aggregate signature scheme [CKY04,Her05]. However, the problem of constructing a scheme enjoying full aggregation where the aggregate signature is of the same length as the individual signatures and where no cooperation between the signers is needed, has not yet been solved.

Our scheme allows a form of aggregation in which multiple signatures from different signers on a single message can be combined into an aggregate signature with a more efficient representation than the original set of signatures. The resulting aggregate signature is more commonly know as a multisignature. The computational cost of aggregate verification is also lower than the cost of individual verification of all the signatures in the set. The aggregation and verification of an aggregated signature are provided by the algorithms **Aggregate** and **Aggregate Verify** defined below.

Aggregate Let $\sigma_1, \ldots, \sigma_t$, $\sigma_i = (V_i, R_{u_i}, R_{m_i})$ be signatures, constructed by the identities $\mathfrak{u}_1, \ldots, \mathfrak{u}_t$, on a single message $\mathfrak{m}$. The aggregator computes an aggregated signature σ_a as

$$\sigma_a = \left(\prod_{i=1}^{t} V_i, R_{u_1}, \ldots, R_{u_t}, \prod_{i=1}^{t} R_{m_i} \right) \in \mathbb{G}^{t+2}$$

Aggregate Verify Let $\sigma_a = (V_a, R_{u_1}, \ldots, R_{u_t}, R_{m_a})$ be a purported aggregate signature of the identities $\mathfrak{u}_1, \ldots, \mathfrak{u}_t$ on a message $\mathfrak{m}$. A verifier accepts σ_a if the following equation holds true:

$$e(V_a, g) = e(g_2, g_1)^t e\Bigl(m' \prod_{k \in \mathcal{M}} m_k, R_{m_a}\Bigr) \prod_{i=1}^{t} e\Bigl(u' \prod_{j \in \mathcal{U}_i} u_j, R_{u_i}\Bigr)$$

where $\mathcal{U}_i \subset \{1, \ldots, n_u\}$ is the set of indicies j such that the jth bit of $\mathfrak{u}_i$ is 1.

This aggregation will roughly reduce the space requirements by a factor of 3 and the computational requirements by a factor of 4 as compared to storing and verifying individual signatures.

Herranz [Her05] defined a security model for identity-based aggregate signature schemes by extending the model in [BGLS03]. It is easy to see that the above aggregation is secure in Herranz's model, assuming that users do not re-randomise their private keys. The key observation is that given a forgery of an aggregate signature, it is possible to extract a forgery of a signature of the original signature scheme and thereby break the existential unforgeability proven in Section 5.

9 Conclusion

We have presented what is, as far as we know, the first identity-based signature scheme that is provably secure in the standard model. Our basic scheme is computationally efficient, and we have presented a variety of techniques to improve its space requirements and to increase the range of parameter choices. We have also briefly analysed the aggregation properties of our scheme. The problem of constructing a secure identity-based signature scheme enjoying full aggregation remains an open problem.

References

[Bar] Paulo Barreto. The pairing-based crypto lounge. http://paginas.terra.com.br/informatica/paulobarreto/pblounge.html.

[BB04a] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In Cachin and Camenisch [CC04], pages 223–238.

[BB04b] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer, 2004.

[BB04c] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Cachin and Camenisch [CC04], pages 56–73.

[BBP04] Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In Cachin and Camenisch [CC04], pages 171–188.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[BGLS03] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer, 2003.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *J. Cryptology*, 17(4):297–319, 2004.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73, New York, NY, USA, 1993. ACM Press.

[CC03] Jae Choon Cha and Jung Hee Cheon. An identity-based signature from gap Diffie-Hellman groups. In Yvo Desmedt, editor, *Public Key Cryptography 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 18–30. Springer, 2003.

[CC04] Christian Cachin and Jan Camenisch, editors. *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*. Springer, 2004.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *STOC*, pages 209–218, 1998.

[CKY04] Jung Hee Cheon, Yongdae Kim, and Hyo Jin Yoon. A new ID-based signature with batch verification. Cryptology ePrint Archive, Report 2004/131, 2004. http://eprint.iacr.org/.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *IMA Int. Conf.*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer, 2001.

[Gal05] S. D. Galbraith. Pairings. In G. Seroussi I.F. Blake and N.P. Smart, editors, *Advances in Elliptic Curve Cryptography*, volume 317 of *Lecture Note Series*, pages 183–212. Cambridge University Press, 2005.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.

[Her05] Javier Herranz. Deterministic identity-based signatures for partial aggregation. Cryptology ePrint Archive, Report 2005/313, 2005. http://eprint.iacr.org/.

[Hes02] Florian Hess. Efficient identity based signature schemes based on pairings. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 310–324. Springer, 2002.

[Nac05] David Naccache. Secure and *practical* identity-based encryption. Cryptology ePrint Archive, Report 2005/369, 2005. http://eprint.iacr.org/.

[Pat02] Kenneth G. Paterson. ID-based signatures from pairings on elliptic curves. *IEEE Communications Letters*, 38(18):1025–1026, 2002.

[SC05] Palash Sarkar and Sanjit Chatterjee. Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. *Proceedings of ICISC*, 2005. To appear.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 84*, pages 47–53, 1984.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.

[Yi03] Xun Yi. An identity-based signature scheme from the Weil pairing. *IEEE Communications Letters*, 7(2), 2003.