Pairing-Free ID-Based Signatures as Secure as Discrete Logarithm in AGM

Jia-Chng Loh(✉), Fuchun Guo, and Willy Susilo

School of Computing and Information Technology, Institute of Cybersecurity and Cryptology, University of Wollongong, Wollongong, Australia
{jial,fuchun,wsusilo}@uow.edu.au

Abstract. Identity-based signatures (IBS) allow the signer's identity information to be used as the public key for signature verification, eliminating the need for managing certificates to establish ownership of the corresponding public key. The Schnorr-like IBS due to Galindo and Garcia is known as the most efficient IBS based on the discrete logarithm (DL) problem, without the need for computationally expensive pairing operations. This makes it a lightweight and efficient solution for signature generation and verification. Unfortunately, the security reduction of Schnorr-like IBS is not tight under the standard EUF-CMA in the ID-based setting. Recently, by using the algebraic group model (AGM), where adversary computation is algebraic, the EUF-CMA security of ordinary Schnorr signatures has been proven tightly secure under DL assumption with random oracles. However, one could not trivially apply the reduction of Schnorr signatures in AGM to achieve tight security for the Schnorr-like IBS scheme because of the inability to capture the chosen identity-and-message attacks. In this work, we show that, with the adoption of AGM, it is feasible to tighten the EUF-CMA security for IBS without pairing under DL assumption with random oracles. We resolve the chosen identity-and-message attacks by adopting the OR-proof technique to generate the user's private key containing the DL of either one of the two random group elements, leading to a new pairing-free IBS scheme. We provide a concrete security analysis for the scheme in AGM showing that by embedding the DL problem instance into one of the randomness, the algebraic adversary could only return a non-reducible forgery and representations with half of the success probability.

Keywords: Pairing-free · Identity-based signatures · Tight security reduction · Algebraic group model

1 Introduction

Digital signatures ensure data authenticity and integrity in digital communication. A signature is valid if the verification algorithm holds, which requires the

W. Susilo—Supported by the ARC Australian Laureate Fellowship FL230100033.
F. Guo—Supported by the ARC Future Fellowship FT220100046.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
T. Zhu and Y. Li (Eds.): ACISP 2024, LNCS 14895, pp. 347–367, 2024.
https://doi.org/10.1007/978-981-97-5025-2_18

signer's public key as input. The identity-based signatures (IBS), introduced by Shamir [39], replace the signer's public key with a user's identity (e.g., email and identity card number), eliminating the need for a Public Key Infrastructure (PKI) whose responsible to bind between the public key and the signer, thereby substantially reducing costs associated with key management and certificate issuance in practice.

In modern cryptography, constructing a provably secure signature scheme usually comes with a reduction algorithm in some pre-defined security models, which turns algorithm $\mathcal{A}$ breaking the scheme with probability ϵ_A into algorithm $\mathcal{B}$ breaks some hard problems with probability $\epsilon_B \geq \frac{\epsilon_A}{L}$, such that $L \geq 1$ is the loss factor, and if L is small constant, we say the reduction is tight. A tight reduction has a meaningful feature as one could choose the optimal parameter size for the scheme when deploying it into practice, hence the scheme preserves the efficiency as proposed. However, it is a challenging task to achieve tight security for the IBS scheme under its standard security model, namely the existential unforgeability against chosen identity-and-message attacks (EUF-CMA). In particular, such a tight reduction algorithm must have a constant success probability near one that can (i) respond to all adversary's queries and (ii) extract problem solutions based on any given forgery.

With the emergence of pairing-based cryptography, several IBS schemes [3,5,9,14,21,26,34,35] have been proposed over time, each aiming to strike a balance between security and efficiency. For example, Waters-IBS [35] and BBG-IBS [26] are proven in the standard model without relying the random oracles [6]. Pairing-based cryptography, although powerful, may not be desirable under certain applications, especially in resource-constrained environments where computational power is limited, e.g., Internet of Things devices [1,31] and wireless sensor networks [32,37,41]. The Schnorr-like IBS scheme due to Galindo and Garcia [19] at AfricaCrypt'09 has been known as the most efficient IBS scheme due to its simple construction, using well-known Schnorr signatures [38] based on the discrete logarithm (DL) problem, without costly pairing operations.

The EUF-CMA security of Schnorr-like IBS was first proven under the DL assumption [19]. Although its security was revised and improved by Chatterjee et al. [8], the security loss of the scheme is still loosely reduced to the DL problem because of the need for the reset lemma [36], and schemes proven with reset lemma are known to suffer from the tightness barrier [25].

Since the introduction of the algebraic group model (AGM) [15], which idealizes the adversary's computation to be algebraic, such that any returned group element from the algebraic adversary must be described along with the representation based on all received group elements, the EUF-CMA security of ordinary Schnorr signatures [16] has been proven tightly secure under DL assumption in AGM. Recently, Loh et al. [28] proposed an extended BLS signatures [15] in an ID-based setting, namely the BLS-IBS scheme. With the help of the AGM, its EUF-CMA security can be proven tightly reduced to the DL problem. In addition, they mentioned that the Schnorr-like IBS scheme may not be tightly EUF-CMA secure even with the help of the algebraic adversary in the AGM

because of the difficulty of designing a reduction algorithm that can capture all kinds of adversary queries and forgeries. While there is no further discussion of this, it is, therefore, an interesting challenge to investigate if there exists any potential way to achieve a tightly secure IBS scheme without pairing.

1.1 Our Contribution

In this work, with the adoption of the AGM, we show the possibility of improving security reductions for pairing-free IBS schemes under DL assumption and the EUF-CMA security model. We first study the main reduction issue encountered in Schnorr-like IBS, for which the simulator could not respond to all adversary's queries, i.e., the inability to simulate the user's private key and extract problem solutions based on any given forgery. We resolve the simulator aborts issue by using the OR-proof technique [10,20] to generate the user's private key. In particular, the user's private key contains two group elements and a Schnorr's signature on the user's identity that binds with one of the randomness. While computing signatures, the user proves the possession of the key belongs to one of the two randomnesses. This leads to a new pairing-free IBS scheme.

We then prove the EUF-CMA security of the proposed IBS scheme can be tightly reduced to the DL problem in the AGM. Given a DL problem instance, we design a reduction algorithm in which there exists a simulator that can randomly embed the problem instance into one of the two group elements of the user's secret key, hence the simulator implicitly manages to respond to all adversary's queries, i.e., signing queries and key extraction queries. In the forgery phase, the algebraic adversary has no advantage in revealing which one is the embedded randomness, hence with the returned forgery and representations, our simulator can successfully extract the DL solution with a success probability of $\frac{1}{2}$.

Table 1 summarizes EUF-CMA secure IBS schemes in cyclic groups. Some IBS schemes with pairing [26,35] offer security in the standard model (SM) but lack tight reductions under the DL assumption. While most pairing-free IBS schemes under DL assumption are proven in the random oracle model (ROM) with loose reductions [5,7,19], except for [17,18] proven under decisional problems. Achieving tight reductions under DL assumption involves leveraging the AGM.

1.2 EUF-CMA Security of Schnorr in AGM

We first take a look at how the EUF-CMA security of Schnorr signatures [38] is proven tightly secure in the algebraic group model (AGM) [15]. Let g be a group generator of a group $\mathbb{G}$ in the prime order of p, $H : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ be a cryptography hash function, and $Z = g^z$ be the user's public key for some $z \in \mathbb{Z}_p^*$ and $r \in \mathbb{Z}_p^*$ be some randomness. The Schnorr signature $\sigma_m = (R, y)$ on message m is computed as

$$R = g^r, \quad y = r + z \cdot H(m, R).$$

Table 1. Summary of EUF-CMA Secure IBS Schemes

	Pairing-free	Hardness Assumption	Tight Reduction	SM/ROM/ AGM
ChCh-IBS [9]	×	CDH	×	ROM
BBMQ-IBS [3]	×	q-SDH	×	ROM
Waters-IBS [35]	×	CDH	×	SM
BBG-IBS [26]	×	mCDH	×	SM
BNN-IBS [5], Beth-IBS [7], Schnorr-like IBS [19]	✓	DL	×	ROM
FH-IBS [17,18]	✓	DDH	✓	ROM
BLS-IBS [28]	×	DL	✓	ROM+AGM
Ours	✓	DL	✓	ROM+AGM

At EuroCrypt'20, Fuchsbauer et al. [16] showed that its EUF-CMA security can be proven tightly secure in AGM. The security proof in AGM requires the adversary to be algebraic, such that an adversary that outputs a group element $X \in \mathbb{G}$ also outputs a vector $\vec{c} = (c_0, ..., c_n)$ representing how X is the linear combination $X = g^{c_0} C_1^{c_1} \cdots C_n^{c_n}$ of all group elements $g, C_0, ..., C_n \in \mathbb{G}$ that the adversary has obtained.

Suppose the algebraic adversary, in the forgery phase of the EUF-CMA game, returns a forgery $\sigma_{m^*} = (R^*, y^*)$ on chosen message m^* (without query any signature) and a vector $\vec{u}$, such that $R^* = g^{u_0} Z^{u_1}$. While forgery σ_{m^*} is valid, i.e. $g^{y^*} = R^* \cdot Z^{H(m^*, R^*)}$ holds, we obtain

$$R^* = g^{u_0} Z^{u_1} = g^{y^*} (Z)^{-H(m^*, R^*)}.$$

Now, assume that the DL problem instance is embedded into the master public key, i.e. $Z = g^{\alpha}$ where α is unknown. By observing the above equation, one may solve for $\alpha = \frac{y^* - u_0}{u_1 + H(m^*, R^*)}$ where the algebraic adversary has negligible success probability to set $u_1 = -H(m^*, R^*)$. We defer the full proof to [16]. A question arises as to why it is not trivial to adopt the same proof technique for the EUF-CMA security of the Schnorr-like IBS scheme [19].

1.3 Challenge to Achieve Tight Reduction for Schnorr-Like IBS

We first recall the Schnorr-like IBS scheme by Galindo and Garcia [19], which is constructed using two concatenated Schnorr signatures [38], and we discuss the main fact of designing a tight reduction.

Let g be a group generator of a group $\mathbb{G}$ in the prime order of p and $H_1, H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ be two cryptography hash functions. In the setup phase of the

IBS scheme, the TTP selects $z \in \mathbb{Z}_p^*$ and the master public and secret key pair as $(mpk = Z = g^z, msk = z)$. To generate a user's private key d_{ID}, the TTP produces a Schnorr signature on the identity of the user by using the master secret key. Let $r \in \mathbb{Z}_p^*$ be some randomness, and the user's private key is defined as $d_{ID} = (R = g^r, y = r + z \cdot H_1(ID, R))$. Next, the user uses its private key $d_{ID} = (R, y)$ to compute a Schnorr-like signature $\sigma_{ID,m} = (R, A, s)$. Let $a \in \mathbb{Z}_p^*$ be some randomness,

$$A = g^a, \quad s = a + y \cdot H_2(ID, m, A).$$

Non-tight Reduction. One notable challenge to design a tight reduction algorithm for Schnorr-like IBS in the EUF-CMA security model under DL assumption is the ability to respond to all adversary's queries and extract problem solution α on any given forgery. In particular, it is hard to design a reduction that the simulator can first respond to signature query $\mathcal{O}_S(ID, m) \rightarrow \sigma_{ID,m} = (R, A, s)$ then user's private key query $\mathcal{O}_E(ID) \rightarrow d_{ID} = (R, y)$ with the same R.

Given a DL problem instance tuple (g, g^α) where $\alpha \in \mathbb{Z}_p^*$, the simulator may embed g^α into either master public key $mpk = Z$, user's private key randomness R, or signature randomness A.

Non-simulatable User's Private Key. Suppose Z, A are embedded with g^α. This yields a critical issue when the adversary queries $\mathcal{O}_S(ID', m)$, then $\mathcal{O}_E(ID')$. Upon receiving the extraction query, the simulator must abort as it cannot simulate $d_{ID'} = (R', y')$, such that R' that was used to compute $\sigma_{ID',m} = (R', A, s)$.

Simulatable User's Private Keys But Non-reducible. As noted in [28], the query phase of the EUF-CMA security game for IBS in AGM can be successful with overwhelming success probability if the simulator can simulate the user's private key $d_{ID'}$ and the corresponding signatures $\sigma_{ID',m}$. That is said, if Z, R' are embedded with g^α, the simulator can randomly choose $A = g^a$ for some $a \in \mathbb{Z}_p^*$ to simulate $\sigma_{ID',m} = (R', A, s)$ on (ID, m). Unfortunately, this is not enough to achieve tight reduction for the Schnorr-like IBS scheme. The following discussion explains the fact.

Suppose $ID' = ID^*$ is the forgery identity and $d_{ID^*} = (R^*, y^*)$ is simulatable. Note that the query phase is perfect if the simulator sets $Z = g^\alpha$ and $R^* = g^{r'^*} Z^{-h_{ID^*}}$, where $r'^*, h_{ID^*} \in \mathbb{Z}_p^*$ are randomly chosen, such that $h_{ID^*} = H_1(ID^*, R^*)$ is programmable due to random oracles. Therefore, upon receiving a queried signature $\sigma_{ID^*,m} = (R^*, A, s) \leftarrow \mathcal{O}_S(ID^*, m)$, we assume the algebraic adversary can return a valid forgery $\sigma_{ID^*,m^*} = (R^*, A^*, s^*)$ on (ID^*, m^*) and two vectors $\vec{u}, \vec{x}$ that representing how R^*, A^* are computed, respectively, such that $R^* = g^{u_0} Z^{u_1} R^{*u_2} A^{u_3}$ and $A^* = g^{x_0} Z^{x_1} R^{*x_2} A^{x_3}$. Since the forgery σ_{ID^*,m^*} is valid, the verification equation holds, such that

$$g^{s^*} = A^* (R^* Z^{h_{ID^*}})^{h_{ID^*,m^*}} \tag{1}$$

where $h_{ID^*} = H_1(ID^*, R^*)$ and $h_{ID^*,m^*} = H_2(ID^*, m^*, A^*)$.

Notably, the above forgery and representations can be non-reducible by simply reusing the obtained $R^* = g^{r'^*} Z^{-h_{ID^*}}$ from the query phase. Therefore, based on Eq. (1), it becomes

$$g^{s^*} = A^* (g^{r'^*})^{h_{ID^*, m^*}}$$

The algebraic adversary can easily compute A^* using representations $\vec{x}$ as the linear combination without any embedded elements, i.e. Z, R^*. Hence, the simulator does not have sufficient knowledge to extract the DL solution α, i.e. coefficients of α are zero.

1.4 Our Techniques

Based on the above observations, we identify that the main challenge is to enable the reduction (i) to answer all adversary's queries; and (ii) to ensure the forgery on (ID^*, m^*) is reducible even though the adversary has queried for signatures on ID^*, i.e. d_{ID^*} was simulated. We hence attempt to use an OR-proof technique to generate the user's private key in the Schnorr-like IBS scheme. The idea is similar to the Twin signatures in [29] that sign with two possible secret keys, but here we adopt this technique to enable simulating the user's private key. In particular, the TTP now selects two randomness $r_0, r_1 \in \mathbb{Z}_p^*$ and computes the user's private key $d_{ID} = (R_0, R_1, b, y)$ as follows. Let $b \in \{0, 1\}$ be a random bit,

$$y = r_b + z \cdot H_1(ID, R_0, R_1), \quad R_0 = g^{r_0}, \quad R_1 = g^{r_1}.$$

Next, the user computes signature $\sigma_{ID,m}$, which can be viewed as a proof of knowledge of y with either the DL of randomness R_0, R_1. The user first retrieves $b \in \{0, 1\}$, y, R_0, R_1, and select new randomness $a'_0, a'_1, c_{1-b} \in \mathbb{Z}_p^*$ to compute A_0, A_1, s_{1-b} as follows

$$\begin{cases} b = 0, & A_0 = g^{a'_0}, \quad A_1 = g^{a'_1} (R_1 \cdot Z^{h_{ID}})^{-c_{1-b}}, \quad s_1 = a'_1, \\ b = 1, & A_0 = g^{a'_0} (R_0 \cdot Z^{h_{ID}})^{-c_{1-b}}, \quad A_1 = g^{a'_1}, \quad s_0 = a'_0. \end{cases}$$

Lastly, the user computes c_b, s_b such that

$$c_b = H_2(ID, m, A_0, A_1) - c_{1-b}, \quad s_b = a'_b + y \cdot c_b.$$

The resulted IBS is returned as $\sigma_{ID,m} = ((R_0,\ A_0,\ s_0,\ c_0), (R_1,\ A_1,\ s_1,\ c_1))$.

1.5 Overview of the Security Proof

Now we elaborate on how the security proof works at a high level. Given a DL problem instance tuple (g, g^α), we construct a simulation that embeds g^α to master public key Z and, based on random bit $b \in \{0, 1\}$, to either randomness (R_0, A_1) or (R_1, A_0). The user's private key and corresponding signature can be simulated as follows.

Simulating User's Private Keys. Let L_E be a list of users' private keys. Upon receiving the ID_i query, the simulator checks if d_{ID_i} exits in L_E and returns as it. Otherwise, the simulator embeds g^α into either one of the randomness $R_{i,0}, R_{i,1}$, which are programmed as follows. It first randomly selects $r'_{i,0}, r'_{i,1}, h_{ID_i} \in \mathbb{Z}_p$. Then, based on a chosen random bit $b_i \in \{0,1\}$, it sets $y_i = r'_{i,b_i}$,

$$\begin{cases} b=0, & R_{i,0} = g^{r'_{i,0}} \cdot (g^\alpha)^{-h_{ID_i}},\ R_{i,1} = g^{r'_{i,1}}, \\ b=1, & R_{i,0} = g^{r'_{i,0}},\ R_{i,1} = g^{r'_{i,1}} \cdot (g^\alpha)^{-h_{ID_i}}, \end{cases}$$

and hash oracle is programmed as $H_1(ID_i, R_{i,0}, R_{i,1}) = h_{ID_i}$. The simulator returns user's private key $d_{ID_i} = (R_{i,0}, R_{i,1}, b_i, y_i)$ and stores $\{ID_i, d_{ID_i}\}$ to the list of users' private keys L_E.

Simulating Signatures. The simulator can simulate the signature σ_{ID_i,m_i} on any user identity and message pair (ID_i, m_i) as any user's private key d_{ID_i} is simulatable – including d_{ID^*} for which ID^* is the chosen identity in the forgery phase. In particular, this resolves the issue that the simulator aborts while the adversary first asks for signatures σ_{ID_i,m_i} on ID_i then its corresponding user's private key d_{ID_i} because the inability to simulate d_{ID_i} with the same randomness generated in the signature query.

Our simulation allows such a query and additionally, the user's private key d_{ID^*} on the chosen identity ID^*, that the adversary is going to forge, is also simulatable. We hence need to enable the corresponding forgery σ_{ID^*,m^*} on (ID^*, m^*) is reducible.

Suppose d_{ID_i} does not exist in L_E. The simulator calls $\mathcal{O}_E(ID_i)$ and retrieves $\{ID_i, d_{ID_i}\}$ from L_E. Let $h_{ID_i} = H_1(ID_i, R_{i,0}, R_{i,1})$. If $b_i = 0$, the simulator obtains $R_{i,0} = g^{r'_{i,0} - \alpha h_{ID_i}}, R_{i,1} = g^{r'_{i,1}}$. It randomly selects $a'_{i,0}, a'_{i,1}, c_{i,1-b} \in \mathbb{Z}_p^*$, and based on $b_i \in \{0,1\}$, it sets

$$\begin{cases} b_i = 0, & A_{i,0} = g^{a'_{i,0}}, \quad A_{i,1} = g^{a'_{i,1}} (R_{i,1} \cdot Z^{h_{ID_i}})^{-c_{i,1-b}}, \\ & s_{i,1} = a'_{i,1}, \\ b_i = 1, & A_{i,0} = g^{a'_{i,0}} (R_{i,0} \cdot Z^{h_{ID_i}})^{-c_{i,1-b}}, \quad A_{i,1} = g^{a'_{i,1}}, \\ & s_{i,0} = a'_{i,0}. \end{cases}$$

It then calls for hash oracle and obtains $h_{ID_i,m_i} = H_2(ID_i, m_i, A_{i,0}, A_{i,1})$ and sets $c_{b_i} = h_{ID_i,m_i} - c_{i,1-b_i}$ and $s_{b_i} = a'_{b_i} + y_i \cdot c_{b_i}$. It is worth noting that in either case, the signature is simulatable and indistinguishable. The signature is returned as $\sigma_{ID_i,m_i} = ((R_{i,0}, A_{i,0}, s_{i,0}, c_{i,0}),\ (R_{i,1}, A_{i,1}, s_{i,1}, c_{i,1}))$, and the following verification algorithms hold

$$\begin{aligned} g^{s_0} &= A_0 (R_0 \cdot Z^{h_{ID}})^{c_0}, \text{ and} \\ g^{s_1} &= A_1 (R_1 \cdot Z^{h_{ID}})^{c_1}, \text{ and} \\ h_{ID,m} &= c_0 + c_1. \end{aligned}$$

Forgery. During the forgery phase, the algebraic adversary returns a forged signature $\sigma_{ID^*,m^*} = ((R_0^*, A_0^*, s_0^*, c_0^*), (R_1^*, A_1^*, s_1^*, c_1^*))$ on (ID^*, m^*) that passes the

Let $Z = g^{\alpha}$ be the master public key.
Given a forgery $\sigma_{ID^*,m^*} = \left((R_0^*, A_0^*, s_0^*, c_0^*), (R_1^*, A_1^*, s_1^*, c_1^*)\right)$ on (ID^*, m^*) and representations $\pi_{u,0}, \pi_{u,1}, \pi_{v,0}, \pi_{v,1}, \pi_{x,0}, \pi_{x,1}, \pi_{y,0}, \pi_{y,1} \in \mathbb{Z}_p$, such that

$$R_0^* = g^{\alpha\pi_{u,1}+\pi_{u,0}}, \quad R_1^* = g^{\alpha\pi_{v,1}+\pi_{v,0}},$$
$$A_0^* = g^{\alpha\pi_{x,1}+\pi_{x,0}}, \quad A_1^* = g^{\alpha\pi_{y,1}+\pi_{y,0}}.$$

The validity of the forgery holds, such that

$$g^{s_0^*} = A_0^*\left(R_0^* Z^{h_{ID^*}}\right)^{c_0^*}, \quad g^{s_1^*} = A_1^*\left(R_1^* Z^{h_{ID^*}}\right)^{c_1^*},$$
$$c_0^* + c_1^* = H_2(ID^*, m^*, A_0^*, A_1^*).$$

Hence, the simulator obtains following two modular equations:

$$\boxed{s_0^* = \alpha\pi_{x,1} + \pi_{x,0} + c_0^*\left(\alpha\pi_{u,1} + \pi_{u,0} + \alpha h_{ID^*}\right)}, \text{ and}$$
$$\boxed{s_1^* = \alpha\pi_{y,1} + \pi_{y,0} + c_1^*\left(\alpha\pi_{y,1} + \pi_{y,0} + \alpha h_{ID^*}\right)}$$

The simulator solves for

$$\alpha = \frac{s_0^* - \left(\pi_{x,0} + c_0^*\pi_{u,0}\right)}{\pi_{x,1} + c_0^*(\pi_{u,1} + h_{ID^*})}, \text{ or}$$
$$\alpha = \frac{s_1^* - \left(\pi_{y,0} + c_1^*\pi_{v,0}\right)}{\pi_{y,1} + c_1^*(\pi_{v,1} + h_{ID^*})}.$$

The reduction fails if the following **abort condition** holds, i.e. coefficients of α are zero:

$$\pi_{x,1} + c_0^*\left(\pi_{u,1} + h_{ID^*}\right) = 0, \text{and } \pi_{y,1} + c_1^*\left(\pi_{v,1} + h_{ID^*}\right) = 0.$$

Since coefficients can be freely crafted by the adversary, we, therefore, analyze the **abort condition** under the following three cases.

Case 1: $\pi_{u,1} + h_{ID^*} = \pi_{v,1} + h_{ID^*} = 0$

It implies that $\pi_{x,1} = \pi_{y,1} = 0$ is set.
While $h_{ID^*} = H_1(ID^*, R_0^*, R_1^*)$ is randomly chosen after $R_0^* = g^{\alpha\pi_{u,1}+\pi_{u,0}}$ and $R_1^* = g^{\alpha\pi_{v,1}+\pi_{v,0}}$ were queried to hash oracles, the adversary has negligible success probability to set both $\pi_{u,1} = \pi_{v,1} = -h_{ID^*}$, such that

$$\Pr\left[\pi_{u,1} = \pi_{v,1} = -h_{ID^*}\right] = \frac{1}{p}.$$

Case 2: $\pi_{u,1} + h_{ID^*} \neq 0$ and $\pi_{v,1} + h_{ID^*} \neq 0$

Since $c_0^* + c_1^* = h_{ID^*,m^*} = H_2(ID^*, m^*, A_0^*, A_1^*)$ holds. The **abort condition** holds if the following behaviors are set:

$$\mathcal{F}_0\text{: } \pi_{x,1} = -\left(h_{ID^*,m^*} - c_1^*\right)\left(\pi_{u,1} + h_{ID^*}\right), \text{ and}$$
$$\mathcal{F}_1\text{: } \pi_{y,1} = -\left(h_{ID^*,m^*} - c_0^*\right)\left(\pi_{v,1} + h_{ID^*}\right).$$

While $h_{ID^*,m^*} = H_2(ID^*, m^*, A_0^*, A_1^*)$ is randomly chosen after $A_0^* = g^{\alpha\pi_{x,1}+\pi_{x,0}}$ and $A_1^* = g^{\alpha\pi_{y,1}+\pi_{y,0}}$ were queried to hash oracles, the adversary has negligible success probability to launch $\mathcal{F}_0$ and $\mathcal{F}_1$, such that

$$\Pr[\mathcal{F}_1 \wedge \mathcal{F}_2] = \frac{1}{p}.$$

Case 3: $\pi_{u,1} + h_{ID^*} = 0$ or $\pi_{v,1} + h_{ID^*} = 0$.

Based on simulation definition, the adversary may query for signatures by calling $\mathcal{O}_S(ID^*, \cdot)$.
This means that the simulator simulated the corresponding user's private key $d_{ID^*} = (R_0', R_1', b^*, y^*)$, which implies that

$$\begin{cases} b^* = 0, & R_0^* = g^{\alpha(-h_{ID^*})+r_0'^*}, R_1^* = g^{r_1'^*}, \text{ such that } \pi_{u,1} = -h_{ID^*} \text{ and } \pi_{v,1} = 0. \\ & \pi_{x,1} + c_0^*(-h_{ID^*} + h_{ID^*}) = 0 \text{ and } \pi_{y,1} + c_1^*(0 + h_{ID^*}) = 0. \\ b^* = 1, & R_0^* = g^{r_0'^*}, R_1^* = g^{\alpha(-h_{ID^*})+r_1'^*}, \text{ such that } \pi_{u,1} = 0 \text{ and } \pi_{v,1} = -h_{ID^*}. \\ & \pi_{x,1} + c_0^*(0 + h_{ID^*}) = 0 \text{ and } \pi_{y,1} + c_1^*(-h_{ID^*} + h_{ID^*}) = 0. \end{cases}$$

The **abort condition** holds if either of the two behaviors is set:

$$\begin{cases} b^* = 0, & \mathcal{G}_0\text{: } \pi_{x,1} = 0, \quad \pi_{y,1} = -c_1^* h_{ID^*}; \\ b^* = 1, & \mathcal{G}_1\text{: } \pi_{x,1} = -c_0^* h_{ID^*}, \quad \pi_{y,1} = 0. \end{cases}$$

While $b^* \in \{0,1\}$ is perfectly hidden from the view of the adversary, it is depending on random guessing $b' \in \{0,1\}$ to launch $\mathcal{G}_{b'}$, hence

$$\Pr[b^* = b'] = \frac{1}{2}.$$

Fig. 1. An Overview of Successful Reduction

verification algorithms. While the adversary is algebraic, the simulator obtains representations $\vec{u}, \vec{v}, \vec{x}, \vec{y}$ that describe how $R_0^*, R_1^*, A_0^*, A_1^*$ are linearly combined, respectively. The overview of the successful reduction can be found in Fig. 1.

Suppose $Z = g^{\alpha}$, $R_0^* = g^{\alpha\pi_{u,1}+\pi_{u,0}}, R_1^* = g^{\alpha\pi_{v,1}+\pi_{v,0}}, A_0^* = g^{\alpha\pi_{x,1}+\pi_{x,0}}$, and $A_1^* = g^{\alpha\pi_{y,1}+\pi_{y,0}}$, where $\pi_{u,0}, \pi_{u,1}, \pi_{v,0}, \pi_{v,1}, \pi_{x,0}, \pi_{x,1}, \pi_{y,0}, \pi_{y,1}$ are computable coefficients according to the simulator's randomness and algebraic adversary's representations. Based on all the above definitions and verification algorithms, the simulator can derive the following modular equations

$$\begin{aligned} s_0^* &= \alpha\pi_{x,1} + \pi_{x,0} + c_0^*(\alpha\pi_{u,1} + \pi_{u,0} + \alpha h_{ID^*}), \\ s_1^* &= \alpha\pi_{y,1} + \pi_{y,0} + c_1^*(\alpha\pi_{v,1} + \pi_{v,0} + \alpha h_{ID^*}), \\ h_{ID^*,m^*} &= c_0^* + c_1^*, \end{aligned}$$

such that $h_{ID^*} = H_1(ID^*, R_0^*, R_1^*)$ and $h_{ID^*,m^*} = H_2(ID^*, m^*, A_0^*, A_1^*)$ were made. This allows the simulator to solve for DL solution α as long as its coefficients are non-zero.

It now remains to analyze the overall successful reduction. The reduction fails if the simulator aborts when the following abort condition holds, i.e. all coefficients of α are zero, such that $\pi_{x,1} + c_0^*(\pi_{u,1} + h_{ID^*}) = \pi_{y,1} + c_1^*(\pi_{v,1} + h_{ID^*}) = 0$. While $\pi_{u,1}, \pi_{v,1}, \pi_{x,1}, \pi_{y,1} \in \mathbb{Z}_p$ can be crafted by the algebraic adversary, we classify them into three potential cases. In particular, $\pi_{u,1} + h_{ID^*} = \pi_{v,1} + h_{ID^*} = 0$ in **Case 1**; and $\pi_{u,1} + h_{ID^*} \neq 0$ and $\pi_{v,1} + h_{ID^*} \neq 0$ in **Case 2**; and either $\pi_{u,1} + h_{ID^*} = 0$ or $\pi_{v,1} + h_{ID^*} = 0$ in **Case 3**. Fortunately, the overall probability of forgery and representations are non-reducible is $\frac{1}{2}$ because the algebraic adversary has no advantage to reveal the indistinguishable randomness, but a random guessing over a random bit $b^* \in \{0, 1\}$. We defer this analysis to the full proof in Sect. 3.2.

1.6 Related Work

The first EUF-CMA secure IBS scheme under DL assumption, known as Beth-IBS, was derived from Bellare et al.'s framework [5] transformed Beth's identification scheme [7] to IBS. However, due to the lack of security analysis for the Beth-IBS scheme, Galindo and Garcia [19] proposed a new IBS scheme based on the well-known Schnorr signature [38] in the random oracle model, namely Schnorr-like IBS scheme. The Schnorr-like IBS scheme is considered the most efficient IBS to date, due to its pairing-free setting. Although its security was improved by Chatterjee et al. [8], it is still loosely reduced to the DL problem due to the need for the reset lemma [36], which has been proven that the resulting security loss must suffer from the tightness barrier – the reduction loss cannot be tight [25].

Recently, Fukumitsu and Hasegawa [17,18] proposed enhancements to the Galindo and Garcia's Schnorr-like IBS [19] to design pairing-free IBS schemes with a tight security reduction. However, these two schemes are only proven under a strong assumption, i.e., the decisional Diffie-Hellman assumption. While in the weak-EUF-CMA security model [42], where the adversary is restricted

from asking for a user's private key $\mathcal{O}_E(ID')$ if the identity has already been requested for signatures $\mathcal{O}_S(ID', \cdot)$, making the reduction easier but non-standard as the challenge to respond to any adversary's query is omitted.

Other Signature Schemes in AGM. The AGM has been widely adopted in variants of signature schemes to achieve better efficiency and enhanced properties. For instance, multi-signature schemes [4,11,24,27,30] allow multiple signers to collaborate on signing a single message. Blind signature schemes [16,22,23,40] enable a signer to create a valid signature without revealing the message's content. Threshold signature schemes [2,11,12] necessitate at least 't-out-of-n' signers to jointly produce a valid signature. Notably, a threshold blind signature scheme was recently proposed in [13].

To the extent of our current understanding, in the AGM setting that focuses on achieving tight reductions under the DL assumption and standard EUF-CMA security, only Schnorr signatures [16], BLS signatures [15], and its identity-based adaptation [28] have been investigated.

2 Preliminaries

2.1 Definition of Identity-Based Signatures (IBS)

An identity-based signature (IBS) scheme consists of the following algorithms:

- $Setup(1^k)$: On input security parameters 1^k, it returns the master public and secret key pair (mpk, msk).
- $Extract(mpk, msk, ID)$: On input master public and secret key pair (mpk, msk) and a user identity ID, it returns a user private key d_{ID}.
- $Sign(mpk, d_{ID}, m)$: On input (mpk, d_{ID}) and message m, it returns a signature $\sigma_{ID,m}$.
- $Verify(mpk, ID, m, \sigma_{ID,m})$: On input $(mpk, ID, m, \sigma_{ID,m})$, it returns 1 or 0 which indicates accept or reject respectively.

Correctness. For user identity ID and user private key pair (ID, d_{ID}) that is extracted based on the master public and secret key pair (mpk, msk), signing a message m with d_{ID} must return a correct signature $\sigma_{ID,m}$ that is valid on user ID, i.e. $Verify(mpk, ID, m, \sigma_{ID,m}) = 1$.

2.2 The EUF-CMA Security for IBS

The notion of existential unforgeability against chosen identity-and-message attacks (EUF-CMA) [5] security model is defined between a challenger and an adversary as follows.

- **Setup Phase:** Let L_E, L_S be two sets of extract queries and signing queries, respectively. The challenger runs *Setup* algorithm to compute a master public and secret key pair (mpk, msk). The challenger forwards mpk to the adversary.

- **Query Phase:** The adversary may adaptively ask for user private key d_{ID_i} on any chosen identity ID_i to the extraction oracle $\mathcal{O}_E(ID_i)$ and signatures σ_{ID_i,m_i} on any chosen identity-and-message pair (ID_i, m_i) to the signing oracle $\mathcal{O}_S(ID_i, m_i)$.
  - Extraction oracle $\mathcal{O}_E(ID_i)$: On input i-th query for ID_i, it first checks whether $\langle ID_i, \cdot \rangle \in L_E$. The challenger returns d_{ID_i} if it finds $\langle ID_i, d_{ID_i} \rangle \in L_E$. Otherwise, the challenger calls $Extract(mpk, msk, ID_i) \rightarrow d_{ID_i}$ algorithm and stores $\langle ID_i, d_{ID_i} \rangle$ to L_E. The challenger returns d_{ID_i}.
  - Signing oracle $\mathcal{O}_S(ID_i, m_i)$: On input i-th query for (ID_i, m_i), it first checks and retrieves d_{ID_i} if $\langle ID_i, d_{ID_i} \rangle \in L_E$ exists. If $\langle ID_i, d_{ID_i} \rangle \notin L_E$, the challenger calls $Extract(mpk, msk, ID_i) \rightarrow d_{ID_i}$ algorithm and stores (ID_i, d_{ID_i}) to L_E. The challenger calls $Sign(mpk, d_{ID_i}, m_i) \rightarrow \sigma_{ID_i,m_i}$ and stores $\langle ID_i, m_i, \sigma_{ID_i,m_i} \rangle$ to L_S. The challenger returns σ_{ID_i,m_i} that is valid and verifiable by running $Verify(mpk, ID_i, m_i, \sigma_{ID_i,m_i}) = 1$.
- **Forgery Phase:** The adversary returns challenge identity, message and signature pair $(ID^*, m^*, \sigma_{ID^*,m^*})$. The adversary wins if the verification holds, such that $Verify(mpk, ID^*, m^*, \sigma_{ID^*,m^*}) = 1$, where ID^* has not been queried to the extraction oracle $\mathcal{O}_E(ID^*)$ and (ID^*, m^*) has not been queried to the signing oracle $\mathcal{O}_S(ID^*, m^*)$.

Definition 1. *An IBS scheme is (ϵ, q_e, q_s, t)-secure in the EUF-CMA security model if it is infeasible that any probabilistic polynomial-time adversary who runs in t polynomial time and makes at most q_e extraction queries and q_s signing queries has advantage at most ϵ in winning the game, where ϵ is negligible function of the input security parameter.*

2.3 Discrete Logarithm Problem

Let $\mathbb{G}$ be a group of prime order p with generator $g \in \mathbb{G}$. The discrete logarithm (DL) problem is to compute $\alpha \in \mathbb{Z}_p$, given a random group element $g^\alpha \in \mathbb{G}$.

Definition 2. *The (ϵ, t)-DL assumption holds in $\mathbb{G}$ if there is no probabilistic polynomial-time adversary who runs in t polynomial time has the advantage at most ϵ to solve the DL problem in $\mathbb{G}$.*

2.4 The Algebraic Adversary

The algebraic adversary was studied in [33], and was formalized in the AGM [15]. The security reduction in the AGM is like the standard model, but we say the computation of the adversary is algebraic. Informally, suppose algebraic adversary is given $(g, C_1, ..., C_n) \in \mathbb{G}^{n+1}$ group elements. For any group element it outputs $X \in \mathbb{G}$, it is also required to return a representation vector $\vec{c} = (c_0, c_1, ..., c_n) \in \mathbb{Z}_p^{n+1}$, that indicates how a returned group element X been generated based on the received group elements, such that

$$X = g^{c_0} \cdot C_1^{c_1} \cdots C_n^{c_n} = g^{c_0} \prod_{i=1}^{n} C_i^{c_i}.$$

3 The Proposed Pairing-Free Identity-Based Signatures

3.1 Scheme

– `Setup`: On input security parameter 1^k, it selects $g \in \mathbb{G}$, where g is the generator of group $\mathbb{G}$ in the prime order of p, and $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p, H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p$ be two cryptography hash functions. It then selects $z \in \mathbb{Z}_p$ and computes $Z = g^z$. The master public and secret key pair (mpk, msk) is returned to a trusted third party (TTP) as follows, such that

$$mpk = (g, p, \mathbb{G}, H_1, H_2, Z), \quad msk = z.$$

– `Extract`: On input (mpk, msk) and user identity $ID \in \{0,1\}^*$. It selects $r_0, r_1 \in \mathbb{Z}_p^*$ and sets $R_0 = g^{r_0}, R_1 = g^{r_1}$. Let $h_{ID} = H_1(ID, R_0, R_1)$, it then selects a random bit $b \in \{0,1\}$ and returns user private key $d_{ID} = (R_0, R_1, b, y)$, such that

$$y = r_b + z \cdot h_{ID}.$$

In practice, TTP would securely distribute d_{ID} to the user. Hence one may assume that TTP stores a list of user private keys $L_E = \langle ID, d_{ID} \rangle$, which reflects the security proof that returns the same d_{ID} upon receiving extraction query on ID.

– `Sign`: On input (mpk, d_{ID}, m). Let $h_{ID} = H_1(ID, R_0, R_1)$. It retrieves $b \in \{0,1\}$ and randomly selects $a'_0, a'_1, c_{1-b} \in \mathbb{Z}_p^*$. It then computes A_0, A_1, s_{1-b}, such that

$$\begin{aligned} A_0 &= g^{a'_0}(R_0 \cdot Z^{h_{ID}})^{(-c_{1-b}) \cdot b}, \\ A_1 &= g^{a'_1}(R_1 \cdot Z^{h_{ID}})^{(-c_{1-b}) \cdot (1-b)}, \\ s_{1-b} &= a'_{1-b}. \end{aligned}$$

Let $h_{ID,m} = H_2(ID, m, A_0, A_1)$, it then computes c_b and s_b, such that

$$c_b = h_{ID,m} - c_{1-b}, \quad s_b = a'_b + y \cdot c_b.$$

The signature $\sigma_{ID,m}$ is returned as follows, where

$$\sigma_{ID,m} = ((R_0,\ A_0,\ s_0,\ c_0), (R_1,\ A_1,\ s_1,\ c_1)).$$

– `Verify`: On input $(mpk, ID, m, \sigma_{ID,m})$. Let $h_{ID} = H_1(ID, R_0, R_1)$ and $h_{ID,m} = H_2(ID, m, A_0, A_1)$, it checks the following equations, such that

$$g^{s_0} = A_0(R_0 \cdot Z^{h_{ID}})^{c_0}, \tag{2}$$

$$g^{s_1} = A_1(R_1 \cdot Z^{h_{ID}})^{c_1}, \tag{3}$$

$$h_{ID,m} = c_0 + c_1. \tag{4}$$

It returns 1 if Eqs. (2), (3), and (4) hold. Otherwise, it returns 0.

Correctness. Recall that $Z = g^z$ is from mpk. The scheme is correct if the signature $\sigma_{ID,m} = ((R_0,\ A_0,\ s_0,\ c_0), (R_1,\ A_1,\ s_1,\ c_1))$ on message m generated by the user ID with user private key $d_{ID} = (R_0, R_1, b, y)$ holds. Let $h_{ID} = H_1(ID, R_0, R_1)$ and $h_{ID,m} = H_2(ID, m, A_0, A_1)$. Suppose $b = 0$, we have

$$\begin{aligned} R_0 &= g^{r_0}, & R_1 &= g^{r_1}, \\ A_0 &= g^{a_0}, & A_1 &= g^{a_1}(R_1 \cdot Z^{h_{ID}})^{-c_1}, \\ s_0 &= a_0 + y \cdot c_0, & s_1 &= a_1. \\ y &= r_0 + z \cdot h_{ID}, & c_1 &= h_{ID,m} - c_0. \end{aligned}$$

Otherwise, if $b = 1$, we have

$$\begin{aligned} R_0 &= g^{r_0}, & R_1 &= g^{r_1}, \\ A_0 &= g^{a_0}(R_0 \cdot Z^{h_{ID}})^{-c_0}, & A_1 &= g^{a_1}, \\ s_0 &= a_0, & s_1 &= a_1 + y \cdot c_1. \\ y &= r_1 + z \cdot h_{ID}, & c_0 &= h_{ID,m} - c_1. \end{aligned}$$

It is easy to verify Eqs. (2), (3), and (4) hold in either $b \in \{0, 1\}$.

3.2 Security Analysis

Theorem 1. *If an adversary can break the EUF-CMA of the proposed IBS scheme in the AGM, then there exists an adversary that can solve the DL problem with a success probability of* $\frac{1}{2}$.

Proof. Suppose there exists an algebraic adversary who can (t, q_e, q_s, ϵ)-break the proposed IBS scheme under EUF-CMA in the AGM. Given as input a DL problem instance (g, g^α) over the cyclic group tuple $(p, g, \mathbb{G})$. We design a simulator that can solve the DL problem with the forgery and representations given by the algebraic adversary.

- **Setup.** During the setup phase, the simulator embeds the DL problem instance to $Z = g^\alpha$ and returns the master public mpk, such that

$$mpk = (p, g, \mathbb{G}, Z).$$

- **Hash Oracles.** At the beginning, the simulator prepares two empty sets L_{H1}, L_{H2} to record all hash queries and responses as follows.
  $\mathcal{H}_1(ID_i, R_{i,0}, R_{i,1})$: On an i-th random oracle query, i.e. on input $(ID_i, R_{i,0}, R_{i,1})$. If the tuple exists, such that $\langle ID_i, R_{i,0}, R_{i,1}, h_{ID_i}\rangle \in L_{H1}$, the simulator responds to this query following the record, such that

$$\mathcal{H}_1(ID_i, R_{i,0}, R_{i,1},) = h_{ID_i}.$$

  Otherwise, the simulator randomly selects $h_{ID_i} \in \mathbb{Z}_p^*$ and stores $\langle ID_i, R_{i,0},\ R_{i,1}, h_{ID_i}\rangle$ into L_{H1}. The simulator returns $\mathcal{H}_1(ID_i, R_{i,0}, R_{i,1},) = h_{ID_i}$.

$\mathcal{H}_2(ID_i, m_i, A_{i,0}, A_{i,1})$: On an i-th random oracle query, i.e. on input $(ID_i, m_i, A_{i,0}, A_{i,1})$. If the tuple exists, such that $\langle ID_i, m_i, A_{i,0}, A_{i,1}, h_{m_i}\rangle \in L_{H2}$, the simulator responds to this query following the record, such that

$$\mathcal{H}_2(ID_i, m_i, A_{i,0}, A_{i,1}) = h_{ID_i,m_i}$$

Otherwise, the simulator randomly selects $h_{ID_i,m_i} \in \mathbb{Z}_p^*$ and stores $\langle ID_i, m_i, A_{i,0}, A_{i,1}, h_{m_i}\rangle$ into L_{H2}. The simulator returns $\mathcal{H}_2(ID_i, m_i, A_{i,0}, A_{i,1}) = h_{ID_i,m_i}$.

– **Query Phase.** During the query phase, the adversary may adaptively make extraction queries and signing queries. The simulator prepares an empty set L_E to record the simulated user's private keys.

- $\mathcal{O}_E(ID_i)$: On an i-th extraction query (ID_i), the simulator checks whether $\langle ID_i, d_{ID_i}\rangle \in L_E$ exits. If there is, the simulator retrieves the user's private key $d_{ID_i} = (R_{i,0}, R_{i,1}, b_i, y_i)$, which is based on the definition as follows. Otherwise, it first randomly selects $b_i \in \{0,1\}$ and $r'_{i,0}, r'_{i,1}, h_{ID_i} \in \mathbb{Z}_p^*$. It computes $R_{i,0}, R_{i,1}$ in either cases, such that

$$\begin{cases} b=0, & R_{i,0} = g^{r'_{i,0}} \cdot (g^{\alpha})^{-h_{ID_i}},\ R_{i,1} = g^{r'_{i,1}}, \\ b=1, & R_{i,0} = g^{r'_{i,0}},\ R_{i,1} = g^{r'_{i,1}} \cdot (g^{\alpha})^{-h_{ID_i}}. \end{cases}$$

It then sets $y_i = r'_{i,b_i}$ and programs hash oracle as follows

$$H_1(ID_i, R_{i,0}, R_{i,1}) = h_{ID_i}.$$

The simulator returns user private key $d_{ID_i} = (R_{i,0}, R_{i,1}, b_i, y_i)$ and stores $\{ID_i, d_{ID_i}\}$ into a list of user's private key L_E and $\{ID_i, R_{i,0}, R_{i,1}, h_{ID_i}\}$ into L_{H1}. It is worth noting that the extraction query may fail if the hash value h_{ID_i} was called by the adversary. However, since $R_{i,0}, R_{i,1}$ are randomly selected by the simulator, this abort probability is negligible.

- $\mathcal{O}_S(ID_i, m_i)$: On an i-th signing query (ID_i, m_i). It checks whether $\langle ID_i, d_{ID_i}\rangle \in L_E$. It calls $\mathcal{O}_E(ID_i) \rightarrow d_{ID_i}$ to generate a fresh user private key if it does not exist. Otherwise, it retrieves $d_{ID_i} = (R_{i,0}, R_{i,1}, b_i, y_i)$ from L_E. Let $h_{ID_i} = H_1(ID_i, R_{i,0}, R_{i,1})$. It randomly selects $a'_{i,0}, a'_{i,1}, c_{i,1-b} \in \mathbb{Z}_p^*$, and based on $b_i \in \{0,1\}$, it sets

$$\begin{cases} b_i = 0, & A_{i,0} = g^{a'_{i,0}}, \quad A_{i,1} = g^{a'_{i,1}}(R_{i,1} \cdot Z^{h_{ID_i}})^{-c_{i,1-b}}, \\ & s_{i,1} = a'_{i,1}, \\ b_i = 1, & A_{i,0} = g^{a'_{i,0}}(R_{i,0} \cdot Z^{h_{ID_i}})^{-c_{i,1-b}}, \quad A_{i,1} = g^{a'_{i,1}}, \\ & s_{i,0} = a'_{i,0}. \end{cases}$$

It then calls for hash oracle and obtains $h_{ID_i,m_i} = H_2(ID_i, m_i, A_{i,0}, A_{i,1})$ and sets

$$c_{b_i} = h_{ID_i,m_i} - c_{i,1-b_i}, \quad s_{b_i} = a'_{b_i} + y_i \cdot c_{b_i}.$$

The signature σ_{ID_i,m_i} is returned as follows.

$$\sigma_{ID_i,m_i} = ((R_{i,0},\ A_{i,0},\ s_{i,0},\ c_{i,0}),\ (R_{i,1},\ A_{i,1},\ s_{i,1},\ c_{i,1})).$$

– **Forgery Phase.** Assume the algebraic adversary returns a forged signature $\sigma_{ID^*,m^*} = ((R_0^*, A_0^*, s_0^*, c_0^*), (R_1^*, A_1^*, s_1^*, c_1^*))$ on (ID^*, m^*), such that Eqs. (2), (3), and (4) hold, and representations $\vec{u}, \vec{v}, \vec{x}, \vec{y}$, i.e., for $\pi \in [u, v, x, y]$

$$\vec{\pi} = (\pi_0, \pi_1, \pi_{2,1}, ..., \pi_{2,q_e}, \pi_{3,1}, \pi_{3,q_e}, \pi_{4,1}, ..., \pi_{4,q_s},$$
$$\pi_{5,1}, ..., \pi_{5,q_s}, \pi_{6,1}, ..., \pi_{6,q_s}, \pi_{7,1}, ..., \pi_{7,q_s}),$$

that describe the linear combination of $R_0^*, R_1^*, A_0^*, A_1^*$, respectively, such that

$$R_0^* = g^{u_0} Z^{u_1} \prod_{i=1}^{q_e} (R_{i,0})^{u_{2,i}} (R_{i,1})^{u_{3,i}} \prod_{i=1}^{q_s} (R_0^*)^{u_{4,i}} (R_1^*)^{u_{5,i}} (A_{i,0})^{u_{6,i}} (A_{i,1})^{u_{7,i}},$$
$$R_1^* = g^{v_0} Z^{v_1} \prod_{i=1}^{q_e} (R_{i,0})^{v_{2,i}} (R_{i,1})^{v_{3,i}} \prod_{i=1}^{q_s} (R_0^*)^{v_{4,i}} (R_1^*)^{v_{5,i}} (A_{i,0})^{v_{6,i}} (A_{i,1})^{v_{7,i}},$$
$$A_0^* = g^{x_0} Z^{x_1} \prod_{i=1}^{q_e} (R_{i,0})^{x_{2,i}} (R_{i,1})^{x_{3,i}} \prod_{i=1}^{q_s} (R_0^*)^{x_{4,i}} (R_1^*)^{x_{5,i}} (A_{i,0})^{x_{6,i}} (A_{i,1})^{x_{7,i}},$$
$$A_1^* = g^{y_0} Z^{y_1} \prod_{i=1}^{q_e} (R_{i,0})^{y_{2,i}} (R_{i,1})^{y_{3,i}} \prod_{i=1}^{q_s} (R_0^*)^{y_{4,i}} (R_1^*)^{y_{5,i}} (A_{i,0})^{y_{6,i}} (A_{i,1})^{y_{7,i}},$$

where q_e is the number of extraction queries and q_s is the number is signing queries been made. Note that for simplicity, it is reasonable to assume the adversary obtains signatures on chosen ID^* from the signing query $\sigma_{ID^*,m_i} \leftarrow \mathcal{O}_S(ID^*, m_i)$, and for signatures on other ID_i, the adversary may query user's private key $d_{ID_i} \leftarrow \mathcal{O}_E(ID_i)$ and compute the corresponding valid signatures.

To simplify the above definition, suppose representations $\vec{u}, \vec{v}, \vec{x}, \vec{y}$ and chosen randomness $r'_{i,0}, r'_{i,1}, r'^*_0, r'^*_1, a'_{i,0}, a'_{i,1}, c_{i,0}, c_{i,1}, h_{ID_i}, h_{ID^*} \in \mathbb{Z}_p^*$ are sorted by unknown α, such that $\pi_{u,1}, \pi_{v,1}, \pi_{x,1}, \pi_{y,1} \in \mathbb{Z}_p$ are defined as coefficients of α, and $\pi_{u,0}, \pi_{v,0}, \pi_{x,0}, \pi_{y,0} \in \mathbb{Z}_p$ are the remaining coefficients. Therefore, $R_0^*, R_1^*, A_0^*, A_1^*$ can be simplified as follows

$$R_0^* = g^{\alpha\pi_{u,1}+\pi_{u,0}}, \quad R_1^* = g^{\alpha\pi_{v,1}+\pi_{v,0}},$$
$$A_0^* = g^{\alpha\pi_{x,1}+\pi_{x,0}}, \quad A_1^* = g^{\alpha\pi_{y,1}+\pi_{y,0}}.$$

By combining the forgery and representations and verification Eqs. (2), (3) and (4), such that

$$g^{s_0} = A_0(R_0 \cdot Z^{h_{ID}})^{c_0},$$
$$g^{s_1} = A_1(R_1 \cdot Z^{h_{ID}})^{c_1},$$
$$h_{ID,m} = c_0 + c_1$$

The simulator can obtain the following three modular equations

$$\begin{aligned} s_0^* &= \alpha\pi_{x,1} + \pi_{x,0} + c_0^*(\alpha\pi_{u,1} + \pi_{u,0} + \alpha h_{ID^*}), \\ s_1^* &= \alpha\pi_{y,1} + \pi_{y,0} + c_1^*(\alpha\pi_{v,1} + \pi_{v,0} + \alpha h_{ID^*}), \\ h_{ID^*,m^*} &= c_0^* + c_1^*, \end{aligned}$$

where $h_{ID^*} = H_1(ID^*, R_0^*, R_1^*)$ and $h_{ID^*,m^*} = H_2(ID^*, m^*, A_0^*, A_1^*)$.

Abort. The simulator aborts if the following abort condition holds, such that coefficients of α are zero, i.e.

$$\begin{aligned} &\pi_{x,1} + c_0^*(\pi_{u,1} + h_{ID^*}) = 0, \text{ and} \\ &\pi_{y,1} + c_1^*(\pi_{v,1} + h_{ID^*}) = 0. \end{aligned}$$

Otherwise, the simulator can solve for DL solution α via either of the following equations, such that

$$\alpha = \frac{s_0^* - (\pi_{x,0} + c_0^*\pi_{u,0})}{\pi_{x,1} + c_0^*(\pi_{u,1} + h_{ID^*})}, \text{ or} \tag{5}$$

$$\alpha = \frac{s_1^* - (\pi_{y,0} + c_1^*\pi_{v,0})}{\pi_{y,1} + c_1^*(\pi_{v,1} + h_{ID^*})}. \tag{6}$$

Indistinguishable Simulation. The simulation is correct as the correctness of every simulated user's private key and signature hold, which has been described in the query phase. The simulator sets all elements with perfectly hidden randomness. For example, master public key $Z = g^z$, user's private key randomness $R_{i,0} = g^{r_{i,0}}, R_{i,1} = g^{r_{i,1}}$ and signature randomnesses $R_0^* = g^{r_0^*}, R_1^* = g^{r_1^*}, A_{i,0} = g^{a_{i,0}}, A_{i,1} = g^{a_{i,1}}$. They are, respectively, simulated based on random bit $b_i, b^* \in \{0,1\}$ as

$$\begin{aligned} z = \alpha, \quad & r_{i,0} = r'_{i,0} - \alpha h_{ID_i}(1 - b_i), \quad r_{i,1} = r'_{i,1} - \alpha h_{ID_i}(b_i), \\ & r_0^* = r_0'^* - \alpha h_{ID^*} \cdot (1 - b^*), \quad r_1^* = r_1'^* - \alpha h_{ID^*}(b^*), \\ a_{i,0} = a'_{i,0} - c_{i,1-b^*}(r_0^* + \alpha h_{ID^*})(b^*), & \quad a_{i,1} = a'_{i,1} - c_{i,1-b^*}(r_1^* + \alpha h_{ID^*})(1 - b^*). \end{aligned}$$

While $b_i, b^* \in \{0,1\}$ and $\alpha, r'_{i,0}, r'_{i,1}, r_0'^*, r_1'^*, a'_{i,0}, a'_{i,1}, c_{i,0}, c_{i,1}, h_{ID_i}, h_{ID^*} \in \mathbb{Z}_p^*$ are all randomly chosen by the simulator, group elements $Z, R_{i,0}, R_{i,1}, R_0^*, R_1^*$, $A_{i,0}, A_{i,1}$ are random and indistinguishable from the view of the adversary. Hence, the simulation is indistinguishable from the proposed IBS scheme.

Probability of Aborts. Based on Eqs. (5) and (6), we note that the simulator aborts if the following abort condition holds, such that

$$\pi_{x,1} + c_0^*(\pi_{u,1} + h_{ID^*}) = \pi_{y,1} + c_1^*(\pi_{v,1} + h_{ID^*}) = 0. \tag{7}$$

Therefore, we classify the forgery and representations into three cases as follows to analyze the probability of setting Eqs. (7) holds.

Case 1. The first case is to analyze the probability that both $\pi_{u,1} + h_{ID^*} = 0$ and $\pi_{v,1} + h_{ID^*} = 0$ hold, which would also implies that setting $\pi_{x,1} = \pi_{y,1} = 0$ based

on Eq. (7). Recall that $R_0^* = g^{\alpha\pi_{u,1}+\pi_{u,0}}$ and $R_1^* = g^{\alpha\pi_{v,1}+\pi_{v,0}}$ are algebraically computed, and they must be submitted to hash oracles $H_1(ID^*, R_0^*, R_1^*) = h_{ID^*}$ in advance. While $h_{ID^*} \in \mathbb{Z}_p^*$ is randomly chosen after R_0^*, R_1^* and $\vec{u}, \vec{v}$ were submitted to hash oracles, the success probability of setting both $\pi_{u,1} = \pi_{v,1} = -h_{ID^*}$ is negligible, such that $\Pr[\pi_{u,1} = \pi_{v,1} = -h_{ID^*}] = \frac{1}{p}$.

Case 2. Now, we analyze the case that setting $\pi_{u,1}+h_{ID^*} \neq 0$ and $\pi_{v,1}+h_{ID^*} \neq 0$ hold. Based on Eq. (7), we then need to analyze

$$\pi_{x,1} + c_0^*(\pi_{u,1} + h_{ID^*}) = 0, \quad \text{and} \quad \pi_{y,1} + c_1^*(\pi_{v,1} + h_{ID^*}) = 0.$$

The above equations hold if $\pi_{x,1} = -c_0^*(\pi_{u,1} + h_{ID^*})$ and $\pi_{y,1} = -c_1^*(\pi_{v,1} + h_{ID^*})$ are set. Recall that $H_2(ID^*, m^*, A_0^*, A_1^*) = h_{ID^*,m^*} = c_0^* + c_1^*$, we can observe that the above equations hold if the adversary could launch the following behaviours $\mathcal{F}_0, \mathcal{F}_1$, such that

$$\begin{aligned} \mathcal{F}_0: \quad \pi_{x,1} &= -(h_{ID^*,m^*} - c_1^*)(\pi_{u,1} + h_{ID^*}), \text{ and} \\ \mathcal{F}_1: \quad \pi_{y,1} &= -(h_{ID^*,m^*} - c_0^*)(\pi_{v,1} + h_{ID^*}). \end{aligned}$$

Recall that $A_0^* = g^{\alpha\pi_{x,1}+\pi_{x,0}}$ and $A_1^* = g^{\alpha\pi_{y,1}+\pi_{y,0}}$ are algebraically computed, and they must be submitted to hash oracles $H_2(ID^*, m^*, A_0^*, A_1^*) = h_{ID^*,m^*}$ in advance. While $h_{ID^*,m^*} \in \mathbb{Z}_p^*$ is randomly chosen after A_0^*, A_1^* and $\vec{x}, \vec{y}$ were submitted to random oracles, the success probability of launching both behaviours $\mathcal{F}_0, \mathcal{F}_1$ is negligible, such that $\Pr[\mathcal{F}_0 \wedge \mathcal{F}_1] = \frac{1}{p}$.

Case 3. Lastly, we analyze the case that setting either $\pi_{u,1} + h_{ID^*} = 0$ or $\pi_{v,1} + h_{ID^*} = 0$ holds. This would be possible when $\mathcal{O}_S(ID^*, m_i)$ has been made, such that $d_{ID^*} = (R_0^*, R_1^*, b^*, y^*)$ was simulated by the simulator. So R_0^*, R_1^* must become part of the forgery. It is worth noting that both $b^* \in \{0, 1\}$ and $y^* \in \mathbb{Z}_p^*$ are information-theoretically hidden from the view of the adversary. This means that for some $r_0'^*, r_1'^*, h_{ID^*} \in \mathbb{Z}_p^*$,

$$\begin{cases} b^* = 0, & R_0^* = g^{r_0'^*} \cdot (g^\alpha)^{-h_{ID^*}}, \ R_1^* = g^{r_1'^*}, \\ b^* = 1, & R_0^* = g^{r_0'^*}, \ R_1^* = g^{r_1'^*} \cdot (g^\alpha)^{-h_{ID^*}}. \end{cases}$$

The hash oracle is programmed, i.e. $h_{ID^*} = H_1(ID^*, R_0^*, R_1^*)$ is set. Based on previous definition, $R_0^* = g^{\alpha\pi_{u,1}+\pi_{u,0}}$ and $R_1^* = g^{\alpha\pi_{v,1}+\pi_{v,0}}$, this indicates that $\pi_{u,1} = -h_{ID^*} \cdot (1-b^*)$ and $\pi_{v,1} = -h_{ID^*} \cdot b^*$ hold. We see Eq. (7) can be rewritten as follows, such that

$$\begin{aligned} \pi_{x,1} + c_0^*(-h_{ID^*} \cdot (1 - b^*) + h_{ID^*}) = 0, \text{ and} \\ \pi_{y,1} + c_1^*(-h_{ID^*} \cdot b^* + h_{ID^*}) = 0. \end{aligned}$$

Recall that $A_0^* = g^{\alpha\pi_{x,1}+\pi_{x,0}}$ and $A_1^* = g^{\alpha\pi_{y,1}+\pi_{y,0}}$. Therefore, the above equations hold if the adversary could launch either one of the following behaviours $\mathcal{G}_0, \mathcal{G}_1$ correctly, such that

$$\begin{cases} b^* = 0, & \mathcal{G}_0: \quad \pi_{x,1} = 0, & \pi_{y,1} = -c_1^* h_{ID^*}, \\ b^* = 1, & \mathcal{G}_1: \quad \pi_{x,1} = -c_0^* h_{ID^*}, & \pi_{y,1} = 0. \end{cases}$$

Fortunately, the adversary succeeds depending on the random guessing $b' \in \{0, 1\}$, i.e. either launching $\mathcal{G}_0$ or $\mathcal{G}_1$. Since b^* is information-theoretically hidden from the view of the adversary, the success probability is half, such that $\Pr[b^* = b'] = \frac{1}{2}$.

The Security Loss. The security loss is calculated based on the success probability of the simulation. We describe the success probability of extracting the DL solution, according to the above probability of abort, as follows

$$\begin{aligned}\Pr[\text{Success}] &= 1 - (\Pr[\textbf{Case 1}] + \Pr[\textbf{Case 2}] + \Pr[\textbf{Case 3}]) \\ &= 1 - (\frac{1}{p} + \frac{1}{p} + \frac{1}{2}) \approx \frac{1}{2}.\end{aligned}$$

Therefore, the success probability of extracting the DL problem solution is at least $\frac{1}{2}$, which concludes that the security loss of the proposed IBS scheme under EUF-CMA and DL assumption in AGM is 2. This completes the proof of the Theorem 1. □

4 Conclusion

In this work, through the adoption of the AGM, we addressed the challenge of achieving tight security reductions for pairing-free IBS under EUF-CMA and DL assumption. We obtained the first pairing-free IBS scheme, which is inspired by using an OR-proof technique to generate the user's private keys for the Schnorr-like IBS. We also proposed a tight reduction algorithm that shows how the obtained pairing-free IBS can achieve tight EUF-CMA security under the DL assumption in the AGM.

This work provided insight that one must carefully analyze the security of a scheme when proving security in the AGM. For example, the Schnorr-like IBS scheme is constructed based on Schnorr signatures, although both schemes are proven under DL assumption, the additional algorithm and chosen-identity attacks in the ID-based setting may lead to a loose reduction. It is noteworthy to mention that, as indicated in the summary provided in Table 1, among the current theoretical studies, adopting the AGM may be necessary to achieve IBS schemes under the DL assumption and EUF-CMA security with tight reductions.

Acknowledgement. We extend our gratitude to the anonymous reviewers for their valuable feedback.

References

1. Ahed, K., Benamar, M., El Ouazzani, R.: Content delivery in named data networking based internet of things. In: 2019 15th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1397–1402. IEEE (2019)
2. Bacho, R., Loss, J.: On the adaptive security of the threshold BLS signature scheme. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 193–207 (2022)

3. 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). https://doi.org/10.1007/11593447_28
4. Bellare, M., Dai, W.: Chain reductions for multi-signatures and the HBMS scheme. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021. LNCS, vol. 13093, pp. 650–678. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92068-5_22
5. Bellare, M., Namprempre, C., Neven, G.: Security proofs for identity-based identification and signature schemes. J. Cryptol. **22**(1), 1–61 (2009)
6. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62–73 (1993)
7. Beth, T.: Efficient zero-knowledge identification scheme for smart cards. In: Barstow, D., Brauer, W., Brinch Hansen, P., Gries, D., Luckham, D., Moler, C., Pnueli, A., Seegmüller, G., Stoer, J., Wirth, N., Günther, C.G. (eds.) EUROCRYPT 1988. LNCS, vol. 330, pp. 77–84. Springer, Heidelberg (1988). https://doi.org/10.1007/3-540-45961-8_7
8. Chatterjee, S., Kamath, C., Kumar, V.: Galindo-Garcia identity-based signature revisited. In: Kwon, T., Lee, M.-K., Kwon, D. (eds.) ICISC 2012. LNCS, vol. 7839, pp. 456–471. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37682-5_32
9. Choon, J.C., Hee Cheon, J.: An identity-based signature from gap Diffie-Hellman groups. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 18–30. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36288-6_2
10. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48658-5_19
11. Crites, E., Komlo, C., Maller, M.: How to prove Schnorr assuming Schnorr: Security of multi-and threshold signatures. Cryptology ePrint Archive (2021)
12. Crites, E., Komlo, C., Maller, M.: Fully adaptive Schnorr threshold signatures. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023. LNCS, vol. 14081, pp. 678–709. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38557-5_22
13. Crites, E., Komlo, C., Maller, M., Tessaro, S., Zhu, C.: Snowblind: a threshold blind signature in pairing-free groups. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023. LNCS, vol. 14081, pp. 710–742. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38557-5_23
14. Du, H., Wen, Q.: An efficient identity-based short signature scheme from bilinear pairings. In: 2007 International Conference on Computational Intelligence and Security (CIS 2007), pp. 725–729. IEEE (2007)
15. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10992, pp. 33–62. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96881-0_2
16. Fuchsbauer, G., Plouviez, A., Seurin, Y.: Blind Schnorr signatures and signed ElGamal encryption in the algebraic group model. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020. LNCS, vol. 12106, pp. 63–95. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45724-2_3
17. Fukumitsu, M., Hasegawa, S.: A Galindo-Garcia-like identity-based signature with tight security reduction. In: 2017 Fifth International Symposium on Computing and Networking (CANDAR), pp. 87–93. IEEE (2017)

18. Fukumitsu, M., Hasegawa, S.: A Galindo-Garcia-like identity-based signature with tight security reduction, revisited. In: 2018 Sixth International Symposium on Computing and Networking (CANDAR), pp. 92–98. IEEE (2018)
19. Galindo, D., Garcia, F.D.: A Schnorr-like lightweight identity-based signature scheme. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 135–148. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02384-2_9
20. Gjøsteen, K., Jager, T.: Practical and tightly-secure digital signatures and authenticated key exchange. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10992, pp. 95–125. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96881-0_4
21. Hess, F.: Efficient identity based signature schemes based on pairings. In: Nyberg, K., Heys, H. (eds.) SAC 2002. LNCS, vol. 2595, pp. 310–324. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36492-7_20
22. Kastner, J., Loss, J., Xu, J.: The Abe-Okamoto partially blind signature scheme revisited. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022. LNCS, vol. 13794, pp. 279–309. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-22972-5_10
23. Kastner, J., Loss, J., Xu, J.: On pairing-free blind signature schemes in the algebraic group model. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022. LNCS, vol. 13178, pp. 468–497. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-97131-1_16
24. Kılınç Alper, H., Burdges, J.: Two-round trip Schnorr multi-signatures via delinearized witnesses. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12825, pp. 157–188. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84242-0_7
25. Kiltz, E., Masny, D., Pan, J.: Optimal security proofs for signatures from identification schemes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 33–61. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_2
26. Kiltz, E., Neven, G.: Identity-based signatures. In: Joye, M., Neven, G. (eds.) Identity-Based Cryptography, Cryptology and Information Security Series, vol. 2, pp. 31–44. IOS Press (2009). https://doi.org/10.3233/978-1-58603-947-9-31
27. Lee, K., Kim, H.: Two-round multi-signature from Okamoto signature. Cryptology ePrint Archive (2022)
28. Loh, J.C., Guo, F., Susilo, W., Yang, G.: A tightly secure id-based signature scheme under dl assumption in AGM. In: Simpson, L., RezazadehBaee, M.A. (eds.) ACISP 2023. LNCS, vol. 13915, pp. 199–219. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-35486-1_10
29. Naccache, D., Pointcheval, D., Stern, J.: Twin signatures: an alternative to the hash-and-sign paradigm. In: Proceedings of the 8th ACM Conference on Computer and Communications Security, pp. 20–27 (2001)
30. Nick, J., Ruffing, T., Seurin, Y.: MuSig2: simple two-round Schnorr multi-signatures. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12825, pp. 189–221. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84242-0_8
31. Nour, B., et al.: Internet of things mobility over information-centric/named-data networking. IEEE Internet Comput. **24**(1), 14–24 (2019)
32. Oliveira, L.B., et al.: TinyPBC: pairings for authenticated identity-based non-interactive key distribution in sensor networks. Comput. Commun. **34**(3), 485–493 (2011)
33. Paillier, P., Vergnaud, D.: Discrete-log-based signatures may not be equivalent to discrete log. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 1–20. Springer, Heidelberg (2005). https://doi.org/10.1007/11593447_1

34. Paterson, K.G.: Id-based signatures from pairings on elliptic curves. Electron. Lett. **38**(18), 1025–1026 (2002)
35. Paterson, K.G., Schuldt, J.C.N.: Efficient identity-based signatures secure in the standard model. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 207–222. Springer, Heidelberg (2006). https://doi.org/10.1007/11780656_18
36. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. J. Cryptol. **13**(3), 361–396 (2000)
37. Rahman, S.M.M., El-Khatib, K.: Private key agreement and secure communication for heterogeneous sensor networks. J. Parallel Distrib. Computi. **70**(8), 858–870 (2010)
38. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_22
39. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
40. Tessaro, S., Zhu, C.: Short pairing-free blind signatures with exponential security. Cryptology ePrint Archive (2022)
41. Xiong, W., Wang, R., Wang, Y., Zhou, F., Luo, X.: CPPA-D: efficient conditional privacy-preserving authentication scheme with double-insurance in VANETs. IEEE Trans. Veh. Technol. **70**(4), 3456–3468 (2021)
42. Zhang, X., Liu, S., Gu, D., Liu, J.K.: A generic construction of tightly secure signatures in the multi-user setting. Theoret. Comput. Sci. **775**, 32–52 (2019)