A Tightly Secure ID-Based Signature Scheme Under DL Assumption in AGM

Jia-Chng Loh[1(✉)], Fuchun Guo[1], Willy Susilo[1], and Guomin Yang[2]

[1] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, Australia
{jial,fuchun,wsusilo}@uow.edu.au
[2] Singapore Management University, Singapore, Singapore
gmyang@smu.edu.sg

Abstract. Identity-based signatures (IBS) can be verified using the signer identity information as the public key, and hence, there is no need for certificate management that proves the corresponding public key ownership. Unfortunately, none of the existing IBS schemes has security proven as tight as the discrete logarithm (DL) problem, *the hardest problem* in the cyclic group setting, under the standard EUF-CMA security model. Recently, the introduction of proving security in the algebraic group model (AGM), where the adversary's computation is algebraic, enables some ordinary signature schemes to be proven tightly reducible under DL assumption and EUF-CMA. To date, however, it remains unknown whether IBS schemes can also be proven as secure as the DL problem in the AGM. Achieving tight security in IBS schemes under standard EUF-CMA is challenging, due to the need to take extra precautions against adaptive queries on user private keys by the adversary. In this work, we show, for the first time, an IBS scheme with tight security under DL assumption and EUF-CMA in the AGM. The scheme features a minimal signature size of two group elements, with a reduction loss factor of two.

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

1 Introduction

Identity-Based Signature with the Ideal Security. Digital signature is one of the main cryptographic primitives to enable authenticated electronic communications [15]. However, the public key infrastructure (PKI) is necessary to provide a certificate verifying the validity of the signer's public key in practical use. The notion of identity-based signatures (IBS) was introduced by Shamir [37], where the signer's identity, such as an ID number or email address, serves as the public key, eliminating the need for a certificate verifying the signer's authority. Therefore, IBS can be publicly verified via the signer's identity. The standard security model for IBS, namely existential unforgeability against chosen identity-and-message attacks (EUF-CMA) [5], guarantees that it's impossible to forge a

L. Simpson and M. A. Rezazadeh Baee (Eds.): ACISP 2023, LNCS 13915, pp. 199–219, 2023.
https://doi.org/10.1007/978-3-031-35486-1_10

new signature on a unique message and identity pair, even if the attacker has access to many user private keys and many signatures.

When proving the security of a cryptographic scheme, the security reduction is tight if security loss L is a constant or a small number $L \geq 1$. Security loss is an important factor when it comes to deciding the concrete size of the security parameter of a scheme in practice because we must increase the size of the security parameter so that it compensates for the security loss, unfortunately, it degrades the scheme efficiency. On the other hand, the security of a scheme is determined by the difficulty of the underlying problem. The harder the problem, the stronger the security provided by the scheme. Considering the cyclic groups setting, it is known that the discrete logarithm (DL) problem is the hardest one because algorithms that can solve the DL problem can be used to solve the computational Diffie-Hellman (CDH) problem and its variants [22,28]. Therefore, schemes with ideal security, i.e., as secure as the DL problem in the standard EUF-CMA security model, offer the strongest security assurance in the context of cyclic groups.

There are several IBS schemes that have been proven secure under the EUF-CMA security model and DL assumption [5,12,20,31] or CDH assumption and its variants [3,13,23,27,33,34]. However, none of these schemes achieve the ideal security, meaning their security cannot be tightly reduced to the DL problem under EUF-CMA.

The Algebraic Group Model. The concept of the algebraic algorithm was initially introduced by Boneh and Venkatesan in [9]. It has since been employed in several studies, including those that aim to demonstrate impossibilities [1,9,10, 14,21,26,32]. The formalization of the algebraic group model (AGM) was later carried out by Fuchsbauer, Kiltz, and Loss in [16]. In the AGM, the security of a scheme is proven through a security reduction that demonstrates how a probabilistic polynomial-time (PPT) adversary can compromise the scheme, by idealizing the adversary's computations as algebraic, yielding representations that describe how the output group element can be generated from received group elements.

In the EUF-CMA security model, due to the algebraic nature of the adversary's forgeries and representations, some standard signature schemes such as BLS [8] and Schnorr [36] can be reduced in the AGM with random oracles to extract DL solutions with an overwhelming success probability. This has been shown in works such as [16,17], respectively. On the other hand, variants of secure signature schemes have been analyzed in the AGM [2,4,24,25,30]. However, to date, the security of identity-based signatures (IBS) in the AGM has not yet been studied. Hence, it is unknown whether the security of IBS schemes can also be proven as tight as the DL problem in the AGM under EUF-CMA.

Challenge. It is not a trivial task to have tight security for IBS schemes, especially in the standard EUF-CMA security model, due to the adversary may adaptively ask for user private keys during the query phase. Therefore, a major challenge in achieving tight reduction for IBS under the DL assumption and EUF-CMA is developing a reduction algorithm that can simultaneously (1)

respond to user private key queries and (2) extract problem solutions based on forged signatures. However, these two tasks are often conflicting, if the user private key is simulated, the forged signature is often not reducible in security reductions.

Existing reduction techniques [11,13,20,27,34] address the first challenge by dividing the simulator's handling of user identity and signature queries, resulting in at least one target forgery with a non-simulatable user private key. If the chosen identity of the forgery matches the target, the simulator can extract the problem solution during the forgery phase, bypassing the second challenge. However, this approach leads to a loose reduction due to the random selection of the target identity and a security loss of at least q query times.

1.1 Contribution

In this work, we show, for the first time, how to obtain a tightly secure IBS scheme under DL assumption and EUF-CMA by adopting the algebraic adversary. The contributions are summarized as follows.

We first present a new IBS scheme, named BLS-IBS, which is extended from the ordinary BLS signature scheme [8] to the identity-based setting. This results in signatures consisting of only two group elements. For example, let $(p, g, \mathbb{G})$ be a bilinear group tuple in prime order p with a generator $g \in \mathbb{G}$ and $\mathcal{H}_1, \mathcal{H}_2 :\rightarrow \mathbb{G}$ be cryptographic hash functions. The signature $\sigma_{ID,m} = (\sigma^{(1)}_{ID,m}, \sigma^{(2)}_{ID,m})$ on identity and message pair (ID, m) is defined as

$$\sigma^{(1)}_{ID,m} = d_{ID} \cdot \mathcal{H}_2(m)^r, \quad \sigma^{(2)}_{ID,m} = g^r,$$

where $d_{ID} = \mathcal{H}_1(ID)^x$ is the user private key that can be obtained based on the BLS signatures with master secret key $x \in \mathbb{Z}_p$.

To demonstrate that the security of the BLS-IBS scheme can be tightly reduced to the DL problem, we propose a security reduction in the AGM that addresses the two aforementioned challenges. Specifically,

- Our reduction can simulate any user private key, which enables the simulator to respond to all user private key queries and signature queries without aborting during the query phase of the EUF-CMA security game, regardless of the identities that have been queried for signatures.
- It is possible to reduce any forgery, even if the private key of the forged identity is simulatable. The algebraic adversary's forgery and representations play a crucial role in this reduction process, helping the simulator find a solution to the problem.

It is also worth noting that the security analysis in the AGM of BLS-IBS is more complex than that of ordinary BLS in the AGM due to the fact that the adversary's forgery and representations can be classified into several cases, rather than just two cases. In particular, we design two indistinguishable simulations in the AGM with random oracles that can simulate any user private key and extract DL solution with at least half of the success probability, resulting in a tight reduction with a loss factor of two.

1.2 Other Related Work

IBS Schemes Under DL Assumption and EUF-CMA. The first secure EUF-CMA 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 [20] proposed a new IBS scheme based on the well-known Schnorr signature [36] in the random oracle model [6], 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. [11], it is still loosely reduced to the DL problem due to the need for the reset lemma [35], which has been proven that the resulting security loss must suffer from the tightness barrier – the reduction loss cannot be tight [26].

IBS Schemes Under CDH Assumption. The first IBS scheme without random oracles was proposed by Paterson and Schuldt [34], based on Waters' signatures [38], and secure under the EUF-CMA and CDH assumption. Narayan and Parampalli [29] further improved the scheme by reducing the size of the public parameters. Unfortunately, their security reductions are not tight and the computation cost is high. Recently, Yi et al. [39] proposed a new IBS scheme with a more efficient computation cost, but the reduction loss remains non-tight under the CDH assumption.

Tightly Secure IBS Schemes Under the Strong Assumption and Weak Security Model. Recently, Fukumitsu and Hasegawa [18,19] proposed enhancements to the Galindo and Garcia's Schnorr-like IBS [20] to design 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, and in the weak-EUF-CMA security model [40], where the adversary is restricted to asking for a user's private key if the identity has already been requested for signatures.

2 Preliminaries

2.1 Definition of Identity-Based Signatures

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

- $Setup(1^k)$: A setup algorithm that generates the master public and secret key pair (mpk, msk) when given the security parameters 1^k as input.
- $Extract(mpk, msk, ID)$: A user private key extraction algorithm that takes as input the master public and secret key pair (mpk, msk) and a user identity ID, and returns the user private key d_{ID}.
- $Sign(mpk, d_{ID}, m)$: An algorithm for generating a signature, which takes as input the master public key mpk, user's private key d_{ID} and message m, and outputs a signature $\sigma_{ID,m}$.
- $Verify(mpk, ID, m, \sigma_{ID,m})$: An algorithm for verifying inputs $(mpk, ID, m, \sigma_{ID,m})$ which outputs either 1 for acceptance or 0 for rejection.

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

2.2 Security of EUF-CMA for IBS

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

- **Setup Phase:** Let L_E be a set of extraction queries. 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 adaptively asks for user private keys d_{ID_i} on any identity ID_i to the extraction oracle $\mathcal{O}_E$ and signatures σ_{ID_i,m_i} on any identity and message pair (ID_i, m_i) to the signing oracle $\mathcal{O}_S$.
  - Extraction oracle $\mathcal{O}_E(ID_i)$: On input i-th query for ID_i, it first checks whether $\langle ID_i, \cdot \rangle \in L_E$ exists. The challenger retrieves user private key d_{ID_i} if it finds $\langle ID_i, d_{ID_i} \rangle$. 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 $(ID_i, d_{ID_i}) \notin L_E$, the challenger calls extraction algorithm $Extract(mpk, msk, ID_i) \rightarrow d_{ID_i}$ and stores user private key $\langle ID_i, d_{ID_i} \rangle$ to L_E. At the end, the challenger calls signing algorithm $Sign(mpk, d_{ID_i}, m_i) \rightarrow \sigma_{ID_i,m_i}$. The challenger returns σ_{ID_i,m_i}.
- **Forgery Phase:** The adversary returns a forged signature σ_{ID^*,m^*} on a chosen identity and message pair (ID^*, m^*). The adversary wins if the verification holds $Verify(mpk, ID^*, m^*, \sigma_{ID^*,m^*}) = 1$, such that ID^* has not been queried to $\mathcal{O}_E$ and (ID^*, m^*) has not been queried to $\mathcal{O}_S$.

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 an advantage at most ϵ in winning the game, where ϵ is a negligible function of the input security parameter.*

2.3 Bilinear Discrete Logarithm Problem

Let $\mathbb{G}, \mathbb{G}_T$ be groups of prime order p with generator $g \in \mathbb{G}$. A bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is defined in the pairing group setting. The discrete logarithm (DL) problem is to compute $a \in \mathbb{Z}_p$, given a random group element $g^a \in \mathbb{G}$.

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 an advantage at most ϵ to solve the DL problem in $\mathbb{G}$.*

2.4 The Algebraic Algorithms

Fuchsbauer, Kiltz, and Loss [16] formalized the algebraic group model (AGM), which considers the adversary's computation as algebraic. The definition of the algebraic algorithm is described as follows.

Definition 3. *(Algebraic Algorithm, Definition 2.1 of [16]) Suppose the adversary in the security game is algebraic, and it is given* $(X_0, X_1, ..., X_n) \in \mathbb{G}^{n+1}$ *group elements, such that* $X_0 = g \in \mathbb{G}$ *be the group generator. Whenever the adversary outputs* $Z \in \mathbb{G}$*, it also attaches a representation vector* $[\vec{\pi}] \in \mathbb{Z}_p^{n+1}$*, where* $[\vec{\pi}] = (\pi_0, \pi_1, ..., \pi_n) \in \mathbb{Z}_p^{n+1}$*, that indicates how* Z *is generated based on received group elements, such that* $Z = \prod_{i=0}^{n} X_i^{\pi_i}$*.*

3 The BLS-IBS Scheme

The BLS-IBS scheme is extended based on the BLS signature [8] to the identity-based setting. The scheme's algorithms are defined as follows.

- *Setup*: On input security parameter 1^k. Let $\mathbb{G}$ be the notion of groups, $g \in \mathbb{G}$ be the generator of group, p be the prime order, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map, where $\mathbb{G}_T$ denotes the *target group*, and $\mathcal{H}_1, \mathcal{H}_2 : \{0,1\}^* \rightarrow \mathbb{G}$ be two cryptography hash functions. It selects $x \in \mathbb{Z}_p^*$ and computes $X = g^x$. The master public and secret key pair (mpk, msk) is returned as

$$mpk = (g, p, e, \mathbb{G}, \mathbb{G}_T, \mathcal{H}_1, \mathcal{H}_2, X), \quad msk = x.$$

- *Extract*: On input (mpk, msk) and user identity $ID \in \{0,1\}^*$. It calls $\mathcal{H}_1(ID) = H_{ID} \in \mathbb{G}$ and computes user private key d_{ID} as

$$d_{ID} = (H_{ID})^x.$$

- *Sign*: On input (mpk, d_{ID}, m). It calls $\mathcal{H}_2(m) = H_m \in \mathbb{G}$ and randomly selects $r \in \mathbb{Z}_p^*$. The signature $\sigma_{ID,m} = (\sigma_{ID,m}^{(1)}, \sigma_{ID,m}^{(2)})$ on (ID, m) is returned as

$$\sigma_{ID,m}^{(1)} = d_{ID} \cdot (H_m)^r,$$
$$\sigma_{ID,m}^{(2)} = g^r.$$

- $Verify$: On input $(mpk, ID, m, \sigma_{ID,m})$, it checks whether or not the following equation holds

$$e(\sigma_{ID,m}^{(1)}, g) = e(H_{ID}, X) \cdot e(H_m, \sigma_{ID,m}^{(2)}).$$

Correctness. Recall that $X = g^x$ is part of mpk. The scheme is correct if the validity of signature $\sigma_{ID,m} = (\sigma_{ID,m}^{(1)}, \sigma_{ID,m}^{(2)})$ on identity and message pair (ID, m) generated with user private key d_{ID} holds.

$$\begin{aligned} e(\sigma_{ID,m}^{(1)}, g) &= e(H_{ID}, X) \cdot e(H_m, \sigma_{ID,m}^{(2)}) \\ &= e(\mathcal{H}_1(ID)^x \cdot \mathcal{H}_2(m)^r, g). \end{aligned}$$

4 Security Proof

In this section, we begin by presenting the high-level security proof for the BLS-IBS scheme in the AGM with random oracles under EUF-CMA and DL assumption. Then, we detail how we design two indistinguishable simulations that can simulate any user private key. Afterwards, we classify the algebraic adversary's forgery and representations into several cases and show that the simulator can extract for problem solution in any of them. Finally, we provide full security proof.

High-Level. Given a DL problem instance tuple (g, g^a), we design two indistinguishable simulations $\mathcal{R}_1, \mathcal{R}_2$, which control hash responses as the random oracles. We set simulation $\mathcal{R}_1$ to embed g^a into master public key X and every signature randomness $\sigma^{(2)}_{ID_i,m_i}$; or simulation $\mathcal{R}_2$ to embed g^a into every hash responses H_{ID_i}, H_{m_i} – including H_{ID^*}, H_{m^*}. The simulation setup is detailed as in Table 1, where all $x, h_{ID_i}, h_{m_i}, u_{1,i}, v_{1,i}, u_{2,i}, v_{2,i}, s_i, t_i, r_i \in \mathbb{Z}_p^*$ are randomly chosen. During the forgery phase, we classify the algebraic adversary's forgery and representations into several cases as illustrated in Fig. 1. By obtaining the discrete logarithm of any of the embedded elements through forgery and representations, we demonstrate that the DL problem solution can be found in either simulation $\mathcal{R}_1$ or $\mathcal{R}_2$.

Table 1. Simulations of the BLS-IBS scheme

Elements	Simulation $\mathcal{R}_1$	Simulation $\mathcal{R}_2$
X	g^a	g^x
H_{ID_i}	$g^{h_{ID_i}}$	$g^{au_{1,i}+v_{1,i}}$
H_{m_i}	$g^{h_{m_i}}$	$g^{au_{2,i}+v_{2,i}}$
d_{ID_i}	$g^{ah_{ID_i}}$	$g^{au_{1,i}x+v_{1,i}x}$
$\sigma^{(1)}_{ID_i,m_i}$	$g^{a(h_{ID_i}+h_{m_i}s_i)+h_{m_i}t_i}$	$g^{a(u_{1,i}x+u_{2,i}r_i)+v_{1,i}x+v_{2,i}r_i}$
$\sigma^{(2)}_{ID_i,m_i}$	$g^{as_i+t_i}$	g^{r_i}

In the security of the BLS-IBS scheme, for some randomly chosen x, h_{ID_i}, h_{m_i}, $r_i \in \mathbb{Z}_p^*$, let $X = g^x, H_{ID_i} = g^{h_{ID_i}}, H_{m_i} = g^{h_{m_i}}$, and $\sigma^{(2)}_{ID_i,m_i} = g^{r_i}$. It is easy to see every user private key d_{ID_i}, including that of the forged identity ID^*, is capable of being simulated in both simulations $\mathcal{R}_1$ and $\mathcal{R}_2$. As a result, our simulator will not abort during the query phase of the EUF-CMA security game as it is capable of responding to any user private key query or signature query.

During the forgery phase of the security game in the AGM and EUF-CMA security model, the algebraic adversary returns a forged signature $\sigma_{ID^*,m^*} = (\sigma^{(1)}_{ID^*,m^*}, \sigma^{(2)}_{ID^*,m^*}) = ((H_{ID^*})^x \cdot (H_{m^*})^{r^*}, g^{r^*})$ on a chosen identity and message pair (ID^*, m^*) and its representation vectors $[\vec{\alpha}], [\vec{\beta}]$ in $\mathbb{Z}_p$. Specifically, for some randomness $r^* \in \mathbb{Z}_p^*$, forgery σ_{ID^*,m^*} is defined as

$$\sigma^{(1)}_{ID^*,m^*} = g^{h_{ID^*}x+h_{m^*}r^*}, \quad \sigma^{(2)}_{ID^*,m^*} = g^{r^*}.$$

And forgery σ_{ID^*,m^*} can also be algebraically described with the corresponding representation vectors $[\vec{\alpha}], [\vec{\beta}]$ in $\mathbb{Z}_p$, such that

$$\begin{aligned}
\sigma^{(1)}_{ID^*,m^*} =& g^{\alpha_0} X^{\alpha_1} \prod_i^{q_{h1}} (H_{ID_i})^{\alpha_{2,i}} \prod_i^{q_{h2}} (H_{m_i})^{\alpha_{3,i}} \prod_i^{q_e} (d_{ID_i})^{\alpha_{4,i}} \\
& \prod_i^{q_s} (\sigma^{(1)}_{ID^*,m_i})^{\alpha_{5,i}} (\sigma^{(2)}_{ID^*,m_i})^{\alpha_{6,i}}, \\
\sigma^{(2)}_{ID^*,m^*} =& g^{\beta_0} X^{\beta_1} \prod_i^{q_{h1}} (H_{ID_i})^{\beta_{2,i}} \prod_i^{q_{h2}} (H_{m_i})^{\beta_{3,i}} \prod_i^{q_e} (d_{ID_i})^{\beta_{4,i}} \\
& \prod_i^{q_s} (\sigma^{(1)}_{ID^*,m_i})^{\beta_{5,i}} (\sigma^{(2)}_{ID^*,m_i})^{\beta_{6,i}},
\end{aligned}$$

where q_{h_1} is number of identity hash queries, q_{h_2} is number of message hash queries, q_e is number of extraction queries, and q_s is number of signing queries. Note that since the private keys of all users can be simulated, it is possible to exclude signatures queries for $(ID_i, \cdot)$ because the adversary can compute the signatures themselves with the knowledge of the user private keys d_{ID_i}, except for d_{ID^*} which the adversary is restricted to querying in the EUF-CMA security model.

Using the above definitions, the simulator can derive the following two modular equations.

$$\begin{aligned}
h_{ID^*}x + h_{m^*}r^* = \alpha_0 + x\alpha_1 + \sum_i^{q_{h1}} h_{ID_i}\alpha_{2,i} + \sum_i^{q_{h2}} h_{m_i}\alpha_{3,i} + \\
\sum_i^{q_e} h_{ID_i}x\alpha_{4,i} + \sum_i^{q_s} (h_{ID^*}x + h_{m_i}r_i)\alpha_{5,i} + r_i\alpha_{6,i}, \\
r^* = \beta_0 + x\beta_1 + \sum_i^{q_{h1}} h_{ID_i}\beta_{2,i} + \sum_i^{q_{h2}} h_{m_i}\beta_{3,i} + \\
\sum_i^{q_e} h_{ID_i}x\beta_{4,i} + \sum_i^{q_s} (h_{ID^*}x + h_{m_i}r_i)\beta_{5,i} + r_i\beta_{6,i}.
\end{aligned}$$

By rearranging with the term x and some $r_i \in [q_s]$, they can be simplified as

$$h_{ID^*}x + h_{m^*}r^* = x\theta + \hat{\theta} + \sum_i^{q_s} r_i\omega_{\alpha_i}, \tag{1}$$

$$r^* = x\delta + \hat{\delta} + \sum_i^{q_s} r_i\omega_{\beta_i}, \tag{2}$$

where $\delta, \hat{\delta}, \theta, \hat{\theta}, \omega_{\alpha_i}, \omega_{\beta_i} \in \mathbb{Z}_p$ are obtained from representation vectors $[\vec{\alpha}], [\vec{\beta}]$ in $\mathbb{Z}_p$ and elements $h_{ID_i}, h_{ID^*}, h_{m_i}, h_{ID^*}$ in $\mathbb{Z}_p^*$. We defer the definition in the full proof later. By using Eq. (1) and (2), we derive a new modular equation

$$x(h_{ID^*} + h_{m^*}\delta - \theta) + \sum_i^{q_s} r_i \left(h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i} \right) = \hat{\theta} - h_{m^*}\hat{\delta}.$$

Because the algebraic adversary is adaptive, it is necessary to analyze all the potential outcomes of their forgeries and representations. Figure 1 provides a diagrammatic overview of how different cases are classified, and how either (x, r_i) or $(h_{ID_i}, h_{m_i}, h_{ID^*}, h_{m^*})$ can be extracted in $\mathbb{Z}_p^*$ through simulation $\mathcal{R}_1$ or $\mathcal{R}_2$. Suppose adversary behaviours $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \overline{\mathbf{A}}, \overline{\mathbf{B}}, \overline{\mathbf{C}}, \overline{\mathbf{D}}$, as described in Fig. 1, are classified into condition $\mathcal{F}_1, \mathcal{F}_2$, such that $\mathcal{F}_1 : \mathbf{A} \vee \mathbf{B}$ and $\mathcal{F}_2 : (\overline{\mathbf{A}} \wedge \overline{\mathbf{B}}) \wedge (\mathbf{C} \vee \mathbf{D} \vee (\overline{\mathbf{C}} \wedge \overline{\mathbf{D}}))$. We see that all behaviours are captured, i.e. $\mathcal{F}_1 \vee \mathcal{F}_2 = 1$.

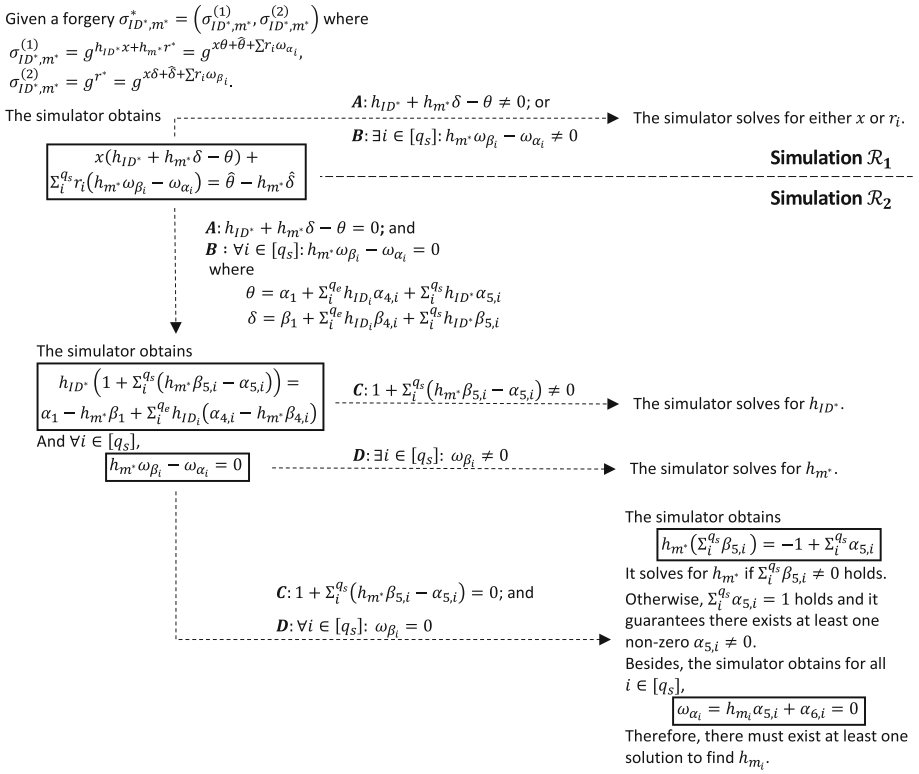

Fig. 1. A diagrammatic representation of the security proof in the AGM

Based on the definition of simulations $\mathcal{R}_1, \mathcal{R}_2$ as defined in Table 1, we see every element $x, r_i, h_{ID_i}, h_{m_i}, h_{ID^*}, h_{m^*}$ has been implicitly embedded with

unknown DL problem solution a, and known randomness $s_i, t_i, u_{1,i}, v_{1,i}, u_{2,i}, v_{2,i}$, $u_1^*, v_1^*, u_2^*, v_2^* \in \mathbb{Z}_p^*$. It is easy to see they are perfectly hidden from the view of the adversary, such that

$$\begin{aligned} x &= a, & r_i &= as_i + t_i, \\ h_{ID_i} &= au_{1,i} + v_{1,i}, & h_{m_i} &= au_{2,i} + v_{2,i}, \\ h_{ID^*} &= au_1^* + v_1^*, & h_{m^*} &= au_2^* + v_2^*. \end{aligned}$$

Therefore, the simulator manages to extract DL solution a as long as it obtains any of them $x, r_i, h_{ID_i}, h_{m_i}, h_{ID^*}, h_{m^*}$ in $\mathbb{Z}_p^*$. The full proof is detailed as follows.

Theorem 1. *Suppose hash functions $\mathcal{H}_1, \mathcal{H}_2$ are random oracles. If the DL problem is hard, the BLS-IBS scheme is secure against the EUF-CMA in the AGM with random oracles with a loss factor of* 2.

Proof. Suppose there exists an algebraic adversary who can (t, q_e, q_s, ϵ)-break the BLS-IBS scheme in the EUF-CMA model. We construct a simulator to solve the DL problem. Given as input a problem instance (g, g^a) over the pairing group tuple $\mathbb{PG} = (p, g, e, \mathbb{G}, \mathbb{G}_T)$, the simulator controls the random oracles $\mathcal{H}_1, \mathcal{H}_2$ and runs the algebraic adversary in a randomly chosen simulation $\mathcal{R}_1$ or $\mathcal{R}_2$ under condition $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively.

Simulation $\mathcal{R}_1$: Suppose the algebraic adversary returns forgery on (ID^*, m^*) under condition $\mathcal{F}_1$, given the DL problem instance g^a, the simulation is programmed by embedding g^a to master public key X and every queried signature element $\sigma^{(2)}_{ID^*,m_i} = g^{r_i}$ as follows.

Setup. The simulator sets $X = g^a$. The master public key is returned $mpk = (g, p, e, \mathbb{G}, \mathbb{G}_T, \mathcal{H}_1, \mathcal{H}_2, X)$.

Hash Query. At the beginning, the simulator prepares two empty sets L_{H1}, L_{H2} to record all hash queries and responses as follows.

$\mathcal{H}_1(ID_k)$: On an i-th random oracle query (ID_i). If $\langle ID_i, H_{ID_i}, h_{ID_i} \rangle \in L_{H1}$, the simulator responds to this query following the record. Otherwise, the simulator randomly selects $h_{ID_i} \in \mathbb{Z}_p$ and sets

$$H_{ID_i} = g^{h_{ID_i}}.$$

It programs $\mathcal{H}_1(ID_i) = H_{ID_i}$ and stores $\langle ID_k, H_{ID_i}, h_{ID_i} \rangle$ into L_{H1}. The simulator returns $\mathcal{H}_1(ID_i) = H_{ID_i}$.

$\mathcal{H}_2(m_i)$: On an i-th random oracle query (m_i). If $\langle m_i, H_{m_i}, h_{m_i} \rangle \in L_{H2}$, the simulator responds to this query following the record. Otherwise, the simulator randomly selects $h_{m_i} \in \mathbb{Z}_p$ and sets

$$H_{m_i} = g^{h_{m_i}}.$$

It programs $\mathcal{H}_2(m_i) = H_{m_i}$ and stores $\langle m_i, H_{m_i}, h_{m_i} \rangle$ into L_{H2}. The simulator returns $\mathcal{H}_2(m_i) = H_{m_i}$.

Query. The adversary makes extraction queries and signing queries in this phase. The simulator prepares an empty set L_E to record all queries and responses as follows.

$\mathcal{O}_E(ID_i)$: 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 user private key d_{ID_i} which is based the definition as follows. Otherwise, it calls $\mathcal{H}_1(ID_i) \to H_{ID_i} = g^{h_{ID_i}}$, retrieves h_{ID_i} from L_{H1}, and sets

$$d_{ID_i} = (g^a)^{h_{ID_i}}.$$

The simulator stores $\langle ID_i, d_{ID_i}\rangle$ into L_E. The user private key d_{ID_i} is returned.

Correctness. It is easy to see d_{ID_i} is a valid user private key, such that

$$d_{ID_i} = (g^a)^{h_{ID_i}} = \mathcal{H}_1(ID_i)^x.$$

$\mathcal{O}_S(ID_i, m_i)$: 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 $\mathcal{O}_E(ID_i) \to d_{ID_i}$ to generate fresh user private key if it does not exist. Otherwise, it retrieves d_{ID_i} from L_E. The simulator calls $\mathcal{H}_2(m_i) \to H_{m_i} = g^{h_{m_i}}$ and retrieves h_{m_i} from L_{H2}, randomly chooses $s_i, t_i \in \mathbb{Z}_p$, and sets signature $\sigma_{ID_i,m_i} = (\sigma^{(1)}_{ID_i,m_i}, \sigma^{(2)}_{ID_i,m_i})$ as follows. Let $r_i = as_i + t_i$,

$$\begin{aligned}\sigma^{(1)}_{ID_i,m_i} &= g^{a(h_{ID_i}+s_ih_{m_i})}g^{t_ih_{m_i}},\\ \sigma^{(2)}_{ID_i,m_i} &= g^{r_i} = g^{as_i+t_i}.\end{aligned}$$

Correctness. We see σ_{ID_i,m_i} is a valid signature on (ID_i, m_i), such that

$$\begin{aligned}\sigma^{(1)}_{ID_i,m_i} &= g^{a(h_{ID_i}+s_ih_{m_i})}g^{t_ih_{m_i}}\\ &= g^{ah_{ID_i}}g^{h_{m_i}(as_i+t_i)}\\ &= d_{ID_i}\cdot\mathcal{H}_2(m_i)^{r_i}.\end{aligned}$$

Forgery. Assume the forgery is returned under condition $\mathcal{F}_1$. The algebraic adversary returns a forged signature $\sigma_{ID^*,m^*} = (\sigma^{(1)}_{ID^*,m^*}, \sigma^{(2)}_{ID^*,m^*})$ on a chosen identity and message pair (ID^*, m^*) following the definition, such that for some $r^* \in \mathbb{Z}_p$,

$$\begin{aligned}\sigma^{(1)}_{ID^*,m^*} &= (H^*_1)^x(H^*_2)^{r^*}\\ &= g^{h_{ID^*}x+h_{m^*}r^*},\\ \sigma^{(2)}_{ID^*,m^*} &= g^{r^*}.\end{aligned}$$

It also attaches together with some representation vectors $[\vec{\alpha}], [\vec{\beta}]$ in $\mathbb{Z}_p$, such that

$$\begin{aligned}[\vec{\alpha}] = (&\alpha_0, \alpha_1, \alpha_{2,1}, \cdots, \alpha_{2,q_{h1}}, \alpha_{3,1}, \cdots, \alpha_{3,q_{h2}}, \alpha_{4,1},\\ &\cdots, \alpha_{4,q_e}, q_{5,1}, \cdots, \alpha_{5,q_s}, \alpha_{6,1}, \cdots, \alpha_{6,q_s}),\\ [\vec{\beta}] = (&\beta_0, \beta_1, \beta_{2,1}, \cdots, \beta_{2,q_{h1}}, \beta_{3,1}, \cdots, \beta_{3,q_{h2}}, \beta_{4,1},\\ &\cdots, \beta_{4,q_e}, q_{5,1}, \cdots, \beta_{5,q_s}, \beta_{6,1}, \cdots, \beta_{6,q_s}),\end{aligned}$$

which indicates how the forgery being algebraically computed, i.e.,

$$\begin{aligned}\sigma^{(1)}_{ID^*,m^*} =& g^{\alpha_0} X^{\alpha_1} \prod_i^{q_{h1}} (H_{ID_i})^{\alpha_{2,i}} \prod_i^{q_{h2}} (H_{m_i})^{\alpha_{3,i}}\\ &\prod_i^{q_e} (d_{ID_i})^{\alpha_{4,i}} \prod_i^{q_s} (\sigma^{(1)}_{ID^*,m_i})^{\alpha_{5,i}} (\sigma^{(2)}_{ID^*,m_i})^{\alpha_{6,i}},\\ \sigma^{(2)}_{ID^*,m^*} =& g^{\beta_0} X^{\beta_1} \prod_i^{q_{h1}} (H_{ID_i})^{\beta_{2,i}} \prod_i^{q_{h2}} (H_{m_i})^{\beta_{3,i}}\\ &\prod_i^{q_e} (d_{ID_i})^{\beta_{4,i}} \prod_i^{q_s} (\sigma^{(1)}_{ID^*,m_i})^{\beta_{5,i}} (\sigma^{(2)}_{ID^*,m_i})^{\beta_{6,i}}.\end{aligned}$$

Note that we omit signature query for any $\mathcal{O}_S(ID_i, \cdot)$ as the adversary may obtain $d_{ID_i} \leftarrow \mathcal{O}_E(ID_i)$ from the extraction oracle and sign the signature by itself. In order to simplify the above definition, let $\omega_{\alpha_i} = h_{m_i}\alpha_{5,i} + \alpha_{6,i}$, and $\omega_{\beta_i} = h_{m_i}\beta_{5,i} + \beta_{6,i}$,

$$\theta = \alpha_1 + \sum_i^{q_e} h_{ID_i}\alpha_{4,i} + \sum_i^{q_s} h_{ID^*}\alpha_{5,i}, \quad \hat{\theta} = \alpha_0 + \sum_i^{q_{h1}} h_{ID_i}\alpha_{2,i} + \sum_i^{q_{h2}} h_{m_i}\alpha_{3,i},$$

$$\delta = \beta_1 + \sum_i^{q_e} h_{ID_i}\beta_{4,i} + \sum_i^{q_s} h_{ID^*}\beta_{5,i}, \quad \hat{\delta} = \beta_0 + \sum_i^{q_{h1}} h_{ID_i}\beta_{2,i} + \sum_i^{q_{h2}} h_{m_i}\beta_{3,i}.$$

Based on above definition, it is easy to see $\sigma_{ID^*,m^*} = (\sigma^{(1)}_{ID^*,m^*}, \sigma^{(2)}_{ID^*,m^*})$ can be simply written as follows, such that

$$\sigma^{(1)}_{ID^*,m^*} = g^{x\theta+\hat{\theta}+\sum_i^{q_s} r_i(\omega_{\alpha_i})}, \quad \sigma^{(2)}_{ID^*,m^*} = g^{x\delta+\hat{\delta}+\sum_i^{q_s} r_i(\omega_{\beta_i})},$$

which yields the following two modular equations

$$\begin{aligned}h_{ID^*}x + h_{m^*}r^* &= x\theta + \hat{\theta} + \sum_i^{q_s} r_i\omega_{\alpha_i},\\ r^* &= x\delta + \hat{\delta} + \sum_i^{q_s} r_i\omega_{\beta_i}.\end{aligned}$$

Therefore, given the forgery and representation vectors, the simulator obtains the following modular equation by substituting r^*, such that

$$x(h_{ID^*} + h_{m^*}\delta - \theta) + \sum_{i}^{q_s} r_i(h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i}) = \hat{\theta} - h_{m^*}\hat{\delta}$$

Abort. The simulator aborts if $\overline{\mathbf{A}} : h_{ID^*} + h_{m^*}\delta - \theta = 0$ and $\overline{\mathbf{B}} : \forall i \in [q_s] : h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i} = 0$ hold.

Otherwise, based on above simulation definitions, where $x = a$ and $\forall i \in [q_s] : r_i = as_i + t_i$, the simulator can solve for a under condition $\mathcal{F}_1 : \mathbf{A} \vee \mathbf{B}$, such that behaviour $\mathbf{A} : h_{ID^*} + h_{m^*}\delta - \theta \neq 0$ or $\mathbf{B} : \exists i \in [q_s] : h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i} \neq 0$ holds. We can find a by computing

$$a = \frac{\hat{\theta} - h_{m^*}\hat{\delta} + \sum_i^{q_s} t_i(\omega_{\alpha_i} - h_{m^*}\omega_{\beta_i})}{h_{ID^*} + h_{m^*}\delta - \theta + \sum_i^{q_s} s_i(h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i})}.$$

Success Probability of $\mathcal{R}_1$. There is no abort in the simulation as the simulator manages to respond to every extraction and signing query. By condition $\mathcal{F}_1$, the simulator must obtain reducible forgery and representation as either $\mathbf{A} : h_{ID^*} + h_{m^*}\delta - \theta \neq 0$ or $\mathbf{B} : \sum_i^{q_s} s_i(h_{m^*}\omega_{\beta_i} - \omega_{\alpha_i}) \neq 0$ must hold. In particular, the success probability $\Pr[\text{Success}|\mathcal{R}_1]$ to extract the DL solution in $\mathcal{R}_1$ is based condition $\mathcal{F}_1$, hence $\Pr[\text{Success}|\mathcal{R}_1]$ is described as follows.

$$\begin{aligned}\Pr[\text{Success}|\mathcal{R}_1] &= \Pr[\text{Success}|\mathcal{F}_1] \\ &= \Pr[\mathbf{A} \vee \mathbf{B}] \\ &= 1 - (\Pr[\overline{\mathbf{A}}] \wedge \Pr[\overline{\mathbf{B}}]) \\ &= \frac{3}{4}\end{aligned}$$

Simulation $\mathcal{R}_2$: Suppose the algebraic adversary returns forgery on (ID^*, m^*) under condition $\mathcal{F}_2$, the simulation is programmed by embedding the DL problem instance g^a to each hash query H_{ID_i}, H_{m_i}, including H_{ID^*}, H_{m^*}.

Setup. The simulator randomly chooses $x \in \mathbb{Z}_p$ and sets $X = g^x$. The master public key is returned $mpk = (g, p, e, \mathbb{G}, \mathbb{G}_T, \mathcal{H}_1, \mathcal{H}_2, X)$.

Hash Query. The simulator prepares two empty sets L_{H1}, L_{H2} to record all queries and responses.

$\mathcal{H}_1(ID_i)$: On an i-th random oracle query (ID_i). If $\langle ID_i, H_{ID_i}, u_{1,i}, v_{1,i}\rangle \in L_{H1}$, the simulator responds to this query following the record. Otherwise, the simulator randomly selects $u_{1,i}, v_{1,i} \in \mathbb{Z}_p$ and sets

$$H_{ID_i} = g^{au_{1,i}+v_{1,i}}.$$

It programs $\mathcal{H}_1(ID_i) = H_{ID_i}$ and stores $\langle ID_i, H_{ID_i}, u_{1,i}, v_{1,i}\rangle$ into L_{H1}. The simulator returns $\mathcal{H}_1(ID_i) = H_{ID_i}$.

$\mathcal{H}_2(m_i)$: On an i-th random oracle query (m_i). If $\langle m_i, H_{m_i}, u_{2,i}, v_{2,i}\rangle \in L_{H2}$, the simulator responds to this query following the record. Otherwise, the simulator randomly selects $u_{2,i}, v_{2,i} \in \mathbb{Z}_p$ and sets

$$H_{m_i} = g^{au_{2,i}+v_{2,i}}.$$

It programs $\mathcal{H}_2(m_i) = H_{m_i}$ and stores $\langle m_i, H_{m_i}, u_{2,i}, v_{2,i}\rangle$ into L_{H2}. The simulator returns $\mathcal{H}_2(m_i) = H_{m_i}$.

Query. Similar to the definition of $\mathcal{R}_1$, the adversary makes extraction queries and signing queries in this phase. The simulator prepares an empty set L_E to record all queries and responses as follows.

$\mathcal{O}_E(ID_i)$: On an i-th extraction query (ID_i), the simulator checks whether $\langle ID_i, d_{ID_i}\rangle \in L_E$ exists. If there is, the simulator retrieves user private key d_{ID_i} which is based the definition as follows. Otherwise, it calls $\mathcal{H}_1(ID_i) \rightarrow H_{ID_i} = g^{au_{1,i}+v_{1,i}}$ and retrieves $u_{1,i}, v_{1,i} \in \mathbb{Z}_p$ from L_{H1}, and sets $d_{ID_i} = g^{au_{1,i}x+v_{1,i}x}$. The simulator stores $\langle ID_i, d_{ID_i}\rangle$ into L_E. The user private key d_{ID_i} is returned.

Correctness. It is easy to see d_{ID_i} is a valid user private key, such that

$$\begin{aligned} d_{ID_i} &= g^{au_{1,i}x+v_{1,i}x} \\ &= \mathcal{H}_1(ID_i)^x \end{aligned}$$

$\mathcal{O}_S(ID_i, m_i)$: 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 fresh user private key if it does not exist. Otherwise, it retrieves d_{ID_i}. The simulator calls $\mathcal{H}_2(m_i) = H_{m_i} = g^{au_{2,i}+v_{2,i}}$ and retrieves $u_{2,i}, v_{2,i}$ from L_{H2}, randomly chooses $r_i \in \mathbb{Z}_p$, and sets signature $\sigma_{ID_i,m_i} = (\sigma^{(1)}_{ID_i,m_i}, \sigma^{(2)}_{ID_i,m_i})$ as follows

$$\begin{aligned} \sigma^{(1)}_{ID_i,m_i} &= g^{a(xu_{1,i}+r_iu_{2,i})}g^{xv_11,i+r_iv_{2,i}}, \\ \sigma^{(2)}_{ID_i,m_i} &= g^{r_i}. \end{aligned}$$

Correctness. It is easy to verify σ_{ID_i,m_i} is a valid signature, such that

$$\begin{aligned} \sigma^{(1)}_{ID_i,m_i} &= g^{a(xu_{1,i}+r_iu_{2,i})}g^{xv_{1,i}+r_iv_{2,i}} \\ &= g^{x(au_{1,i}+v_{1,i})}g^{r_i(au_{2,i}+v_{2,i})} \\ &= d_{ID_i} \cdot \mathcal{H}_2(m_i)^{r_i} \end{aligned}$$

Forgery. Assume the forgery is returned under condition $\mathcal{F}_2$. The algebraic adversary returns a forged signature $\sigma_{ID^*,m^*} = (\sigma^{(1)}_{ID^*,m^*}, \sigma^{(2)}_{ID^*,m^*})$ on a chosen identity and message pair (ID^*, m^*) along with the representation vectors $[\vec{\alpha}], [\vec{\beta}]$ in $\mathbb{Z}_p$ following the definition as in $\mathcal{R}_1$. Let

$$g^{\theta} = g^{\alpha_1}\prod_i^{q_e}(H_{ID_i})^{\alpha_{4,i}}\prod_i^{q_s}(H_{ID^*})^{\alpha_{5,i}}, \quad g^{\delta} = g^{\beta_1}\prod_i^{q_e}(H_{ID_i})^{\beta_{4,i}}\prod_i^{q_s}(H_{ID^*})^{\beta_{5,i}},$$

$$g^{\omega_{\alpha_i}} = (H_{m_i})^{\alpha_{5,i}}g^{\alpha_{6,i}}, \qquad g^{\omega_{\beta_i}} = (H_{m_i})^{\beta_{5,i}}g^{\beta_{6,i}}.$$

Abort. The simulator aborts if $e(H_{ID^*}, g)\ e(H_{m^*}, g^{\delta}) \neq e(g, g^{\theta})$ holds and there exist some $i \in [q_s]$, such that $e(H_{m^*}, g^{\omega_{\beta_i}}) \neq e(g, g^{\omega_{\alpha_i}})$.

Based on above simulation definitions, where $\forall i \in [q_{h1}] : h_{ID_i} = au_{1,i} + v_{1,i}$ and $\forall i \in [q_{h2}] : h_{m_i} = au_{2,i} + v_{2,i}$, the simuator can solve for a under condition $\mathcal{F}_2 : (\overline{\mathbf{A}} \wedge \overline{\mathbf{B}}) \wedge (\mathbf{C} \vee \mathbf{D} \vee (\overline{\mathbf{C}} \wedge \overline{\mathbf{D}}))$ in the following three cases. It is worth noting that due to behaviour $\overline{\mathbf{A}} \wedge \overline{\mathbf{B}}$, the simulator must obtains following two equations

$$\overline{\mathbf{A}} : h_{ID^*} + h_{m^*}\delta = \theta \tag{3}$$

$$\overline{\mathbf{B}} : \forall i \in [q_s] : h_{m^*}\omega_{\beta_i} = \omega_{\alpha_i} \tag{4}$$

The simulator runs **Case 1** if $1 \neq g \cdot \prod_i^{q_s}((H_{m^*})^{\beta_{5,i}} \cdot g^{-\alpha_{5,i}})$ holds.
Case 1: $(\overline{\mathbf{A}} \wedge \overline{\mathbf{B}}) \wedge \mathbf{C})$. The simulator obtains Eq. (3): $h_{ID^*} + h_{m^*}\delta = \theta$ where $\theta = \alpha_1 + \sum_i^{q_e} h_{ID_i}\alpha_{4,i} + \sum_i^{q_s} h_{ID^*}\alpha_{5,i}$ and $\delta = \beta_1 + \sum_i^{q_e} h_{ID_i}\beta_{4,i} + \sum_i^{q_s} h_{ID^*}\beta_{5,i}$, which yields

$$\begin{aligned} &h_{ID^*}\,(1 + \Sigma_i^{q_s}(h_{m^*}\beta_{5,i} - \alpha_{5,i})) = \\ &\alpha_1 - h_{m^*}\beta_1 + \Sigma_i^{q_e} h_{ID_i}(\alpha_{4,i} - h_{m^*}\beta_{4,i}). \end{aligned}$$

Based on the definition of simulation $\mathcal{R}_2$, the simulator checks if $\exists i \in [q_e] : \beta_{4,i} \neq 0$ or $\Sigma_i^{q_s}\beta_{5,i} \neq 0$ hold. The simulator manages to derive the following modular quadratic equation

$$a^2\Delta + a\Delta' + \Delta'' = 0,$$

such that $\Delta, \Delta', \Delta''$ in $\mathbb{Z}_p$ can be defined as follows, which are computed based on all chosen randomness $u_{1,i}, v_{1,i}, u_{2,i}, v_{2,i}, u_1^*, v_1^*, u_2^*, v_2^* \in \mathbb{Z}_p^*$ by the simulator and received representation vectors $[\vec{\alpha}], [\vec{\beta}]$ in $\mathbb{Z}_p$,

$$\begin{aligned} \Delta &= \Sigma_i^{q_e} u_2^* u_{1,i}\beta_{4,i} + \Sigma_i^{q_s} u_2^* u_1^* \beta_{5,i}, \\ \Delta' &= u_1^* + u_2^*\beta_1 + \Sigma_i^{q_e}(u_2^* v_{1,i}\beta_{4,i} + v_2^* u_{1,i}\beta_{4,i} - u_{1,i}\alpha_{4,i}) \\ &\quad + \Sigma_i^{q_s}(u_2^* v_1^*\beta_{5,i} + v_2^* u_1^*\beta_{5,i} - u_1^*\alpha_{5,i}), \\ \Delta'' &= v_1^* + v_2^*\beta_1 - \alpha_1 + \Sigma_i^{q_e} v_{1,i}(v_2^*\beta_{4,i} - \alpha_{4,i}) \\ &\quad + \Sigma_i^{q_s} v_1^*(v_2^*\beta_{5,i} - \alpha_{5,i}). \end{aligned}$$

The simulator can find at most two solutions a_0 or a_1 by solving the above modular quadratic equation. It can test for the correct solution via the given DL problem instance $g^a = g^{a_0}$ or $g^a = g^{a_1}$.
Suppose the simulator obtains $\forall i \in [q_e] : \beta_{4,i} = 0$ and $\Sigma_i^{q_s}\beta_{5,i} = 0$ which implicitly sets $\Delta = 0$. In this case, the simulator solves for a by computing

$$a = -\frac{v_1^* + v_2^*\beta_1 - \alpha_1 + \Sigma_i^{q_e} v_{1,i}(-\alpha_{4,i}) + \Sigma_i^{q_s} v_1^*(-\alpha_{5,i})}{u_1^* + u_2^*\beta_1 + \Sigma_i^{q_e}(-u_{1,i}\alpha_{4,i}) + \Sigma_i^{q_s}(-u_1^*\alpha_{5,i})}.$$

Due to behaviour $\mathbf{C} : 1 + \Sigma_i^{q_s}(h_{m^*}\beta_{5,i} - \alpha_{5,i}) \neq 0$ holds, even though $\Sigma_i^{q_s}\beta_{5,i} = 0$ is set, it ensures that $\Sigma_i^{q_s}\alpha_{5,i} \neq 1$. While value u_1^* remains perfectly hidden due to random oracles, such that $h_{ID^*} = au_1^* + v_1^*$ is implicitly set, the adversary has no advantage to reproduce u_1^* if $\Sigma_i^{q_s}\alpha_{5,i} \neq 1$ holds. Therefore, the simulator must be able to find a as the success probability of setting $u_1^* + u_2^*\beta_1 + \Sigma_i^{q_e}(-u_{1,i}\alpha_{4,i}) + \Sigma_i^{q_s}(-u_1^*\alpha_{5,i}) = 0$ is negligible.
Otherwise, the simulator runs **Case 2** if there exists at least one i in $[q_s]$ that $H_{m_i}^{\beta_{5,i}} \cdot g^{\beta_{6,i}} \neq 1$.
Case 2: $(\overline{\mathbf{A}} \wedge \overline{\mathbf{B}}) \wedge \mathbf{D}$). The simulator retrieves Eq. (4):$\forall i \in [q_s] : h_{m^*}\omega_{\beta_i} = \omega_{\alpha_i}$, where $\omega_{\alpha_i} = h_{m_i}\alpha_{5,i} + \alpha_{6,i}$ and $\omega_{\beta_i} = h_{m_i}\beta_{5,i} + \beta_{6,i}$. Due to the definition of simulation $\mathcal{R}_2$, if $\exists i \in [q_s]\beta_{5,i} \neq 0$ holds, the simulator must obtain at least one modular quadratic equation

$$a^2\lambda + a\lambda' + \lambda'' = 0,$$

where $\lambda, \lambda', \lambda''$ in $\mathbb{Z}_p$ are computable by the simulator, such as

$$\begin{aligned}
\lambda &= u_2^* u_{2,i}\beta_{5,i} \\
\lambda' &= u_2^*(v_{2,i}\beta_{5,i} + \beta_{6,i}) + u_{2,i}(v_2^*\beta_{5,i} - \alpha_{5,i}) \\
\lambda'' &= v_2^*(v_{2,i}\beta_{5,i} + \beta_{6,i}) - v_{2,i}\alpha_{5,i} - \alpha_{6,i}.
\end{aligned}$$

The simulator can find at most two solutions a_0 or a_1 by solving the above modular quadratic equation. It can test for the correct solution via the given DL problem instance $g^a = g^{a_0}$ or $g^a = g^{a_1}$. Suppose the simulator obtains $\forall i \in [q_s] : \beta_{5,i} = 0$, which implicitly sets $\lambda = 0$. The simulator can solve for a by computing

$$a = -\frac{v_2^*(\beta_{6,i}) - v_{2,i}\alpha_{5,i} - \alpha_{6,i}}{u_2^*(\beta_{6,i}) + u_{2,i}(-\alpha_{5,i})}.$$

Due to behaviour $\mathbf{D} : \exists i \in [q_s] : \omega_{\beta_i} = h_{m_i}\beta_{5,i} + \beta_{6,i} \neq 0$ holds, even though $\forall i \in [q_s] : \beta_{5,i} = 0$ is set, it ensures that there exists at least one $\beta_{6,i} \neq 0$. Again, value u_2^* is perfectly hidden due to the random oracles, such that $h_{m^*} = au_2^* + v_2^*$ is implicitly set, the adversary has no advantage to reproduce u_2^*. Therefore, the simulator must be able to find a as the success probability to set $\forall i \in [q_s] : v_2^*(\beta_{6,i}) - v_{2,i}\alpha_{5,i} - \alpha_{6,i} = 0$ is negligible.
Otherwise, behaviours $\overline{\mathbf{C}} \wedge \overline{\mathbf{D}}$ must hold, and they yield following new equations

$$\overline{\mathbf{C}} : h_{m^*}\sum_i^{q_s}\beta_{5,i} = -1 + \sum_i^{q_s}\alpha_{5,i} \tag{5}$$

$$\overline{\mathbf{D}} : \forall i \in [q_s] : \omega_{\alpha_i} = h_{m_i}\alpha_{5,i} + \alpha_{6,i} = 0 \tag{6}$$

The simulator runs **Case 3**.

Case 3: $(\overline{\mathbf{A}} \wedge \overline{\mathbf{B}}) \wedge (\overline{\mathbf{C}} \wedge \overline{\mathbf{D}})$. The simulator checks if $\sum_i^{q_s} \beta_{5,i} \neq 0$ holds. Due to the definition of simulation $\mathcal{R}_2$, the simulator retrieves Eq. (5):$h_{m^*} \sum_i^{q_s} \beta_{5,i} = -1 + \sum_i^{q_s} \alpha_{5,i}$ and solves for a by computing

$$a = \frac{-1 + \sum_i^{q_s} (\alpha_{5,i} - v_2^* \beta_{5,i})}{u_2^* \sum_i^{q_s} \beta_{5,i}}.$$

Otherwise, $\sum_i^{q_s} \beta_{5,i} \neq 0$ yields that $\sum_i^{q_s} \alpha_{5,i} = 1$, which implies there exists at least one non-zero $\alpha_{5,i} \neq 0$ in $[q_s]$. Recall that the simulator still obtains Eq. (6):$\forall i \in [q_s] : \omega_{\alpha_i} = h_{m_i} \alpha_{5,i} + \alpha_{6,i} = 0$. Therefore, there is at least one solution to find a by computing

$$a = -\frac{\alpha_{6,i} + v_{2,i} \alpha_{5,i}}{u_{2,i} \alpha_{5,i}}.$$

Success Probability of $\mathcal{R}_2$. There is no abort in the simulation as the simulator manages to respond to every extraction and signing query. By condition $\mathcal{F}_2$ where the simulator obtains Eq. (3) and (4) due to behaviour $\overline{\mathbf{A}} \wedge \overline{\mathbf{B}}$, $1 + \sum_i^{q_s} (h_{m^*} \beta_{5,i} - \alpha_{5,i}) \neq 0$ due to behaviour $\mathbf{C}$, $\exists i \in [q_s] : \sum_i^{q_s} \beta_{5,i} \neq 0$ due to behaviour $\mathbf{D}$, and Eq. (5) and (6) due to behaviour $\overline{\mathbf{C}} \wedge \overline{\mathbf{D}}$, the simulator must obtain reducible forgery and representation as defined in either cases, which ensures solving for a with an overwhelming success probability. In particular, the success probability $\Pr[\text{Success}|\mathcal{R}_2]$ to extract the DL solution in $\mathcal{R}_2$ is based condition $\mathcal{F}_2$, hence $\Pr[\text{Success}|\mathcal{R}_2]$ is described as follows.

$$\begin{aligned} \Pr[\text{Success}|\mathcal{R}_2] &= \Pr[\text{Success}|\mathcal{F}_2] \\ &= \Pr[\textbf{Case 1}] + \Pr[\textbf{Case 2}] + \Pr[\textbf{Case 3}] \\ &= \frac{1}{4} \end{aligned}$$

Indistinguishable Simulations. According to both simulations $\mathcal{R}_1$, $\mathcal{R}_2$, the simulations are correct due to the correctness of every simulated user private key and signature holds. In addition, the simulator sets all elements with perfectly hidden randomness. For example, master public key $X = g^x$, hash responses $H_{ID_i} = g^{h_{ID_i}}, H_{m_i} = g^{h_{m_i}}$, signature randomnesses $\sigma^{(2)}_{ID_i,m_i} = g^{r_i}$. They are, respectively, simulated as

$$\begin{cases} a, & h_{ID_i}, & h_{m_i}, & as_i + t_i : & \mathcal{R}_1 \\ x, & au_{1,i} + v_{1,i}, & au_{2,i} + v_{2,i}, & r_i : & \mathcal{R}_2 \end{cases}$$

where $a, h_{ID_i}, h_{m_i}, s_i, t_i, x, u_{1,i}, v_{1,i}, u_{2,i}, v_{2,i}, r_i \in \mathbb{Z}_p^*$ are all randomly chosen; therefore, all elements look random and independent from the point of view of the adversary.

Reduction Loss. The concrete security is calculated based on the success probability of the two simulations. Since the two simulations are indistinguishable,

the simulator randomly selects $\mu \in \{0, 1\}$ and runs either one of the simulations. We describe the success probability of extracting the DL solution as follows.

$$\begin{aligned}\Pr[\text{Success}] &= \Pr[\text{Success}|\mathcal{R}_1]\Pr[\mu = 0] + \Pr[\text{Success}|\mathcal{R}_2]\Pr[\mu = 1] \\ &= \frac{3}{4}(\frac{1}{2}) + \frac{1}{4}(\frac{1}{2}) = \frac{1}{2}\end{aligned}$$

The given forgery and representations can be reducible in either one of the simulations under the condition $\mathcal{F}_1$ or $\mathcal{F}_2$ with overwhelming success probability. Therefore, the success probability to extract DL problem solution is at least $\frac{1}{2}$, which concludes that the security loss of the BLS-IBS scheme under EUF-CMA and DL assumption is 2.

This completes the proof of Theorem 1. □

5 Conclusion

In this work, we achieved a breakthrough by demonstrating tight security for IBS in the AGM. In particular, we established the security of the BLS-IBS scheme, which is derived from BLS signatures, under the DL assumption and EUF-CMA with random oracles in the AGM. Our security proof involves two simulations that can simulate any user private key and signature. Although the forgery and representations by the algebraic adversary were classified into several cases, we were able to capture them with either of the two simulations, resulting in a reduction loss of two. The research outcome of this paper provides valuable insight into achieving tight security for IBS in the AGM.

Further research could examine the possibility of a pairing-free IBS scheme that provides ideal security. The Schnorr-like IBS by Galindo and Garcia is currently the most efficient IBS without pairing, however, we identified that its security cannot be proven tight even in the AGM as it is unable to both respond to private key queries and extract the problem solution from any forgery in a simulation.

Acknowledgement. We would like to thank the anonymous reviewers for their constructive comments. Fuchun Guo was supported by ARC Future Fellowship (FT220100046).

References

1. Abe, M., Groth, J., Ohkubo, M.: Separating short structure-preserving signatures from non-interactive assumptions. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 628–646. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_34
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., et al. (eds.) EUROCRYPT 1988. LNCS, vol. 330, pp. 77–84. Springer, Heidelberg (1988). https://doi.org/10.1007/3-540-45961-8_7
8. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. J. Cryptol. **17**(4), 297–319 (2004)
9. Boneh, D., Venkatesan, R.: Breaking RSA may not be equivalent to factoring. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 59–71. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054117
10. Bresson, E., Monnerat, J., Vergnaud, D.: Separation results on the "One-More" computational problems. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 71–87. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79263-5_5
11. 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
12. Chin, J.J., Tan, S.Y., Heng, S.H., Phan, R.C.W.: Twin-schnorr: a security upgrade for the schnorr identity-based identification scheme. Sci. World J. 2015 (2015)
13. 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
14. Coron, J.-S.: Optimal security proofs for PSS and other signature schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_18
15. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)
16. 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
17. 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
18. 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)
19. 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)

20. 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
21. Garg, S., Bhaskar, R., Lokam, S.V.: Improved bounds on security reductions for discrete log based signatures. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 93–107. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_6
22. Goh, E.J., Jarecki, S., Katz, J., Wang, N.: Efficient signature schemes with tight reductions to the Diffie-Hellman problems. J. Cryptol. **20**(4), 493–514 (2007)
23. 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
24. 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)
25. 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
26. 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
27. Kurosawa, K., Heng, S.-H.: From digital signature to ID-based identification/signature. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 248–261. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24632-9_18
28. Maurer, U.M., Wolf, S.: The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms. SIAM J. Comput. **28**(5), 1689–1721 (1999)
29. Narayan, S., Parampalli, U.: Efficient identity-based signatures in the standard model. IET Inf. Secur. **2**(4), 108–118 (2008)
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. 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). https://doi.org/10.1007/3-540-48071-4_3
32. 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
33. Paterson, K.G.: Id-based signatures from pairings on elliptic curves. Electron. Lett. **38**(18), 1025–1026 (2002)
34. 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
35. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. J. Cryptol. **13**(3), 361–396 (2000)

36. 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
37. 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
38. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_7
39. Yi, P.: An efficient identity-based signature scheme with provable security. Inf. Sci. **576**, 790–799 (2021)
40. 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)