A Schnorr-Like Lightweight Identity-Based Signature Scheme

David Galindo[1,⋆] and Flavio D. Garcia[2,⋆⋆]

[1] University of Luxembourg
david.galindo@uni.lu
[2] Institute for Computing and Information Sciences,
Radboud University Nijmegen, The Netherlands
flaviog@cs.ru.nl

Abstract. The use of concatenated Schnorr signatures [Sch91] for the hierarchical delegation of public keys is a well-known technique. In this paper we carry out a thorough analysis of the identity-based signature scheme that this technique yields. The resulting scheme is of interest since it is intuitive, simple and does not require pairings. We prove that the scheme is secure against existential forgery on adaptive chosen message and adaptive identity attacks using a variant of the Forking Lemma [PS00]. The security is proven in the Random Oracle Model under the discrete logarithm assumption. Next, we provide an estimation of its performance, including a comparison with the state of the art on identity-based signatures. We draw the conclusion that the Schnorr-like identity-based signature scheme is arguably the most efficient such scheme known to date.

Keywords: identity-based signature, lightweight cryptography, provable security, Schnorr, random oracle model.

1 Introduction

Digital signatures are fundamental primitives in public key cryptography. They ensure the authenticity of the originator of a digital document and the integrity of that document, while they also prevent that the originator can repudiate that very same document later on. A signature on a given bit-string is valid if it passes the associated verification test, which takes as inputs a verification key, a purported signature, and a document. In traditional signature schemes the verification key is a mathematical object taken at random from some set. An

⋆ Work done while at University of Malaga. Partially funded by the Spanish Ministry of Science and Education through the projects ARES (CSD2007-00004) and CRISIS (TIN2006-09242).

⋆⋆ partially supported by the research program Sentinels (www.sentinels.nl), project PEARL (7639). Sentinels is being financed by Technology Foundation STW, the for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

B. Preneel (Ed.): AFRICACRYPT 2009, LNCS 5580, pp. 135–148, 2009.

external binding between the verification key and the signing entity is therefore needed. This binding takes the form of a certificate, which is created by a Certification Authority.

The concept of identity-based signature (IBS) was introduced by Shamir [Sha85]. The idea is that the identity of the signer is used as the verification key. This dispenses with the need of an external binding. The identity can be any string that singles out the entity, for instance, a social security number or an IP address. Identity based systems have a drawback, namely, the entity cannot build the signing key by itself. Instead, a Trusted Third Party (TTP) is in charge of assigning and delivering, via a confidential channel, the secret key to each user. This in sharp contrast to the traditional setting, where the secret key is generated by the user itself and kept secret. Still scenarios where a TTP creates a signing/verification key pair on behalf of the user are quite often found in practice. This is the case of the Spanish Electronic Identity Card eDNI [oIA08, EG08] and the digital signature token services provided by DigiNotar [Dig08]. There, the key pair is stored together with a certificate in a smart card to be owned by the relevant user.

Several IBS schemes based on factoring or RSA were proposed in the late eighties and early nineties, for instance [Sha85, FS87, GQ90, Oka93] to name just a few. After the revival of research on identity-based cryptography due to the use of pairings (cf. [BSS05] Chapter 5 for an introduction to pairings), many other IBS schemes have been proposed, for instance [SOK00, Hes03, CC02]. As a consequence of this revival, Bellare, Namprempre and Neven [BNN04] provided a framework for deriving security proofs for identity-based signature and identification schemes, and compiled most previous work on IBS. The most efficient scheme of those studied in [BNN04] turns out to be an IBS scheme obtained from a conventional signature scheme by Beth [Bet88]. Unfortunately, Beth's scheme lacks a security proof.

Our contribution. This work studies the identity-based signature scheme resulting from sequentially delegating Schnorr signatures [Sch91]. Such a technique is certainly well-known and it has been in use for many years and in different contexts, e.g. [Gir91, PH97, AO99, CJT04, BSNS05, BFPW07]. We prove that the scheme is secure against existential forgery on adaptive chosen message and adaptive identity attacks using a variant of the Forking Lemma [PS00] by Boldyreva, Palacio and Warinschi [BPW03]. The security is proven in the Random Oracle Model under the discrete logarithm assumption. We show that the resulting scheme is among the most efficient provably-secure IBS schemes known to date, be it based on factoring, discrete logarithm or pairings. In particular it has the same performance that the aforementioned Beth IBS scheme. This makes it attractive for application in resource-constrained environments where saving in computation, communication and implementation code area are a premium.

Organization of this paper. Section 2 introduces standard definitions from the literature. Section 3 describes our new identity-based signature scheme. In

Section 4 we prove the security of the scheme. Section 5 compares the computational and length complexity of our scheme to other schemes from the literature. Finally Section 6 concludes the paper.

2 Preliminaries

This section introduces the syntax and security definitions of identity-based signatures and the discrete logarithm assumption. Most of it is standard, we refer the reader to [BNN04] for a thorough explanation. We introduce some basic notation. If S is a set then $s_1, \ldots, s_n \xleftarrow{\$} S$ denotes the operation of picking n elements s_i of S independently and uniformly at random. We write $\mathcal{A}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $x, y, \ldots$ and by $z \leftarrow \mathcal{A}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and letting z be the output.

2.1 Identity-Based Signatures

An *identity-based signature scheme* (IBS) is a quadruple $(\mathcal{G}, \mathcal{E}, \mathcal{S}, \mathcal{V})$ of probabilistic polynomial-time algorithms, where

- $\mathcal{G}$ is the *parameter-generation algorithm*. It takes as input the security parameter η (a positive integer and outputs the system public parameters mpk and the master secret key msk.
- $\mathcal{E}$ is the *key-extraction algorithm* that takes as input parameters mpk, a master key msk and an identity id and outputs a private key $\mathsf{sk}_{\mathsf{id}}$ corresponding to the user with this identity.
- $\mathcal{S}$ is a *signing algorithm* that takes as input parameters mpk, a private key $\mathsf{sk}_{\mathsf{id}}$ and a message m and outputs a signature σ.
- $\mathcal{V}$ is a deterministic *signature verification algorithm* that takes as input parameters mpk, a signature σ, a message m and an identity id and outputs whether or not σ is a valid signature of m relative to $(\mathsf{mpk}, \mathsf{id})$.

Definition 1 (EUF-IBS-CMA). An identity-based signature scheme $\Sigma = (\mathcal{G}, \mathcal{E}, \mathcal{S}, \mathcal{V})$ is said to be secure against *existential forgery on adaptively chosen message and identity attacks* if for all probabilistic polynomial-time adversaries $\mathcal{A}$, the probability of the experiment $\mathbf{EUF\text{-}IBS\text{-}CMA}_{\Sigma}(\mathcal{A}) = 1$ defined below is a negligible function of η. During this experiment $\mathcal{A}$ has access to two oracles: a key-extraction oracle $\mathcal{O}_{\mathcal{E}}$ that takes as input an identity id and outputs $\mathcal{E}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$; and a signature oracle $\mathcal{O}_{\mathcal{S}}$ that takes as input an identity id and a message m and returns a signature $\mathcal{S}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, m)$.

$\mathbf{EUF\text{-}IBS\text{-}CMA}_{\Sigma}(\mathcal{A})$:
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathcal{G}(\eta)$
$(\mathsf{id}^\star, m^\star, \sigma^\star) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{E}}(\cdot), \mathcal{O}_{\mathcal{S}}(\cdot,\cdot)}(\mathsf{mpk})$
return $\mathcal{V}(\mathsf{mpk}, \sigma^\star, m^\star, \mathsf{id}^\star)$

Some restrictions apply. In particular, it is required that $\mathsf{id}^\star$ and $(\mathsf{id}^\star, m^\star)$ are not equal to any query made to the oracles $\mathcal{O}_{\mathcal{E}}(\cdot)$ and $\mathcal{O}_{\mathcal{S}}(\cdot,\cdot)$ respectively, and that the same id cannot be queried to $\mathcal{O}_{\mathcal{E}}(\cdot)$ twice (see [BNN04] for details).

Definition 2 (Discrete Logarithm Assumption). We say that a group generation function $(\mathcal{G}, g, q) \leftarrow \mathsf{Gen}(\eta)$ generates DL-secure groups if for all probabilistic polynomial-time algorithms $\mathcal{A}$, the probability

$$\mathbb{P}[(\mathcal{G}, g, q) \leftarrow \mathsf{Gen}(\eta); a \stackrel{\$}{\leftarrow} \mathbb{Z}_q : a \leftarrow \mathcal{A}(\mathcal{G}, q, g, g^a)]$$

is a negligible function of η.

3 The Construction

The idea behind the construction is to use two concatenated Schnorr signatures [Sch91]. This technique is certainly not new and has been used elsewhere [Gir91, PH97, AO99, CJT04, BSNS05, BFPW07]. This said, the benefits of this technique in terms of efficiency and simplicity for the design of identity-based signatures seem to have gone unnoticed in the relevant literature as far as we are aware of (see Section 5).

Roughly speaking, the scheme works as follows. Firstly, the TTP produces a Schnorr signature on the identity of the user by using the master secret key. This signature implicitly defines a unique Schnorr-like public key for which only the user knows the corresponding private key. Next, the user builds a second Schnorr-like signature, this time on the message, by using its private key. The verification of a signature on message m under identity id implicitly checks whether the two concatenated Schnorr signatures are correct. Details are given below.

- The global parameters generation algorithm $\mathcal{G}$ on input η outputs public parameters mpk and a master secret key msk where: $(\mathcal{G}, g, q)$ is generated by calling the group generation algorithm Gen on input η. $\mathcal{G}$ is the description of a group of prime order $q = q(\eta)$ with $2^\eta \leq q < 2^{\eta+1}$ and g is a generator of $\mathcal{G}$. $G\colon \{0,1\}^* \to \mathbb{Z}_q$, $H\colon \{0,1\}^* \to \mathbb{Z}_q$ are descriptions of hash functions. Let z be a random element from $\mathbb{Z}_q$ and set $(\mathsf{mpk}, \mathsf{msk}) = \big((\mathcal{G}, g, q, g^z, G, H), z\big)$.
- The key-extraction algorithm $\mathcal{E}$ on input global parameters mpk, a master secret key $\mathsf{msk} = z$ and an identity id, picks $r \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ and sets $y = r + z \cdot H(g^r, \mathsf{id}) \mod q$. Then it outputs the secret key $\mathsf{sk}_{\mathsf{id}} = (y, g^r)$. Note that g^r is actually public information even though it is part of the secret key.
- The signing algorithm $\mathcal{S}$ on input parameters mpk, user private key $\mathsf{sk}_{\mathsf{id}} = (y, g^r)$ and a message m proceeds as follows. It selects $a \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ and computes $b = a + y \cdot G(\mathsf{id}, g^a, m)$. Then it outputs the signature $\sigma = (g^a, b, g^r)$.
- The verification algorithm $\mathcal{V}$ on input parameters $\mathsf{mpk} = (\mathcal{G}, g, q, G, H, g^z)$, a signature $\sigma = (g^a, b, g^r)$, a message m and an identity id proceeds as follows. It outputs whether or not the equation $g^b = g^a (g^r g^{zc})^d$ holds, where $c = H(g^r, \mathsf{id})$ and $d = G(\mathsf{id}, g^a, m)$.

4 Security of the Construction

This section analyzes the security of the proposed scheme in the random oracle model.

Theorem 3. *The construction above is secure with respect to Definition 1, in the random oracle model, if the group generation function* Gen *generates discrete logarithm secure groups. More precisely, there exist adversaries* $\mathcal{B}_1, \mathcal{B}_2$ *satisfying either*

$$\mathsf{Adv}^{DL}_{\mathcal{B}_1} \geq \frac{\mathsf{Adv}^{\text{EUF-IBS-CMA}}_{\mathcal{A}}(\eta)}{Q_G}\left(\frac{\mathsf{Adv}^{\text{EUF-IBS-CMA}}_{\mathcal{A}}(\eta)}{Q_G^2} - \frac{1}{2^\eta}\right)$$

or

$$\mathsf{Adv}^{DL}_{\mathcal{B}_2}(\eta) \geq \mathsf{Adv}^{\text{EUF-IBS-CMA}}_{\mathcal{A}}(\eta)\left(\frac{(\mathsf{Adv}^{\text{EUF-IBS-CMA}}_{\mathcal{A}}(\eta))^3}{(Q_G Q_H)^6} - \frac{3}{2^\eta}\right).$$

Proof. Assume there is an adversary $\mathcal{A}$ that wins the game EUF-IBS-CMA with non-negligible probability. Eventually, $\mathcal{A}$ outputs an attempted forgery of the form $\sigma = (A, B, R)$. Let E be the event that σ is a valid signature and R was contained in an answer of the signing oracle $\mathcal{O}_{\mathcal{S}}$. Let NE be the event that σ is a valid signature and R was never part of an answer of $\mathcal{O}_{\mathcal{S}}$. Clearly, $\mathsf{Adv}^{\text{EUF-IBS-CMA}}_{\mathcal{A}}(\eta) = \mathbb{P}[E] + \mathbb{P}[NE]$.

Next we build the following adversaries $\mathcal{B}_1, \mathcal{B}_2$ against the discrete logarithm assumption. Intuitively, $\mathcal{B}_1$ breaks the DL-assumption when the event E happens and $\mathcal{B}_2$ in case of NE.

$\mathcal{B}_1$ takes as argument the description of a group $(\mathcal{G}, q, g)$ and a challenge g^r with $r \xleftarrow{\$} \mathbb{Z}_q$ and tries to extract the discrete logarithm r. To do so it will run the adversary $\mathcal{A}$, for which simulates its environment as follows:

- $\mathcal{B}_1$ picks a random $i \leftarrow [Q_G]$, where Q_G is the maximal number of queries that the adversary $\mathcal{A}$ performs to the random oracle G. Let $\mathsf{id}^\star$ (the target identity) be the identity in the i-th query to the random oracle G. Next, $\mathcal{B}_1$ chooses $z \xleftarrow{\$} \mathbb{Z}_q$ and sets public parameters $\mathsf{mpk} = (\mathcal{G}, q, g, G, H, \mathsf{mpk} = g^z)$ where G, H are descriptions of hash functions modeled as random oracles. As usual, $\mathcal{B}_1$ simulates these oracles by keeping two lists L_G, L_H containing the queried values together with the answers given to $\mathcal{A}$.
- Every time $\mathcal{A}$ queries the key extraction oracle $\mathcal{O}_{\mathcal{E}}$, for user id, $\mathcal{B}_1$ chooses $c, y \xleftarrow{\$} \mathbb{Z}_q$, sets $R = g^{-zc}g^y$ and adds $((R, \mathsf{id}), c)$ to the list L_H. Then it returns the key (y, R) to $\mathcal{A}$.
- When $\mathcal{A}$ makes a call to the signature oracle (id, m) with $\mathsf{id} \neq \mathsf{id}^\star$, $\mathcal{B}_1$ simply computes id's private key as described in the previous bullet. Then it runs the signing algorithm $\mathcal{S}$ and returns the produced signature to $\mathcal{A}$.
- When $\mathcal{A}$ makes a call to the signature oracle (id, m) with $\mathsf{id} = \mathsf{id}^\star$, $\mathcal{B}_1$ chooses $t \xleftarrow{\$} \mathbb{Z}_q, B \xleftarrow{\$} \mathcal{G}$, sets $R = g^{-zc}(g^r)^t$, $c = H(\mathsf{id}, g^r)$ and $A = B(g^r g^{zc})^{-d}$. Then it returns the signature (A, B, R) to the adversary $\mathcal{A}$.

- $\mathcal{B}_1$ runs the algorithm $\mathsf{MF}_{Y,1}(\mathsf{mpk})$ as described in Lemma 1. Here algorithm Y is simply a wrapper that takes as explicit input the answers from the random oracles. Then it calls $\mathcal{A}$ and returns its output together with two integers I, J. This integers are the indexes of $\mathcal{A}$'s calls to the random oracles G, H with the target identity $\mathsf{id}^\star$.

algorithm $\mathsf{MF}_{Y,1}(\mathsf{mpk})$:
Pick random coins ρ for Y
$s_1 \dots s_{Q_G} \xleftarrow{\$} \mathbb{Z}_q$
$(I, J, \sigma_0) \leftarrow Y(\mathsf{mpk}, s_1 \dots s_{Q_G}; \rho)$
if $(I = 0 \vee J = 0)$ **then return** $\perp$
$s_I^1 \dots s_{Q_G}^1 \xleftarrow{\$} \mathbb{Z}_q$
$(I_1, J_1, \sigma_1) \leftarrow Y(\mathsf{mpk}, s_1, \dots s_{I-1}, s_I^1 \dots s_{Q_G}^1; \rho)$
if $((I, J) \neq (I_1, J_1) \vee s_I = s_I^1)$ **then return** $\perp$
else return σ_0, σ_1

  In this way we get two forgeries of the form $\sigma_0 = (\mathsf{id}, m, (A, B_1, R))$ and $\sigma_1 = (\mathsf{id}, m, (A, B_2, R))$. Let d_1 be the answer from the random oracle G given to $\mathcal{A}$ in the first execution, i.e., s_I in $\mathsf{MF}_{Y,1}(\mathsf{mpk})$, and let d_2 be the second answer s_I^1. If the identity id is not equal to the target identity $\mathsf{id}^\star$ then $\mathcal{B}_1$ aborts. Otherwise it terminates and outputs the attempted discrete logarithm $(B_1 - B_2)(td_1 - td_2)^{-1}$.

Next we show that $\mathcal{B}_1$'s output is indeed the discrete logarithm r of the challenge G^r. Since both signatures are valid we get that

$$g^{B_1} = A(Rg^{zc})^{d_1} \qquad \text{and} \qquad g^{B_2} = A(Rg^{zc})^{d_2}$$

where $c = H(g^r, \mathsf{id})$ and $R = g^{-zc}g^{rt}$. Hence, $B_1 = \log A + rtd_1$ and $B_2 = \log A + rtd_2$ and therefore $r = (B_1 - B_2)(td_1 - td_2)^{-1}$.

Next we need to lower bound success probability of $\mathcal{B}_1$ against the discrete logarithm assumption. To this aim we use the Multiple-Forking Lemma of Boldyreva, Palacio and Warinschi [BPW03], which we include for the sake of self-containment. The lemma is a further generalization of the General Forking Lemma proposed by Bellare and Neven [BN06] to multiple random oracles and signatures. The General Forking Lemma itself is a generalization of the Forking Lemma, originally proposed in [PS00]. Intuitively, this lemma states that, in the random oracle model, if there is an algorithm (a forger) that can produce a forgery, then it is possible to get a different forgery on the same message (by changing the answer of the random oracle).

Lemma 1 (Multiple-Forking Lemma [BPW03]). *Fix $\alpha \in \mathbb{Z}^+$ and a set S such that $|S| \geq 2$. Let Y be a randomized algorithm that on input x and a sequence of elements $s_1 \dots s_\alpha \in S$, returns a triple (I, J, σ) consisting of two*

algorithm $\mathsf{MF}_{Y,n}(x)$:
Pick random coins ρ for Y
$s_1 \dots s_\alpha \xleftarrow{\$} S$
$(I, J, \sigma_0) \leftarrow Y(x, s_1 \dots s_\alpha; \rho)$
if $(I = 0 \vee J = 0)$ **then return** $\perp$
$s_I^1 \dots s_\alpha^1 \xleftarrow{\$} S$
$(I_1, J_1, \sigma_1) \leftarrow Y(x, s_1, \dots s_{I-1}, s_I^1 \dots s_\alpha^1; \rho)$
if $((I, J) \neq (I_1, J_1) \vee s_I = s_I^1)$ **then return** $\perp$
for $i = 2 ; i < n ; i = i + 2$ **do**
  $s_1^i \dots s_\alpha^i \xleftarrow{\$} S$
  $(I_i, J_i, \sigma_i) \leftarrow Y(x, s_1 \dots s_{J-1}, s_J^i, \dots, s_\alpha^i; \rho)$
  if $((I_i, J_i) \neq (I, J) \vee s_J^i = s_J^{i-1})$ **then return** $\perp$
  $s_I^{i+1} \dots s_\alpha^{i+1} \xleftarrow{\$} S$
  $(I_{i+1}, J_{i+1}, \sigma_{i+1}) \leftarrow Y(x, s_1 \dots s_{J-1}, s_J^i, \dots, s_{I-1}^i, s_I^{i+1}, \dots, s_\alpha^{i+1}; \rho)$
  if $((I_{i+1}, J_{i+1}) \neq (I, J) \vee s_J^{i+1} = s_J^i)$ **then return** $\perp$
endfor
return $\sigma_0 \dots \sigma_n$

Fig. 1. Multiple-forking algorithm

integers associated to Y and a bitstring σ. Let $n \geq 1$ be an odd integer and x a bitstring. The multiple-forking algorithm $\mathsf{MF}_{Y,n}$ associated to Y and n is defined as in Figure 1. Let

$$\mathsf{acc} = \mathbb{P}[x \xleftarrow{\$} PG; s_1 \dots s_\alpha \xleftarrow{\$} S; (I, J, \sigma) \leftarrow Y(x, s_1 \dots s_\alpha) : I \geq 1 \wedge J \geq 1]$$
$$\mathsf{frk} = \mathbb{P}[x \xleftarrow{\$} PG : \mathsf{MF}_{Y,n}(x) \neq \perp].$$

Then

$$\mathsf{frk} \geq \mathsf{acc}\left(\frac{\mathsf{acc}^n}{\alpha^{2n}} - \frac{n}{|S|}\right)$$

and therefore

$$\mathsf{acc} \leq \sqrt[n+1]{\alpha^{2n}\,\mathsf{frk}} + \sqrt[n+1]{\frac{n\alpha^{2n}}{|S|}}$$

To bound the success probability of $\mathcal{B}_1$ against de discrete logarithm assumption first notice that with probability $1/Q_G$ the target identity $\mathsf{id}^\star$ equals the identity id output by the adversary. Then, it follows from Lemma 1 that

$$\mathsf{Adv}_{\mathcal{B}_1}^{DL} \geq \frac{\mathsf{frk}}{Q_G} \geq \frac{\mathsf{Adv}_{\mathcal{A}}^{\text{EUF-IBS-CMA}}(\eta)}{Q_G}\left(\frac{\mathsf{Adv}_{\mathcal{A}}^{\text{EUF-IBS-CMA}}(\eta)}{Q_G^2} - \frac{1}{2^\eta}\right)$$

It lacks to bound the success probability of *NE*. This case is slightly more involved as it uses nested rewindings of the adversary. In this case $\mathcal{B}_2$ attacks the public key of the trusted authority g^z. It takes as argument the description

of a group $(\mathcal{G}, q, g)$ and a challenge g^z with $z \xleftarrow{\$} \mathbb{Z}_q$ and outputs its discrete logarithm z. To do so it will run the adversary $\mathcal{A}$ simulating its environment as follows:

- At the beginning of the experiment, $\mathcal{B}_2$ sets public parameters $\mathsf{mpk} = (\mathcal{G}, q, g, G, H, \mathsf{mpk} = g^z)$ where G, H are description of hash functions modeled as random oracles and g^z is the challenge. As usual, $\mathcal{B}_2$ simulates these oracles by keeping two lists L_G, L_H containing the queried values together with the answers given to $\mathcal{A}$.
- Every time $\mathcal{A}$ queries the key extraction oracle $\mathcal{O}_{\mathcal{E}}$, for user id, $\mathcal{B}_2$ chooses $c, y \xleftarrow{\$} \mathbb{Z}_q$, sets $R = g^{-zc}g^y$ and adds $((R, \mathsf{id}), c)$ to the list L_H. Then it returns the key (y, R) to $\mathcal{A}$.
- When $\mathcal{A}$ makes a call to the signature oracle (id, m), $\mathcal{B}_2$ simply computes id's secret key as described in the previous step. Then computes a signature by calling $\mathcal{S}$, adding the respective call to the oracle G, $((\mathsf{id}, g^a, m), d)$ to the list L_G and gives the resulting signature to the adversary.
- $\mathcal{B}_2$ runs the algorithm $\mathsf{MF}_{\mathcal{A},3}(\mathsf{mpk})$ as described in Lemma 1. In this way, either $\mathcal{B}_2$ aborts prematurely or we get, for some identity id, some message m and some R, four forgeries $(\mathsf{id}, m, (A_k, B_k, R))$, $k = 1 \dots 4$ with $A_1 = A_2$ and $A_3 = A_4$. As all these signatures are valid, the following equations hold

$$B_1 = \log A_1 + (\log R + c_1 z)d_1 \qquad B_2 = \log A_2 + (\log R + c_1 z)d_2$$
$$B_3 = \log A_3 + (\log R + c_2 z)d_3 \qquad B_4 = \log A_4 + (\log R + c_2 z)d_4$$

  with $c_1 \neq c_2$, $d_1 \neq d_2$ and $d_3 \neq d_4$. Since we know $c_1, c_2, d_1, \dots, d_4$, a simple computation yields

$$z = \frac{B_3 + B_2 - B_1 - B_4}{c_2(d_3 - d_4) - c_1(d_1 - d_2)}\,.$$

It follows that the success probability of $\mathcal{B}_2$ is bounded by

$$\mathsf{Adv}^{DL}_{\mathcal{B}_2}(\eta) \geq \mathsf{frk} \geq \mathsf{Adv}^{\text{EUF-IBS-CMA}}_{\mathcal{A}}(\eta)\left(\frac{(\mathsf{Adv}^{\text{EUF-IBS-CMA}}_{\mathcal{A}}(\eta))^3}{(Q_G Q_H)^6} - \frac{3}{2^\eta}\right) \qquad \square$$

5 Comparison to Previous Schemes

This section compares the efficiency of our scheme with previous provably secure IBS schemes in terms of computational complexity and signature size. As it is common for cryptographic schemes with security reductions in the random oracle model, we ignore the tightness of the corresponding security reductions when computing the length of the parameters of the schemes. Given the considerable number of IBS schemes in the literature, we choose not to list all the existing schemes one by one. Instead we classify previous schemes in three categories, depending on whether those schemes are based on factoring, elliptic curve discrete

logarithm (ECDL) or pairing. Then we show that our scheme can be considered as the benchmark in efficiency in each of these categories. The reader interested in the state of the art of IBS schemes is referred to [BNN04].

We need to introduce some terminology. In what follows, an exponentiation refers to computing g^r for randomly taken $g \xleftarrow{\$} G$ and $r \xleftarrow{\$} \mathbb{Z}_t$, where G denotes a finite group, t is its order and r is an integer. A multi-exponentiation mexp(l) refers to computing $g_1^{r_1} \cdots g_l^{r_l}$, where $g_1, \ldots, g_l \xleftarrow{\$} G$ and $r_1, \ldots, r_l \xleftarrow{\$} \mathbb{Z}_t$. This operation can be computed more efficiently than just computing l single exponentiations due to an algorithm by Strauss [Str64], which is sometimes referred as Shamir's trick in the literature. Finally, we write $|G|$ to denote the number of bits needed to represent an element in G.

To start with the comparison, we need to state the efficiency properties of our scheme. For reasons of efficiency we consider that our scheme should be built upon a group of points $\mathcal{G}$ of prime order q of a suitable elliptic curve. A signature on our scheme consists of two elements in $\mathcal{G}$ and one element in $\mathbb{Z}_q$. To sign, one needs to compute one exponentiation (a small number of multiplications in $\mathbb{Z}_q$ can be ignored). To verify, one needs to compute one multi-exponentiation mexp(3). For the sake of concreteness, we fix $|\mathcal{G}| \approx |\mathbb{Z}_q| \approx 256$ bits for a security level equivalent to a 128-bit symmetric key for AES (cf. [ECR]). According to Brumley [Bru06], a multi-exponentiation mexp(3) has a cost of about 1.5 times that of a single exponentiation.

In the first category of our classification we find *factoring-based schemes*. As it is well-known, key sizes for factoring-based schemes are much larger than key sizes for ECDL-based schemes for an equivalent security level. This implies in particular that performing a group exponentiation in a factoring-based scheme, where $G = \mathbb{Z}_n^*$ for an RSA-modulus n, is more expensive than an exponentiation in an ECDL-based scheme. For instance, for the 128-bit security level, $|\mathbb{Z}_n| \approx$ 3072 bits. This already forces us to restrict our attention to the schemes in this category that present the shortest signature length, since otherwise they would be prohibitive in certain resource-constrained devices where communication is expensive, like Wireless Sensor Networks [GST07]. The shortest signature size for existing factoring-based IBS schemes is equivalent to representing two elements in $|\mathbb{Z}_n|$ (cf. [BNN04]), and thus signature size is still considerably bigger than in our scheme. They do present a higher computational cost in signing and verifying, since the most efficient algorithms require at least two exponentiations in $\mathbb{Z}_n$ for signing, and one mexp(2) in verifying. Computing one mexp(2) in $\mathbb{Z}_n$ is more costly than computing one mexp(3) in $\mathcal{G}$ due to the larger key size in factoring-based schemes.

In the second place, let us consider previous provably secure *ECDL-based schemes*. A comparison to previous schemes yields that our scheme is the most efficient in all three features, namely, in signature size, signing cost and verifying cost. Indeed, our scheme enjoys the same efficiency as the scheme named as Beth IBS scheme in [BNN04], which to our knowledge was the most efficient ECDL-based IBS scheme to date. In sharp contrast to our case, the security of the Beth IBS scheme is still unproven (cf. [BNN04]).

It remains to compare our scheme with existing *pairing-based schemes*. To this end, we need to recall some facts about pairings (see [BSS05] for a comprehensive introduction). Let $\mathcal{G}_1, \mathcal{G}_2$ and $\mathcal{G}_T$ be finite Abelian groups in which the discrete logarithm is believed to be hard. A *pairing* is a bilinear function $e : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$. Let $\mathcal{G}_1 = E(\mathbb{F}_p)$ denote an elliptic curve over the finite field $\mathbb{F}_p$. Let the order of $\mathcal{G}_1$ be divisible by a primer r such that r also divides $p^\alpha - 1$, where α is the order of p in $\mathbb{Z}_r^*$ and is called the MOV embedding degree. The modified Tate pairing $e(\cdot, \cdot)$, which is the bilinear map usually recommended, takes values in the subgroup $\mathcal{G}_T$ of $\mathbb{F}_{p^\alpha}^*$ of order r and is defined in two ways, depending on whether E is a supersingular or ordinary curve.

In the supersingular case $\mathcal{G}_1 = \mathcal{G}_2 = E(\mathbb{F}_p)$. For supersingular curves the best parameter choices are in characteristic 3, and in this case $\alpha = 6$ and $|\mathbb{F}_{p^\alpha}^*| \approx 3072$. As a consequence, $|\mathcal{G}_1| \geq 3072/6 = 512$, since in groups equipped with a pairing an algorithm solving the (finite-field) discrete logarithm in $\mathcal{G}_T$ can be used to compute the ECDL in $\mathcal{G}_1$.

Ordinary curves can also be used to implement pairings, but in this case $\mathcal{G}_2$ is set to be a subgroup of $E(\mathbb{F}_{p^\alpha})$. Several techniques have been presented [BKLS02, SB06, GPS06, DSD07] in order to improve efficiency and bandwidth. For instance, by using an appropriate map, certain points in $E(\mathbb{F}_{p^{12}})$ can be compressed to points in a sextic twist $E(\mathbb{F}_{p^2})$. Therefore, typical parameters would be $|\mathcal{G}_1| \geq 256$, $\alpha = 12$ and $|\mathcal{G}_2| \geq 2 \cdot 256 = 512$.

In the pairing-based IBS category, the shortest-length signature schemes for pairings over supersingular curves consist of two elements in $\mathcal{G}_1$. In the ordinary curves category, the shortest signatures consist of two elements in $\mathcal{G}_1$, as in [Her06], or alternatively one element in $\mathcal{G}_1$ and one element in $\mathbb{Z}_q$, where $q = |\mathcal{G}_1|$, as in the scheme [BLMQ05] by Barreto et al. In the supersingular case, a signature has length of at least 1024 bits. This is in favor of our scheme, since our signatures are 768 bits long.

A more detailed comparison is needed in the case of Herranz and Barreto *et al.* schemes when pairings are implemented using ordinary curves. Then in both schemes the signature size is smaller than ours in about 256 bits, which is in our disadvantage. However, our scheme significantly outperforms the above mentioned schemes in computational time. We discuss that below.

5.1 The Case of Herranz and Barreto *et al.* Schemes

In the scheme proposed by Herranz, signing requires 1 exponentiation in $\mathcal{G}_1$ plus 1 hash-to-group evaluation, which generally has a non-negligible computational cost [BF03]. In contrast, our scheme only needs 1 exponentiation in $\mathcal{G}_1$, and therefore our signing algorithm is marginally more efficient than Herranz's. As for verifying, Herranz's scheme needs to compute 1 exponentiation in $\mathcal{G}_1$, 1 hash-to-group operation and 2 pairings. The latter operation can be replaced by the product of two pairings, which by a trick by Granger and Smart [GS06], has a cost of about 1.46 times that of a single pairing.

Thus we need a figure on the computational cost of pairings. The most efficient implementation we are aware of is due to Devegili, Scott and Dahab [DSD07]. They report to compute a pairing for the 128-bit security level using Barreto-Naehrig curves in about $9.04 \cdot 10^7$ clock cycles with the Philips HiPerSmart$^{\text{TM}}$ smartcard. The latter is an instantiation of the MIPS32®-based SmartMIPS® architecture with various instruction set enhancements to facilitate the implementation of popular cryptographic algorithms.

In order to compare the cost of computing a pairing versus the cost of exponentiating in $\mathcal{G}_1$, we use a result by Großschådl, Szekely and Tillich [GST07]. They report that computing an exponentiation in the NIST curve P-256 (that allows for efficient elliptic curve computations) requires $4.25 \cdot 10^6$ clock cycles in the 133 MHz StrongARM processor, the latter being used in resource-constrained devices like wireless sensor networks. Therefore, we can estimate that currently computing a pairing can be as expensive as 21 exponentiations in $\mathcal{G}_1$ at the 128-bit security level. We conclude that Herranz's scheme verification algorithm requires a computational effort equivalent to 31 exponentiations in the NIST curve P-256.

Barreto *et al.* scheme needs to compute one exponentiation in $\mathcal{G}_1$ and one exponentiation in $\mathcal{G}_T$ for signing. For verification, it computes 1 exponentiation in $\mathcal{G}_2$, 1 exponentiation in $\mathcal{G}_T$ plus 1 pairing. Exponentiating in $\mathcal{G}_2$ and $\mathcal{G}_T$ requires a higher computational cost than computing an exponentiation in $\mathcal{G}_1$, the exact cost depending on how the arithmetic on those groups is implemented. We were not able to find explicit estimates for these particular operations, so we leave them un-quantified. In any case, while our scheme requires 1.5 exponentiations in $\mathcal{G}_1$ for signing, Barreto *et al.* requires a computational effort equivalent to 21 exponentiations in $\mathcal{G}_1$ (for the pairing) plus one exponentiation in $\mathcal{G}_2$ plus one exponentiation in $\mathcal{G}_T$.

Figure 2 summarizes the performance comparison between the Schnorr-like scheme and [BLMQ05, Her06]. $\exp_{\mathcal{G}_\star}$ indicates the cost of computing an exponentiation in $\mathcal{G}_\star$, and assumes that a pairing costs about 21 exponentiations in $\mathcal{G}_1$. As a result, our scheme outperforms previous schemes in the pairing-based category, except for signature size, where the benchmark is still [BLMQ05, Her06]. The new scheme drastically outperforms IBS pairing-based schemes in verification time.

Scheme	Signature size	Sign	Verification
Schnorr-like	768 bits	1 $\exp_{\mathcal{G}_1}$	1.5 $\exp_{\mathcal{G}_1}$
Barreto *et al.*	512 bits	1 $\exp_{\mathcal{G}_2}$ + 1 $\exp_{\mathcal{G}_T}$	21 $\exp_{\mathcal{G}_1}$ + 1 $\exp_{\mathcal{G}_2}$ + 1 $\exp_{\mathcal{G}_T}$
Herranz	512 bits	$>$ 1 $\exp_{\mathcal{G}_1}$	31 $\exp_{\mathcal{G}_1}$

Fig. 2. Efficiency comparison among [BLMQ05, Her06] and Schnorr-like schemes for the 128-bit security level. $\exp_{\mathcal{G}_\star}$ indicates the cost of computing an exponentiation in $\mathcal{G}_\star$. For the sake of comparison, $\exp_{\mathcal{G}_1} \leq \exp_{\mathcal{G}_2}$ and $\exp_{\mathcal{G}_1} \leq \exp_{\mathcal{G}_T}$

6 Conclusion

In this work we have studied in detail the identity-based signature scheme yield by the concatenation of two Schnorr signatures. We have proven it secure under the discrete logarithm assumption using the random oracle methodology. The Schnorr-like IBS scheme outperforms in computational cost and underlying security assumption every single previously proposed provably secure identity-based signature scheme. Moreover, our signatures have also smaller bit complexity than any other provably secure scheme in the literature, with the sole exception of [BLMQ05, Her06]. Last but not least, the new scheme avoids the heavy code machinery need by pairing-based schemes. The properties enjoyed by our new scheme make it specially suited for deployment in resource-constrained devices where savings in computation and communication are a premium, e.g. wireless sensor networks.

Acknowledgements

The authors wish to thank the anonymous reviewers for helpful comments.

References

[AO99] Abe, M., Okamoto, T.: Delegation chains secure up to constant length. In: Varadharajan, V., Mu, Y. (eds.) ICICS 1999. LNCS, vol. 1726, pp. 144–156. Springer, Heidelberg (1999)

[Bet88] Beth, T.: Efficient zero-knowledge identification scheme for smart cards. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 77–84. Springer, Heidelberg (1988)

[BF03] Boneh, D., Franklin, M.K.: Identity-Based encryption from the Weil pairing. SIAM Journal of Computing 32(3), 586–615 (2003); This is the full version of an extended abstract of the same title presented in: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–615. Springer, Heidelberg (2001)

[BFPW07] Boldyreva, A., Fischlin, M., Palacio, A., Warinschi, B.: A closer look at PKI: Security and efficiency. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 458–475. Springer, Heidelberg (2007)

[BKLS02] Barreto, P.S.L.M., Kim, H.Y., Lynn, B., Scott, M.: Efficient algorithms for pairing-based cryptosystems. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 354–368. Springer, Heidelberg (2002)

[BLMQ05] Barreto, P.S.L.M., Libert, B., McCullagh, N., Quisquater, J.-J.: Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 515–532. Springer, Heidelberg (2005)

[BN06] Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: Proceedings of the 13th ACM conference on Computer and communications security (CCS 2006), pp. 390–399. ACM, New York (2006)

[BNN04] Bellare, M., Namprempre, C., Neven, G.: Security proofs for identity-based identification and signature schemes. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 268–286. Springer, Heidelberg (2004); The full version appears in Cryptology ePrint Archive: Report 2004/252

[BPW03] Boldyreva, A., Palacio, A., Warinschi, B.: Secure proxy signature schemes for delegation of signing rights. Cryptology ePrint Archive, Report 2003/096 (2003), http://eprint.iacr.org/

[Bru06] Brumley, B.B.: Efficient three-term simultaneous elliptic scalar multiplication with applications. In: Fåk, V. (ed.) Proceedings of the 11th Nordic Workshop on Secure IT Systems—NordSec 2006, Linköping, Sweden, October 2006, pp. 105–116 (2006)

[BSNS05] Baek, J., Safavi-Naini, R., Susilo, W.: Certificateless public key encryption without pairing. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 134–148. Springer, Heidelberg (2005)

[BSS05] Blake, I.F., Seroussi, G., Smart, N.: Advances in Elliptic Curve Cryptography. London Mathematical Society Lecture Note Series, vol. 317. Cambridge University Press, Cambridge (2005)

[CC02] Cha, J.C., Cheon, J.H.: An identity-based signature from gap Diffie-Hellman groups. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 18–30. Springer, Heidelberg (2002)

[CJT04] Castelluccia, C., Jarecki, S., Tsudik, G.: Secret handshakes from CA-oblivious encryption. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 293–307. Springer, Heidelberg (2004)

[Dig08] DigiNotar. Diginotar internet trust services (2008), http://www.diginotar.com

[DSD07] Devegili, A.J., Scott, M., Dahab, R.: Implementing cryptographic pairings over barreto-naehrig curves. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 197–207. Springer, Heidelberg (2007)

[ECR] ECRYPT. Ecrypt yearly report on algorithms and key lengths (2006), http://www.ecrypt.eu.org/documents/D.SPA.21-1.1.pdf revision 1.1 (January 29, 2007)

[EG08] Espinosa-Garcia, J.: The new Spanish electronic identity card: DNI-e. In: Conference on Cryptology and Digital Content Security (2008), http://www.crm.cat/Cryptology/Slides/Espinosa.pdf

[FS87] Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)

[Gir91] Girault, M.: Self-certified public keys. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 490–497. Springer, Heidelberg (1991)

[GPS06] Granger, R., Page, D., Smart, N.P.: High security pairing-based cryptography revisited. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 480–494. Springer, Heidelberg (2006)

[GQ90] Guillou, L.C., Quisquater, J.-J.: A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 216–231. Springer, Heidelberg (1990)

[GS06] Granger, R., Smart, N.: On computing products of pairings. Cryptology ePrint Archive, Report 2006/172 (2006), http://eprint.iacr.org/

[GST07] Großschädl, J., Szekely, A., Tillich, S.: The energy cost of cryptographic key establishment in wireless sensor networks. In: ASIACCS 2007, pp. 380–382. ACM, New York (2007)

[Her06] Herranz, J.: Deterministic identity-based signatures for partial aggregation. Comput. J. 49(3), 322–330 (2006)

[Hes03] Hess, F.: Efficient identity based signature schemes based on pairings. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 310–324. Springer, Heidelberg (2003)

[oIA08] Spanish Ministry of Internal Affairs. Electronic identity card (2008) (in Spanish), `http://www.dnielectronico.es/`

[Oka93] Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 31–53. Springer, Heidelberg (1993)

[PH97] Petersen, H., Horster, P.: Self-certified keys – concepts and applications. In: Communications and Multimedia Security 1997, pp. 102–116 (1997)

[PS00] Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology 13(3), 361–396 (2000)

[SB06] Scott, M., Barreto, P.S.L.M.: Generating more mnt elliptic curves. Des. Codes Cryptography 38(2), 209–217 (2006)

[Sch91] Schnorr, C.-P.: Efficient signature generation by smart cards. Journal of Cryptology 4(3), 161–174 (1991)

[Sha85] Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)

[SOK00] Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems based on pairing. In: The 2000 Symposium on Cryptography and Information Security, Oiso, Japan (2000)

[Str64] Strauss: Addition chains of vectors. American Mathematical Monthly 71(7), 806–808 (1964)