Research Article

Efficient and Provable Secure Pairing-Free Security-Mediated Identity-Based Identification Schemes

Ji-Jian Chin,¹ Syh-Yuan Tan,² Swee-Huay Heng,² and Raphael C.-W. Phan¹

¹ Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia

² Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia

Correspondence should be addressed to Ji-Jian Chin; jjchin@mmu.edu.my

Received 13 March 2014; Accepted 17 April 2014; Published 26 May 2014

Academic Editor: Mirjana Ivanovic

Copyright © 2014 Ji-Jian Chin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Security-mediated cryptography was first introduced by Boneh et al. in 2001. The main motivation behind security-mediated cryptography was the capability to allow instant revocation of a user's secret key by necessitating the cooperation of a security mediator in any given transaction. Subsequently in 2003, Boneh et al. showed how to convert a RSA-based security-mediated encryption scheme from a traditional public key setting to an identity-based one, where certificates would no longer be required. Following these two pioneering papers, other cryptographic primitives that utilize a security-mediated approach began to surface. However, the security-mediated identity-based identification scheme (SM-IBI) was not introduced until Chin et al. in 2013 with a scheme built on bilinear pairings. In this paper, we improve on the efficiency results for SM-IBI schemes by proposing two schemes that are pairing-free and are based on well-studied complexity assumptions: the RSA and discrete logarithm assumptions.

1. Introduction

1.1. Background. Identification schemes allow one party, the prover, to prove itself to another party, the verifier, that it knows its secret key without revealing anything else about itself in the process. The main utilization of the identification primitive is to facilitate one-sided entity authentication and is conventionally deployed in access control mechanisms to facilitate resource control and distribution.

In traditional public key cryptography, certificates are used to ensure that a user's public key is legitimately bound to a particular user and cannot be replaced. This certificate is usually issued by a certificate authority. However, certificate management can become an issue when the number of users in a cryptosystem grows larger. One of the methods of mitigating this potentially costly problem is through the deployment of the cryptographic primitive in an identity-based setting introduced by Shamir [1].

Another issue that traditional public key cryptography faces is the revocation of user secret keys. This would necessarily involve the revocation of a user's certificate along with his public/secret key pair and is a costly operation.

Timeliness is also a factor, as the procedure to revoke a user's keys may be a (relatively) long one and creates additional load on the certificate authority, turning the revocation procedure into a potentially costly process exercise. All these compound the certificate management issue mentioned earlier.

In the identity-based setting, where certificates are not used, key revocation is conventionally done by tagging validity periods onto the user's identity-string as an extension. This also creates an issue of timeliness since revocation is only possible at the end of those validity dates. Without checking certificates, it is also difficult to check if a user is still valid or if his user secret key has been revoked already.

In [2], Boneh et al. proposed the initial groundwork for instant revocation of user keys and privileges, including the public key and certificates by introducing the concept of security-mediated cryptography. The idea of security-mediated cryptography is to necessitate the cooperation of a security mediator, a trusted third party, in any form of transaction that a user needs to use. For example, in an encryption scheme, a security mediator needs to lend his cooperation to the user in order for the user to decrypt a particular ciphertext. And for signatures, a security mediator

has to agree to cooperate with a signer in order to produce a valid signature.

Specifically, this is done by separating the user's secret key into two portions during the key generation. One portion of the key is given to the mediator while the other is given to the user. Therefore, the security mediator cooperates in the form of providing his portion of the secret key to be combined with the user's portion to create the full secret key for the transaction.

1.2. Related Work. Identification schemes were first introduced by Fiat and Shamir [3], while their identity-based counterparts, namely, identity-based identification schemes, were first formalized by Bellare et al. [4] and Kurosawa and Heng [5] independently. In recent years, there have been various advances in the area of identity-based identification, such as the introduction of identity-based identification schemes in the standard model [6, 7], hierarchical model [8–10], and certificateless model [11].

Following Boneh et al.'s initial work, Ding and Tsudik expanded on security-mediated cryptography to cover identity-based encryption using the RSA assumption [12]. Shortly thereafter, Libert and Quisquater proposed the first pairing-based security-mediated identity-based encryption schemes [13]. On the signature front, Cheng et al. proposed the first security-mediated identity-based signature [14].

Chow et al. [15] and Yap et al. [16] both extended security-mediated cryptography into the certificateless setting, where the issue of key escrow was addressed. In certificateless cryptography, first proposed by Al-Riyami and Paterson [17], the key generation center only produces half of the user secret key. The user then produces the other half of the user secret key to be combined into the full user secret key, thus securing their key from even the key generation center. However, the cryptographic primitives in the certificateless setting introduce more operational cost to the scheme and are known to be difficult to be proven secure [18].

1.3. Motivations and Contribution. In 2013, Chin et al. first combined the notion of security-mediated cryptography with identifications in the identity-based setting to propose the first security-mediated identity-based identification (SM-IBI) scheme [19]. In the paper, the authors provided the first formal definitions for SM-IBIs and also proposed a concrete construction. The motivation of the authors was to allow fast revocation of keys for identification schemes in the identity-based setting.

For SM-IBI schemes, a security mediator is required to participate in the identification protocol in order for a user to authenticate himself to a verifier. The security mediator can then hold a revocation list to identify which users' secret keys have been revoked and refuse participation for those users.

However the first concrete scheme that was proposed by the authors was built based on bilinear pairings. The pairing operation is widely known by cryptographers to be a costly operation; thus, it would be beneficial to construct pairing-free alternatives to facilitate more efficient running of the cryptographic primitive.

In this paper, we propose two pairing-free SM-IBI constructs as faster alternatives to the pairing-based scheme proposed by Chin et al. Our two schemes are constructed based on the RSA assumption and the discrete logarithm assumption, respectively. The RSA-based scheme, which we name as the GQ-SM-IBI, is constructed based on the Guillou Quisquater identification scheme constructed by Bellare and Palacio [20]. On the other hand, the discrete logarithm-based scheme, which we name as the BNN-SM-IBI, is constructed based on the BNN identity-based identification scheme proposed by Bellare et al. [4]. We provide security analysis for both schemes, proving them secure against impersonation under passive attacks if the RSA assumption and the discrete logarithm assumption hold and secure against impersonation under active and concurrent attacks if the one-more RSA inversion assumption and the one-more discrete logarithm assumption hold. Lastly, we provide an efficiency analysis, both theoretically and practically, and show that both schemes are significantly faster than Chin et al.'s pairing-based SM-IBI scheme.

The rest of the paper is organized as follows. We begin with some preliminaries and review the formal definitions and security model of SM-IBI schemes in Section 2. Then we show the construction and security analysis for the GQ-SM-IBI scheme in Section 3. This is followed by the construction and security analysis for the BNN-SM-IBI scheme in Section 4. In Section 5 we show the operational costs of both schemes as well as presenting our implementation results. Finally we conclude in Section 6.

2. Preliminaries

2.1. Discrete Logarithm Assumption. Let G be a cyclic group with prime order q and let g be a generator of G . The DL problem (DL) is defined as given a number $A = g^a$ in group G , output a .

Definition 1. The discrete logarithm assumption states that there exists no polynomial-time algorithm M that is able to (t_{DL}, ϵ_{DL}) -solve the discrete logarithm problem with nonnegligible probability such that

$$\Pr [M(G, q, g, A) = a] \geq \epsilon_{DL}. \quad (1)$$

2.2. The One-More Discrete Logarithm Assumption. The one-more discrete logarithm (OMDL) problem was first introduced by Bellare and Palacio [20] in their proof against impersonation under active and concurrent attacks for the standard Schnorr identification scheme. Later work that involves proving security of identification schemes based on discrete logarithms to be secure against active and concurrent attacks for discrete logarithm also makes use of this assumption such as [4, 21].

Let G be a finite cyclic group of order q and let g be a generator of G . Define an experiment E_{DL} where an adversary M is given a challenge oracle $CHALL$ that produces a random group element $W_i \in G$ when queried and a discrete log oracle $DLOG$, which provides the discrete log $w_i \in \mathbb{Z}_q$ corresponding to the query W_i where $g^{w_i} = W_i$. M wins

if after making i queries to $CHALL_{DL}$, M is able to output solutions to all i challenges with only $i - 1$ queries to $DLOG$, meaning M has to solve at least one instance of the discrete logarithm problem without relying on the discrete log oracle. E_{DL} returns 1 if M is successful and 0 otherwise.

Definition 2. The *OMDL* assumption states that there exists no polynomial-time algorithm M that is able to $(t_{OMDL}, q_{OMDL}, \epsilon_{OMDL})$ -solve the *OMDL* problem with non-negligible probability where

$$\Pr [E_{DL} (M (1^k, G, g, CHALL_{DL}, DLOG) = 1)] \geq \epsilon_{OMDL}. \quad (2)$$

2.3. RSA Inversion (RSAI) Assumption. Given $(N, e, X) \xleftarrow{\$} k_{RSA}(1^k)$ compute Y such that $Y = X^d \bmod N$ where $ed = 1 \bmod \phi(N)$.

Definition 3. The *RSAI* assumption states that there exists no polynomial-time algorithm M that is able to $(t_{RSAI}, \epsilon_{RSAI})$ -solve the *RSAI* problem with nonnegligible probability such that

$$\Pr [M (1^k, N, e, X) = Y : Y = X^d \text{ and } ed = 1 \bmod \phi(N)] \geq \epsilon_{RSAI}. \quad (3)$$

2.4. One-More RSA Inversion (OMRSAI) Assumption. This is the interactive variant of the *RSAI* problem first proposed by Bellare and Palacio [20] to prove security of the GQ identification scheme and is analogous to the *OMDL* assumption. This assumption is applied in the proof of security against active and concurrent attacks for RSA-based schemes.

Define an experiment E_{RSAI} where an adversary M is given $(N, e) \xleftarrow{\$} k_{RSA}(1^k)$ as input and access to two oracles $CHALL$ and RSA . $CHALL$ on any input returns a random point W_i , while RSA on any input h will return h^d where $ed = 1 \bmod N$. M is required to compute the *RSAI* solutions to all the target points $W_0, \dots, W_n$ while using strictly less queries to the RSA oracle. In other words, M is required to find $W_0^d, \dots, W_n^d$ while using the RSA oracle only $i < n$ times. E_{RSAI} returns 1 if M is successful and 0 otherwise.

Definition 4. The *OMRSAI* assumption states that there exists no polynomial-time algorithm M that is able to $(t_{OMRSAI}, q_{OMRSAI}, \epsilon_{OMRSAI})$ -win the *OMRSAI* problem with nonnegligible probability where

$$\Pr [E_{RSAI} (M (1^k, N, e, CHALL_{RSA}, RSA)) = 1] \geq \epsilon_{RSAI}. \quad (4)$$

2.5. Definition of Security-Mediated IBI Schemes. In this section, we review the definition of SM-IBI schemes as defined by [19]. The definition follows closely to that of conventional IBI schemes, with the difference being that the prover segment is extended to encompass obtaining tokens from the security mediator. The SM-IBI scheme is defined as five probabilistic polynomial-time algorithms.

(i) *Setup*. It takes in the security parameter 1^k as input and outputs the system parameters $params$ along with the master secret key MSK .

(ii) *Extract*. Upon receiving a user's request for a key, it takes in $params$, MSK and a user's identity ID . Once the secret key is created, the PKG separates the key into two portions, one for the user, USK_{user} , and one for the security mediator, USK_{sem} , and returns the portions to the respective parties.

(iii) *Identification Protocol*. The identification protocol is an interactive protocol run by the 3 algorithms: *User-Prover* and *SEM-prover* on the prover side trying to authenticate himself to the *Verifier*. Both provers are used cooperatively in the interactive three-step canonical honest verifier zero knowledge proof of knowledge protocol with the verifier as follows.

(1) *User-Prover* initiates by sending his identity to ID to the *SEM-prover*. *SEM-prover* checks whether ID 's keys have been revoked and stops with an error code if true. If ID is legitimate then it generates and sends *SEM-COMMIT* to *User-Prover* who then combines *SEM-COMMIT* with his own *USER-COMMIT* to form *FULL-COMMIT* to send to *Verifier*.

(2) *Verifier* selects a random *CHALLENGE* and sends it to the *User-Prover*.

(3) *User-Prover* relays *CHALLENGE* to *SEM-Prover* and receives *SEM-RESPONSE* from *SEM-Prover* which it then combines with his own *USER-RESPONSE* to form *FULL-RESPONSE* and sends to the *Verifier*. The *Verifier* will choose to either accept or reject it.

2.6. Security Model for Security-Mediated IBI. Adversaries of SM-IBI follow the description of standard identification schemes: passive and active/concurrent attackers. However, the adversary for SM-IBI is able to query additionally partial conversation components, specifically the user's prover and the security mediator's prover besides the usual full prover query.

The security of SM-IBI schemes is modelled as a game played by an adversary I against a challenger C as follows.

(i) *Setup*. C runs *Setup*, creates the system parameters $params$ and passes them to I while keeping the master secret key MSK to itself.

(ii) *Phase 1*. This is the training phase. I is allowed to adaptively make the following queries to C .

(a) *User-Extract* (ID). C will run *Extract* but returns only the user's portion of the secret key to I .

(b) *SEM-Extract* (ID). C will run *Extract* but returns only the security mediator's portion of the secret key to I .

- (c) *Full-Extract* (ID). C will run *Extract* and returns both the user's portion of the secret key and the security mediator's portion of the secret key to I .
- (d) *Identification Queries* (ID). For passive adversaries, C will generate transcripts of valid conversations for I . For active/concurrent adversaries, C will act as the cheating prover, engaging I as the cheating verifier in conversations. I is able to issue any one of the following identification queries.
 - (1) *SEM-Identification* (ID). C runs the security mediator's half of the prover session.
 - (2) *User-Identification* (ID). C runs the user's half of the prover session.
 - (3) *Full-Identification* (ID). C combines both security mediator's and user's session to generate a full and valid conversation.
- (iii) *Phase 2*. I will eventually output ID^* on which it wants to be challenged on and begins its role as the cheating prover for both security mediator and user prover sessions. C on the other hand assumes the role of the verifier. I wins the game if it manages to convince C to accept with nonnegligible probability.

We say a security-mediated IBI scheme Π is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -secure under passive or active/concurrent attacks if for any passive or active/concurrent Type-1 impersonator I who runs in time t_{SMIBI} , $\Pr[I \text{ can impersonate}] < \epsilon_{SMIBI}$, where I can make at most q_{SMIBI} full extract queries.

It is interesting to point out that extracting the security mediator's half of the secret key gives no information about the full user key and that neither security mediator's prover sessions nor user's prover sessions done alone will provide a valid conversation, but only the combined session will. This models the security requirement that any user cannot legitimately prove himself to a verifier without the security mediator's help.

3. GQ-SMIBI: RSA-Based Security-Mediated IBI Scheme

The GQ-SM-IBI scheme is derived from the GQ identification scheme proposed in [20] and is provably secure against passive attackers assuming the RSAI assumption and against active/concurrent attackers assuming the OMRSI assumption.

The GQ-SM-IBI scheme is constructed as follows.

- (1) *Setup* (1^k). It takes in the security parameter 1^k , runs the key generation algorithm for RSA, and obtains an RSA instance, $(N, e, d) \xleftarrow{\$} k_{RSA}(1^k)$. It chooses a hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_N^*$ and publishes the system parameters $mpk = \langle N, e, H \rangle$. The master secret key $msk = d$ is kept secret.
- (2) *Extract* (mpk, msk, ID). It takes in $mpk, msk = s$, and ID as input. It calculates $H(ID)$ and sets

$X_{FULL} = H(ID)^d$. It further chooses $d_{SEM} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$, sets $d_{USER} = d - d_{SEM}$, and calculates $X_{USER} = H(ID)^{d_{USER}}$ and $X_{SEM} = H(ID)^{d_{SEM}}$. It gives X_{USER} to the user as its partial secret key and X_{SEM} to the security mediator as its partial private key and keeps X_{FULL} secret.

- (3) *Identification Protocol*. It is run by the User-Prover and SEM-Prover with Verifier as such.

- (a) *User-Prover* sends its ID to *Sem-Prover* and chooses $y_{USER} \xleftarrow{\$} \mathbb{Z}_N^*$ and sets $Y_{USER} = y_{USER}^e$. *SEM-Prover* upon receiving ID checks if it is still valid and returns error if it is revoked. Otherwise, *SEM-Prover* chooses $y_{SEM} \xleftarrow{\$} \mathbb{Z}_N^*$ and sends $Y_{SEM} = y_{SEM}^e$ to *User-Prover*. *User-Prover* combines $Y_{FULL} = Y_{SEM} \times Y_{USER}$ and sends Y_{FULL} to *Verifier*.
- (b) *Verifier* chooses a random challenge $c \xleftarrow{\$} \mathbb{Z}_{l(k)}^*$ where $l(\cdot)$ is a super-logarithmic challenge length such that $2^{l(k)} < e$. *Verifier* then sends it to *User-Prover*. *User-Prover* relays c to *SEM-Prover*.
- (c) *SEM-Prover* calculates its half of the response $z_{SEM} = y_{SEM} X_{SEM}^c$ and sends it to *User-Prover*. *User-Prover* combines it with his response $z_{USER} = y_{USER} X_{USER}^c$ to create $z_{FULL} = z_{SEM} \times z_{USER}$ and sends it as a response to *Verifier*.

Verifier checks if $z_{FULL}^e = Y_{FULL} H(ID)^c$ and accepts if yes; otherwise it outputs reject.

To check for completeness,

$$\begin{aligned}
 Z_{FULL}^e &= [(y_{SEM} X_{SEM}^c) (y_{USER} X_{USER}^c)]^e \\
 &= [y_{SEM} y_{USER} (X_{SEM} X_{USER})^c]^e \\
 &= y_{SEM}^e y_{USER}^e (H(ID)^{(d_{SEM} + d_{USER})ec}) \\
 &= Y_{SEM} Y_{USER} (H(ID)^{dec}) \\
 &= Y_{FULL} (H(ID)^c).
 \end{aligned} \tag{5}$$

3.1. Security Analysis: Impersonation under Passive Attack

Theorem 5. The GQ-SM-IBI scheme is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -secure against impersonation under passive attacks in the random oracle if the RSA Inversion Problem is $(t_{RSAI}, \epsilon_{RSAI})$ -hard where

$$\epsilon_{SMIBI} \leq \sqrt{\epsilon_{RSAI} e (q_{SMIBI} + 1)} + \frac{1}{2^{l(k)}}. \tag{6}$$

Proof. Assume the GQ-SM-IBI scheme is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -breakable; then a simulator M that $(t_{RSAI}, \epsilon_{RSAI})$ -breaks the RSAI problem can be shown. M takes in input (N, e, y) and runs the impersonator I as a subroutine.

Without loss of generality, it can be assumed that any *SEM-Extract*, *User-Extract*, *SEM-Identification*, *User-Identification*, and *Full-Identification* queries are preceded by a *Create-User* query. To avoid collision and consistently respond to these queries, M maintains a list $L_H = \langle ID_i, Q_{ID_i}, f_{ID_i}, X_{FULL_{ID_i}}, X_{SEM_{ID_i}}, X_{USER_{ID_i}}, d_{SEM_{ID_i}}, d_{USER_{ID_i}} \rangle$ which is initially empty. The following shows how M simulates the environment and oracle queries for I .

- (1) *Setup*. M selects a hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_N^*$ and sets the system parameters as $mpk = \langle N, e, H \rangle$. It sends mpk to I .
- (2) *Create-User* (ID_i). M chooses $l \in \{1, \dots, q_H\}$ randomly and lets $ID_l = ID^*$ at this point. Whenever I makes a H query on ID_i , consider the following.

- (a) If $i = l$, M will choose $f_{ID^*} \xleftarrow{\$} \mathbb{Z}_N^*$ sets $Q_{ID^*} = y f_{ID^*}^e$ and will return Q_{ID^*} to I . It adds $\langle ID_l, Q_{ID^*}, f_{ID^*}, X_{FULL_{ID^*}} = \perp, X_{USER_{ID^*}} = \perp, X_{SEM_{ID^*}} = \perp, d_{SEM_{ID^*}} = \perp, d_{USER_{ID^*}} = bot \rangle$ to L_H .
- (b) Otherwise, for $i \neq l$, M chooses $f_{ID_i} \xleftarrow{\$} \mathbb{Z}_q^*$, sets $Q_{ID_i} = f_{ID_i}^e$, and returns Q_{ID_i} to I . M also sets $X_{FULL_{ID_i}} = f_{ID_i}$, chooses $d_{USER_{ID^*}} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$, and sets $X_{SEM_{ID_i}} = Q_{ID_i}^{d_{SEM_{ID_i}}}$ as the SEM portion of the user private key and $X_{USER_{ID_i}} = f_{ID_i} Q_{ID_i}^{-d_{USER_{ID_i}}}$ as the user portion of the user private key. It can be seen that

$$\begin{aligned} X_{SEM_{ID_i}} \times X_{USER_{ID_i}} &= Q_{ID_i}^{d_{SEM_{ID_i}}} \times f_{ID_i} Q_{ID_i}^{-d_{SEM_{ID_i}}} \\ &= f_{ID_i} \\ &= X_{FULL_{ID_i}}. \end{aligned} \quad (7)$$

M adds $\langle ID_i, Q_{ID_i}, f_{ID_i}, X_{FULL_{ID_i}}, X_{USER_{ID_i}}, X_{SEM_{ID_i}}, d_{SEM_{ID_i}}, d_{USER_{ID_i}} \rangle$ to L_H . M returns Q_{ID_i} to I .

- (3) *SEM-Extract* (ID_i). If $ID_i = ID^*$ and *USER-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *USER-Extract* (ID^*) has not been queried before, M chooses $d_{SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$ and sets $X_{SEM_{ID^*}} = Q_{ID^*}^{d_{SEM_{ID^*}}}$ as the SEM portion of the user private key, saves it in L_H , and returns it to I . Otherwise for all other $ID_i \neq ID^*$, M just retrieves $X_{SEM_{ID_i}}$ in L_H and returns it to I .
- (4) *USER-Extract* (ID_i). If $ID_i = ID^*$ and *USER-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *SEM-Extract* (ID^*) has not been queried before, M chooses $d_{USER_{ID^*}} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$ and sets $X_{USER_{ID^*}} = Q_{ID^*}^{d_{USER_{ID^*}}}$ as the USER portion of the user

private key, saves it in L_H , and returns it to I . Otherwise for all other $ID_i \neq ID^*$, M just retrieves $X_{USER_{ID_i}}$ in L_H and returns it to I .

- (5) *Identification* (ID_j , *session*). I will act as the cheating verifier to learn information from valid conversation transcripts from M . If $ID_j \neq ID^*$, M just retrieves ID_j 's entry in L_H and runs the identification protocol for either the SEM's interactions, the user's interactions, or both combined as in the protocol.

Otherwise, for $ID_j = ID^*$, M then creates a full valid transcript for each m th query by picking $z_{m,USER_{ID^*}}, z_{m,SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q^*$, and $c_{m,ID^*} \xleftarrow{\$} \mathbb{Z}_q^*$ and sets $Z_{m,FULL_{ID^*}} = z_{m,USER_{ID^*}} \times z_{m,SEM_{ID^*}}$. M also sets $Y_{m,USER_{ID^*}} = z_{m,USER_{ID^*}}^e Q_{ID^*}^{-(c_{m,ID^*}/2)}$, $Y_{m,SEM_{ID^*}} = z_{m,SEM_{ID^*}}^e Q_{ID^*}^{-(c_{m,ID^*}/2)}$, and $Y_{m,FULL_{ID^*}} = z_{m,FULL_{ID^*}}^e Q_{ID^*}^{-c_{m,ID^*}}$. M tags the session with *session* = m . If I queries

- (i) *SEM-Identification* (ID^* , m), M will return $\langle Y_{m,SEM_{ID^*}}, c_{m,ID^*}, z_{m,SEM_{ID^*}} \rangle$;
- (ii) *User-Identification* (ID^* , m), M will return $\langle Y_{m,USER_{ID^*}}, c_{m,ID^*}, z_{m,USER_{ID^*}} \rangle$;
- (iii) *Full-Identification* (ID^* , m), M will return $\langle Y_{m,FULL_{ID^*}}, c_{m,ID^*}, z_{m,FULL_{ID^*}} \rangle$.

If no session is specified for the query, M just returns the next session in sequence. One can see that *SEM-Identification* (ID^* , m) and *User-Identification* (ID^* , m) can be combined to create a full and valid transcript on session m :

$$\begin{aligned} Y_{m,FULL_{ID^*}} H(ID^*)^{c_{m,ID^*}} &= Y_{m,SEM_{ID^*}} Y_{m,USER_{ID^*}} Q_{ID^*}^{c_{m,ID^*}} \\ &= z_{m,SEM_{ID^*}}^e Q_{ID^*}^{-(c_{m,ID^*}/2)} z_{m,USER_{ID^*}}^e Q_{ID^*}^{-(c_{m,ID^*}/2)} Q_{ID^*}^{c_{m,ID^*}} \\ &= [z_{m,SEM_{ID^*}} z_{m,USER_{ID^*}}]^e Q_{ID^*}^{-c_{m,ID^*}} Q_{ID^*}^{c_{m,ID^*}} \\ &= z_{m,SEM_{ID^*}}^e z_{m,USER_{ID^*}}^e \\ &= z_{m,FULL_{ID^*}}^e. \end{aligned} \quad (8)$$

Eventually, I stops Phase 1 and outputs the challenge ID , ID^* on which it wishes to be challenged on. M checks if $ID^* = ID_l$ from L_H and aborts if not. Otherwise, M runs I now as a cheating prover on ID^* by obtaining its commitment, $Y_{FULL_{ID^*}}$, selecting a challenge $c_1 \xleftarrow{\$} \mathbb{Z}_q^*$ and obtaining the response $z_{FULL_{ID^*},1}$ from I . M then resets I to the step whereby I just sent $Y_{FULL_{ID^*}}$, selects a second challenge $c_2 \xleftarrow{\$} \mathbb{Z}_q^*$, and receives $z_{FULL_{ID^*},2}$ as response. It should hold that $z_{FULL_{ID^*},1}^e = Y_{FULL_{ID^*}} H(ID)^{c_1}$ and $z_{FULL_{ID^*},2}^e = Y_{FULL_{ID^*}} H(ID)^{c_2}$. One can then obtain $(z_{FULL_{ID^*},1}^e / z_{FULL_{ID^*},2}^e)^e = H(ID)^{c_1 - c_2}$. Since e is a prime and

$-e < c_1 - c_2 < e$, $\text{GCD}(e, c_1 - c_2) = 1$. Use the extended Euclidean algorithm to obtain integers S and T such that $eS + (c_1 - c_2)T = 1$. It follows that

$$\begin{aligned}
 H(\text{ID}^*) &= Q_{\text{ID}^*}^{eS+(c_1-c_2)T} \bmod N \\
 &= (yf_{\text{ID}^*}^e)^{eS+(c_1-c_2)T} \bmod N \\
 &= (yf_{\text{ID}^*}^{e^2S})(yf_{\text{ID}^*}^{e(c_1-c_2)T}) \bmod N \\
 &= (yf_{\text{ID}^*}^{e^2S}) \left(\frac{z_{\text{FULL}_{\text{ID}^*,1}}}{z_{\text{FULL}_{\text{ID}^*,2}}} \right)^{eT} \bmod N \\
 &= \left[(yf_{\text{ID}^*}^{eS}) \left(\frac{z_{\text{FULL}_{\text{ID}^*,1}}}{z_{\text{FULL}_{\text{ID}^*,2}}} \right)^T \right]^e \bmod N.
 \end{aligned} \tag{9}$$

M then calculates the solution to the RSAI problem as follows:

$$\begin{aligned}
 yf_{\text{ID}^*}^e &= \left[(yf_{\text{ID}^*}^{eS}) \left(\frac{z_{\text{FULL}_{\text{ID}^*,1}}}{z_{\text{FULL}_{\text{ID}^*,2}}} \right)^T \right]^e \bmod N, \\
 y^d f_{\text{ID}^*} &= (yf_{\text{ID}^*}^{eS}) \left(\frac{z_{\text{FULL}_{\text{ID}^*,1}}}{z_{\text{FULL}_{\text{ID}^*,2}}} \right)^T \bmod N, \\
 y^d &= \frac{(yf_{\text{ID}^*}^{eS}) (z_{\text{FULL}_{\text{ID}^*,1}}/z_{\text{FULL}_{\text{ID}^*,2}})^T}{f_{\text{ID}^*}} \bmod N, \\
 y^d &= (yf_{\text{ID}^*}^{eS-1}) \left(\frac{z_{\text{FULL}_{\text{ID}^*,1}}}{z_{\text{FULL}_{\text{ID}^*,2}}} \right)^T \bmod N.
 \end{aligned} \tag{10}$$

It remains to calculate the probability of M solving the RSAI problem and winning the game. The probability of M successfully extracting two valid conversation transcripts from I is bounded by $(\epsilon_{\text{IBI}} - (1/2^{l(k)}))^2$ as given by the reset lemma [20]:

$$\begin{aligned}
 \Pr[M \text{ wins RSAI}] &= \Pr[M \text{ computes } g^{ab} \wedge \neg \text{abort}] \\
 &= \Pr[M \text{ computes } g^{ab} \mid \neg \text{abort}] \Pr[\neg \text{abort}] \\
 \epsilon_{\text{RSAI}} &\geq \left(\epsilon_{\text{SMIBI}} - \frac{1}{2^{l(k)}} \right)^2 \Pr[\neg \text{abort}].
 \end{aligned} \tag{11}$$

Finally calculate $\Pr[\neg \text{abort}]$. Let δ be the probability that I issues both a *SEM-Extract* and a *USER-Extract* query on ID^* and that I makes a total of q_{SMIBI} of such queries. The probability of M answering all the extraction queries is δ_{SMIBI}^q . In Phase 2, the probability of M not aborting is if I outputs the challenge identity ID^* that was not queried before. This is given by the probability $1 - \delta$. Compiling them, the probability of M not aborting is $\delta_{\text{SMIBI}}^q (1 - \delta)$. This probability is maximised at $\delta_{\text{opt}} = 1 - (1/(q_{\text{SMIBI}} + 1))$. Using δ_{opt} , the probability that M does not abort is at least $1/(2^{l(k)}(q_{\text{SMIBI}} + 1))$ because the value $(1 - (1/(q_{\text{SMIBI}} + 1)))^{q_{\text{SMIBI}}}$

approaches $1/e$ for large q_{SMIBI} . Therefore, the advantage of M , ϵ_{RSAI} , and the bound of the simulation are given as follows:

$$\begin{aligned}
 \epsilon_{\text{RSAI}} &\geq \left(\epsilon_{\text{SMIBI}} - \frac{1}{2^{l(k)}} \right)^2 \frac{1}{(q_{\text{SMIBI}} + 1)} \\
 \epsilon_{\text{RSAI}} (q_{\text{SMIBI}} + 1) &\geq \left(\epsilon_{\text{SMIBI}} - \frac{1}{2^{l(k)}} \right)^2 \\
 \epsilon_{\text{SMIBI}} &\leq \sqrt{\epsilon_{\text{RSAI}} (q_{\text{SMIBI}} + 1)} + \frac{1}{2^{l(k)}}.
 \end{aligned} \tag{12}$$

□

3.2. Security Analysis: Impersonation under Active/Concurrent Attack

Theorem 6. The GQ-SM-IBI scheme is $(t_{\text{SMIBI}}, q_{\text{SMIBI}}, \epsilon_{\text{SMIBI}})$ -secure against impersonation under active/concurrent attacks in the random oracle if the OMCDH Problem is $(t_{\text{OMRSAI}}, q_{\text{OMRSAI}}, \epsilon_{\text{OMRSAI}})$ -hard where

$$\epsilon_{\text{SMIBI}} \leq \sqrt{\epsilon_{\text{OMRSAI}} e (q_{\text{SMIBI}} + 1)} + \frac{1}{2^{l(k)}}. \tag{13}$$

Proof. Assume the GQ-SM-IBI scheme is $(t_{\text{SMIBI}}, q_{\text{SMIBI}}, \epsilon_{\text{SMIBI}})$ -breakable; then a simulator M that $(t_{\text{OMRSAI}}, q_{\text{OMRSAI}}, \epsilon_{\text{OMRSAI}})$ -breaks the OMRSAI problem can be shown. M takes in input (N, e) , is given access to *CHALL* and *RSA* oracles, and runs the impersonator I as a subroutine. Any *SEM-Extract*, *User-Extract*, *SEM-Identification*, *User-Identification*, and *Full-Identification* queries are preceded by a *Create-User* query. M maintains a list $L_H = \langle \text{ID}_i, Q_{\text{ID}_i}, f_{\text{ID}_i}, X_{\text{FULL}_{\text{ID}_i}}, X_{\text{SEM}_{\text{ID}_i}}, X_{\text{USER}_{\text{ID}_i}}, d_{\text{SEM}_{\text{ID}_i}}, d_{\text{USER}_{\text{ID}_i}} \rangle$ which is initially empty. The following shows how M simulates the environment and oracle queries for I .

- (1) *Create-User* (ID_i). M chooses $l \in \{1, \dots, q_H\}$ randomly and lets $\text{ID}_l = \text{ID}^*$ at this point. Whenever I makes a H query on ID_i , consider the following.

- (i) If $i = l$, M will choose $f_{\text{ID}^*} \xleftarrow{\$} \mathbb{Z}_N^*$, queries *CHALL* for w_0 , and sets $Q_{\text{ID}^*} = w_0 f_{\text{ID}^*}^e$ and will return Q_{ID^*} to I . It adds $\langle \text{ID}_i, Q_{\text{ID}^*}, f_{\text{ID}^*}, X_{\text{FULL}_{\text{ID}^*}} = \perp, X_{\text{USER}_{\text{ID}^*}} = \perp, X_{\text{SEM}_{\text{ID}^*}} = \perp, d_{\text{SEM}_{\text{ID}^*}} = \perp, d_{\text{USER}_{\text{ID}^*}} = \perp \rangle$ to L_H .

- (ii) Otherwise, for $i \neq l$, M chooses $f_{\text{ID}_i} \xleftarrow{\$} \mathbb{Z}_q^*$, sets $Q_{\text{ID}_i} = f_{\text{ID}_i}^e$, and returns Q_{ID_i} to I . M also sets $X_{\text{FULL}_{\text{ID}_i}} = f_{\text{ID}_i}$, chooses $d_{\text{USER}_{\text{ID}^*}} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$, and sets $X_{\text{SEM}_{\text{ID}_i}} = Q_{\text{ID}_i}^{d_{\text{SEM}_{\text{ID}_i}}}$ as the SEM portion of the user private key and $X_{\text{USER}_{\text{ID}_i}} = f_{\text{ID}_i} Q_{\text{ID}_i}^{-d_{\text{USER}_{\text{ID}_i}}}$ as

the user portion of the user private key. It can be seen that

$$\begin{aligned}
 X_{SEM_{ID_i}} &\times X_{USER_{ID_i}} \\
 &= Q_{ID_i}^{d_{SEM_{ID_i}}} \times f_{ID_i}^{-d_{SEM_{ID_i}}} Q_{ID_i} \\
 &= f_{ID_i} \\
 &= X_{FULL_{ID_i}}.
 \end{aligned} \tag{14}$$

M adds $\langle ID_i, Q_{ID_i}, f_{ID_i}, X_{FULL_{ID_i}}, X_{USER_{ID_i}}, X_{SEM_{ID_i}}, d_{SEM_{ID_i}}, d_{USER_{ID_i}} \rangle$ to L_H . M returns Q_{ID_i} to I .

- (2) *SEM-Extract* (ID_i). If $ID_i = ID^*$ and *USER-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *USER-Extract* (ID^*) has not been queried before, M chooses $d_{SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$ and sets $X_{SEM_{ID^*}} = Q_{ID^*}^{d_{SEM_{ID^*}}}$ as the SEM portion of the user private key, saves it in L_H , and returns it to I . Otherwise for all other $ID_i \neq ID^*$, M just retrieves $X_{SEM_{ID_i}}$ in L_H and returns it to I .
- (3) *USER-Extract* (ID_i). If $ID_i = ID^*$ and *SEM-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *SEM-Extract* (ID^*) has not been queried before, M chooses $d_{USER_{ID^*}} \xleftarrow{\$} \mathbb{Z}_{\phi(N)}^*$ and sets $X_{USER_{ID^*}} = Q_{ID^*}^{d_{USER_{ID^*}}}$ as the USER portion of the user private key, saves it in L_H , and returns it to I . Otherwise for all other $ID_i \neq ID^*$, M just retrieves $X_{USER_{ID_i}}$ in L_H and returns it to I .
- (4) *Identification* (ID_i , session). I will act as the cheating verifier to learn information from valid conversation interactions from M . If $ID_i \neq ID^*$ M just retrieves ID_i 's entry in L_H and runs the identification protocol for either the SEM's interactions, the user's interactions, or both combined as in the protocol.

Otherwise, for $ID^* = ID^*$, M then creates a full valid conversation for each m th query by querying *CHALL* for W_m and setting $Y_{FULL_{ID^*}} = W_m$. Additionally,

M chooses $Y_{m,SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_N^*$ and sets $Y_{m,USER_{ID^*}} = W_m / Y_{m,SEM_{ID^*}}$. Upon receiving c_{m,ID^*} from I , M will query $RSA(W_m(W_0 f_{ID^*}^e)^{c_{m,ID^*}})$ to receive $V_m = (W_m(W_0 f_{ID^*}^e)^{c_{m,ID^*}})^d$ and sets $z_{FULL_{ID^*}} = V_m$. M then chooses $z_{m,SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_N^*$ and then sets $z_{m,USER_{ID^*}} = V_m / z_{m,SEM_{ID^*}}$. M tags the session with *session* = m .

- (i) If I queries *SEM-Identification* (ID^* , m), M will commit $Y_{m,SEM_{ID^*}}$ and respond with $z_{m,SEM_{ID^*}}$ upon receiving c_{m,ID^*} .
- (ii) If I queries *User-Identification* (ID^* , m), M will commit $Y_{m,USER_{ID^*}}$ and respond with $z_{m,USER_{ID^*}}$ upon receiving c_{m,ID^*} .

- (iii) If I queries *Full-Identification* (ID^* , m), M will commit $Y_{m,FULL_{ID^*}}$ and respond with $z_{m,FULL_{ID^*}}$ upon receiving c_{m,ID^*} .

If no session is specified for the query, M just returns the next session in sequence. One can see that *SEM-Identification* (ID^* , m) and *User-Identification* (ID^* , m) can be combined to create a full and valid conversation on session m :

$$\begin{aligned}
 &Y_{m,FULL_{ID^*}} H(ID^*)^{c_{m,ID^*}} \\
 &= Y_{m,SEM_{ID^*}} Y_{m,USER_{ID^*}} Q_{ID^*}^{c_{m,ID^*}} \\
 &= Y_{m,SEM_{ID^*}} \frac{W_m}{Y_{m,SEM_{ID^*}}} (W_0 f_{ID^*}^e)^{c_{m,ID^*}} \\
 &= W_m (W_0 f_{ID^*}^e)^{c_{m,ID^*}} \\
 &= [W_m (W_0 f_{ID^*}^e)^{c_{m,ID^*}}]^{de} \\
 &= \left[[W_m (W_0 f_{ID^*}^e)^{c_{m,ID^*}}]^d \right]^e \\
 &= [V_m]^e \\
 &= \left(\frac{V_m}{z_{m,SEM_{ID^*}}} \right)^e (z_{m,SEM_{ID^*}})^e \\
 &= z_{m,SEM_{ID^*}}^e z_{m,USER_{ID^*}}^e \\
 &= z_{m,FULL_{ID^*}}^e.
 \end{aligned} \tag{15}$$

Eventually, I stops Phase 1 and outputs the challenge ID , ID^* on which it wishes to be challenged on. M checks if $ID^* = ID_i$ from L_H and aborts if not. Otherwise, M runs I now as a cheating prover on ID^* . M runs by obtaining its commitment, $Y_{FULL_{ID^*}}$, selects a challenge $c_1 \xleftarrow{\$} \mathbb{Z}_q^*$, and obtains the response $z_{FULL_{ID^*},1}$ from I . M then resets I to the step whereby I just sent $Y_{FULL_{ID^*}}$, selects a second challenge $c_2 \xleftarrow{\$} \mathbb{Z}_q^*$, and receives $z_{FULL_{ID^*},2}$ as response. It should hold that $z_{FULL_{ID^*},1}^e = Y_{FULL_{ID^*}} H(ID)^{c_1}$ and $z_{FULL_{ID^*},2}^e = Y_{FULL_{ID^*}} H(ID)^{c_2}$. One can then obtain $(z_{FULL_{ID^*},1}^e / z_{FULL_{ID^*},2}^e)^e = H(ID)^{c_1 - c_2}$. Since e is a prime and $-e < c_1 - c_2 < e$, $GCD(e, c_1 - c_2) = 1$. Use the extended Euclidean algorithm to obtain integers S and T such that $eS + (c_1 - c_2)T = 1$. It follows that

$$\begin{aligned}
 H(ID^*) &= Q_{ID^*}^{eS + (c_1 - c_2)T} \mod N \\
 &= (w_0 f_{ID^*}^e)^{eS + (c_1 - c_2)T} \mod N \\
 &= \left(w_0 f_{ID^*}^{e^2 S} \right) \left(w_0 f_{ID^*}^{e(c_1 - c_2)T} \right) \mod N
 \end{aligned}$$

$$\begin{aligned}
&= \left(w_0 f_{ID^*}^{e^2 S} \right) \left(\frac{z_{FULL_{ID^*,1}}}{z_{FULL_{ID^*,2}}} \right)^{eT} \bmod N \\
&= \left[\left(w_0 f_{ID^*}^{eS} \right) \left(\frac{z_{FULL_{ID^*,1}}}{z_{FULL_{ID^*,2}}} \right)^T \right]^e \bmod N.
\end{aligned} \tag{16}$$

M then calculates the solution to the RSAI problem as follows:

$$\begin{aligned}
w_0 f_{ID^*}^e &= \left[\left(w_0 f_{ID^*}^{eS} \right) \left(\frac{z_{FULL_{ID^*,1}}}{z_{FULL_{ID^*,2}}} \right)^T \right]^e \bmod N, \\
w_0^d f_{ID^*} &= \left(w_0 f_{ID^*}^{eS} \right) \left(\frac{z_{FULL_{ID^*,1}}}{z_{FULL_{ID^*,2}}} \right)^T \bmod N, \\
w_0^d &= \frac{\left(w_0 f_{ID^*}^{eS} \right) \left(z_{FULL_{ID^*,1}} / z_{FULL_{ID^*,2}} \right)^T}{f_{ID^*}} \bmod N, \\
w_0^d &= \left(w_0 f_{ID^*}^{eS-1} \right) \left(\frac{z_{FULL_{ID^*,1}}}{z_{FULL_{ID^*,2}}} \right)^T \bmod N.
\end{aligned} \tag{17}$$

M proceeds to calculate the solutions to the other challenges as follows:

$$w_j = V_j(w_0 f_{ID^*})^{-cm, ID^*}. \tag{18}$$

The probability study for the simulation above is similar to that of the impersonation under passive attack game and is therefore omitted. $\square$

4. BNN-SM-IBI: An DL-Based Security-Mediated IBI Scheme

The BNN-SM-IBI scheme is derived from the BNN-IBI scheme proposed in the work of [4] and is provably secure against passive attackers assuming the DL assumption and against active/concurrent attackers assuming the OMDL assumption.

The BNN-SM-IBI scheme is constructed as follows.

- (1) *Setup* (1^k). It takes in the security parameter 1^k . It randomly selects $x \xleftarrow{\$} \mathbb{Z}_q$, a generator $g \xleftarrow{\$} G$ and computes $X = g^x$. Setup also chooses $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q$ and publishes the system parameters $\text{mpk} = \langle G, q, g, X, H \rangle$. The master private key $\text{msk} = x$ is kept secret.
- (2) *Extract* ($\text{mpk}, \text{msk}, \text{ID}$). It takes in $\text{mpk}, \text{msk} = s$, and ID as input. It calculates $H(\text{ID})$, picks $r_{FULL}, r_{SEM}, s_{SEM} \xleftarrow{\$} \mathbb{Z}_q$, calculates $s_{FULL} = r_{FULL} + xH(\text{ID})$ and $R_{FULL} = g^{r_{FULL}}$, and sets the full user private key as $\text{usk}_{FULL} = \langle s_{FULL}, R_{FULL} \rangle$. Additionally, it sets $s_{USER} = s_{FULL} - s_{SEM}$, $r_{USER} = r_{FULL} - r_{SEM}$, $R_{USER} = g^{r_{USER}}$, $R_{SEM} = g^{r_{SEM}}$. It then sets $\text{usk}_{USER} = \langle s_{USER}, R_{USER} \rangle$ and $\text{usk}_{SEM} = \langle s_{SEM}, R_{SEM} \rangle$. It gives

usk_{USER} to the user as its partial private key and usk_{SEM} to the security mediator as its partial private key and keeps usk_{FULL} secret.

- (3) *Identification Protocol*. It is run by the *User-Prover* and *SEM-Prover* with *Verifier* as such.

- (a) *User-Prover* sends its ID to *Sem-Prover*, chooses

$y_{USER} \xleftarrow{\$} \mathbb{Z}_q$, and sets $Y_{USER} = g^{y_{USER}}$ and $S_{USER} = g^{s_{USER}}$. *SEM-Prover* upon receiving ID checks if it is still valid and returns error if it is revoked. Otherwise, *SEM-Prover* chooses $y_{SEM} \xleftarrow{\$} \mathbb{Z}_q$ and sends $Y_{SEM} = g^{y_{SEM}}$, $S_{SEM} = g^{s_{SEM}}$, and R_{SEM} to *User-Prover*. *User-Prover* combines $Y_{FULL} = Y_{SEM} \times Y_{USER}$, $S_{FULL} = S_{SEM} \times S_{USER}$, $R_{FULL} = R_{SEM} \times R_{USER}$ and sends Y_{FULL}, S_{FULL} , and R_{FULL} to *Verifier*.

- (b) *Verifier* chooses a random challenge $c \xleftarrow{\$} \mathbb{Z}_q$ and sends it to *User-Prover*. *User-Prover* relays c to *SEM-Prover*.

- (c) *SEM-Prover* calculates its half of the response $z_{SEM} = y_{SEM} + cs_{SEM}$ and sends it to *User-Prover*. *User-Prover* combines it with his response $z_{USER} = y_{USER} + cs_{USER}$ to create $z_{FULL} = z_{SEM} + z_{USER}$ and sends it as a response to *Verifier*.

Verifier checks if $g^{z_{FULL}} = Y_{FULL} R_{FULL}^c X^{H(\text{ID})c}$ and accepts if yes; otherwise it outputs reject.

To check for completeness,

$$\begin{aligned}
g^{z_{FULL}} &= g^{z_{SEM} + z_{USER}} \\
&= g^{y_{SEM} + cs_{SEM} + y_{USER} + cs_{USER}} \\
&= g^{y_{SEM} + y_{USER} + c(s_{SEM} + s_{USER})} \\
&= g^{y_{FULL} + c(s_{FULL})} \\
&= g^{y_{FULL} + c(r_{FULL} + xH(\text{ID}))} \\
&= g^{y_{FULL}} g^{r_{FULL}c} g^{xH(\text{ID})c} \\
&= Y_{FULL} R_{FULL}^c X^{H(\text{ID})c}.
\end{aligned} \tag{19}$$

4.1. Security Analysis: Impersonation under Passive Attack

Theorem 7. The BNN-SM-IBI scheme is $(t_{\text{SMIBI}}, q_{\text{SMIBI}}, \epsilon_{\text{SMIBI}})$ -secure against impersonation under passive attacks in the random oracle if the DL problem is $(t_{\text{DL}}, \epsilon_{\text{DL}})$ -hard where

$$\epsilon_{\text{SMIBI}} \leq \sqrt{\epsilon_{\text{DL}} e (q_{\text{SMIBI}} + 1)} + \frac{1}{q}. \tag{20}$$

Proof. Assume the BNN-SM-IBI scheme is $(t_{\text{SMIBI}}, q_{\text{SMIBI}}, \epsilon_{\text{SMIBI}})$ -breakable; then a simulator M that $(t_{\text{DL}}, \epsilon_{\text{DL}})$ -breaks the DL problem can be shown. M takes in input

$(g, A = g^a)$ and runs the impersonator I as a subroutine. Without loss of generality, it can be assumed that any *SEM-Extract*, *User-Extract*, *SEM-Identification*, *User-Identification*, and *Full-Identification* queries are preceded by a *Create-User* query. To avoid collision and consistently respond to these queries, M maintains a list $L_H = \langle ID_i, Q_{ID_i}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, r_{USER_{ID_i}}, R_{FULL_{ID_i}}, R_{SEM_{ID_i}}, R_{USER_{ID_i}} \rangle$ which is initially empty. The following shows how M simulates the environment and oracle queries for I .

- (1) *Setup*. M sets the system parameters as $mpk = \langle G, q, g, X = g^x, H \rangle$ and keeps master private key x secret. It sends mpk to I .
- (2) *Create-User* (ID_i). M chooses $l \in \{1, \dots, q_H\}$ randomly and lets $ID_l = ID^*$ at this point. Whenever I makes a H query on ID_i , consider the following.

(i) If $i = l$, M chooses $Q_{ID^*} \xleftarrow{\$} \mathbb{Z}_q$ and sets $S_{FULL_{ID^*}} = A$ and $R_{FULL_{ID^*}} = AX^{Q_{ID^*}}$ and returns Q_{ID^*} to I . M adds $\langle ID_i, Q_{ID^*}, S_{FULL_{ID^*}}, S_{SEM_{ID^*}} = \perp, S_{USER_{ID^*}} = \perp, S_{FULL_{ID^*}} = \perp, S_{SEM_{ID^*}} = \perp, S_{USER_{ID^*}} = \perp, r_{FULL_{ID^*}} = \perp, r_{SEM_{ID^*}} = \perp, r_{USER_{ID^*}} = \perp, R_{FULL_{ID^*}} = \perp, R_{SEM_{ID^*}} = \perp, R_{USER_{ID^*}} = \perp \rangle$ to L_H .

(ii) Otherwise, for $i \neq l$, M chooses $Q_{ID_i} \xleftarrow{\$} \mathbb{Z}_q^*$ and returns Q_{ID_i} . It then chooses $r_{FULL_{ID_i}} \xleftarrow{\$} \mathbb{Z}_q^*$, calculates $s_{FULL_{ID_i}} = r_{FULL_{ID_i}} + xQ_{ID_i}$ and sets $r_{USER_{ID_i}} = r_{FULL_{ID_i}} - r_{SEM_{ID_i}}$ and $s_{USER_{ID_i}} = s_{FULL_{ID_i}} - s_{SEM_{ID_i}}$. M adds $\langle ID_i, Q_{ID_i}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, r_{USER_{ID_i}}, R_{FULL_{ID_i}}, R_{SEM_{ID_i}}, R_{USER_{ID_i}} \rangle$ to L_H .

- (3) *SEM-Extract* (ID_i). If $ID_i = ID^*$ and *USER-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *USER-Extract* (ID^*) has not been queried before, M will choose $r_{SEM_{ID^*}}, s_{SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q$ and sets $R_{SEM_{ID^*}} = g^{r_{SEM_{ID^*}}}$ and $S_{SEM_{ID^*}} = g^{s_{SEM_{ID^*}}}$. M also sets $S_{USER_{ID^*}} = S_{FULL_{ID^*}}/S_{SEM_{ID^*}}$ and $R_{USER_{ID^*}} = R_{FULL_{ID^*}}/R_{SEM_{ID^*}}$, saves these values in L_H , and returns them to I . These values will be used in the event of an *Identification* query later on. Otherwise for all other $ID_i \neq ID^*$, M just retrieves $\langle s_{SEM_{ID_i}}, R_{SEM_{ID_i}} \rangle$ in L_H and returns it to I .
- (4) *USER-Extract* (ID_i). If $ID_i = ID^*$ and *SEM-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *SEM-Extract* (ID^*) has not been queried before, M will choose $r_{USER_{ID^*}}, s_{USER_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q$ and sets $R_{USER_{ID^*}} = g^{r_{USER_{ID^*}}}$ and $S_{USER_{ID^*}} = g^{s_{USER_{ID^*}}}$. M also sets $S_{SEM_{ID^*}} = S_{FULL_{ID^*}}/S_{USER_{ID^*}}$ and $R_{SEM_{ID^*}} = R_{FULL_{ID^*}}/R_{USER_{ID^*}}$, saves these values in L_H , and returns them to I . These values will be

used in the event of an *Identification* query later on. Otherwise for all other $ID_i \neq ID^*$, M just retrieves $\langle s_{USER_{ID_i}}, R_{USER_{ID_i}} \rangle$ in L_H and returns it to I .

- (5) *Identification* ($ID_j, session$). I will act as the cheating verifier to learn information from valid conversation transcripts from M . If $ID_j \neq ID^*$, M just retrieves ID_j 's entry in L_H and runs the identification protocol for either the SEM's interactions, the user's interactions, or both combined as in the protocol.

Otherwise, for $ID_j = ID^*$, M then creates a full valid transcript for each m th query by picking

$z_{m,FULL_{ID^*}}, z_{m,SEM_{ID^*}}, c_{m,ID^*} \xleftarrow{\$} \mathbb{Z}_q$. M retrieves $S_{FULL_{ID^*}}, R_{FULL_{ID^*}}$ and sets $Y_{m,FULL_{ID^*}} = g^{z_{m,FULL_{ID^*}}}$. Additionally, M chooses $z_{m,SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q$, calculates $z_{m,USER_{ID^*}} = z_{m,FULL_{ID^*}} - z_{m,SEM_{ID^*}}$, and retrieves $S_{SEM_{ID^*}}, R_{SEM_{ID^*}}$ and $S_{USER_{ID^*}}, R_{USER_{ID^*}}$ from L_H as well. M then sets $Y_{m,SEM_{ID^*}} = g^{z_{m,SEM_{ID^*}}} S_{SEM_{ID^*}}^{c_{m,ID^*}}$ and $Y_{m,USER_{ID^*}} = g^{z_{m,USER_{ID^*}}} S_{USER_{ID^*}}^{c_{m,ID^*}}$. M tags the session with $session = m$. If I queries

- (i) *SEM-Identification* (ID^*, m), M will return $\langle Y_{m,SEM_{ID^*}}, S_{SEM_{ID^*}}, R_{SEM_{ID^*}}, c_{m,ID^*}, z_{m,SEM_{ID^*}} \rangle$;
- (ii) *User-Identification* (ID^*, m), M will return $\langle Y_{m,USER_{ID^*}}, S_{USER_{ID^*}}, R_{USER_{ID^*}}, c_{m,ID^*}, z_{m,USER_{ID^*}} \rangle$;
- (iii) *Full-Identification* (ID^*, m), M will return $\langle Y_{m,FULL_{ID^*}}, S_{FULL_{ID^*}}, R_{FULL_{ID^*}}, c_{m,ID^*}, z_{m,FULL_{ID^*}} \rangle$.

If no session is specified for the query, M just returns the next session in sequence. One can see that *SEM-Identification* (ID^*, m) and *User-Identification* (ID^*, m) can be combined to create a full and valid transcript on session m :

$$\begin{aligned}
 & Y_{m,FULL_{ID^*}} R_{FULL_{ID^*}}^{c_{m,ID^*}} X^{H(ID^*)c_{m,ID^*}} \\
 &= Y_{m,USER_{ID^*}} Y_{m,SEM_{ID^*}} R_{USER_{ID^*}}^{c_{m,ID^*}} R_{SEM_{ID^*}}^{c_{m,ID^*}} \\
 &\quad \times X^{H(ID^*)c_{m,ID^*}} \\
 &= (g^{z_{m,USER_{ID^*}}} S_{USER_{ID^*}}^{-c_{m,ID^*}} g^{z_{m,SEM_{ID^*}}}) S_{SEM_{ID^*}}^{-c_{m,ID^*}} \\
 &\quad \times (R_{USER_{ID^*}} R_{SEM_{ID^*}})^{c_{m,ID^*}} X^{H(ID^*)c_{m,ID^*}} \\
 &= g^{z_{m,USER_{ID^*}}} g^{z_{m,SEM_{ID^*}}} (S_{USER_{ID^*}} S_{SEM_{ID^*}})^{-c_{m,ID^*}} \\
 &\quad \times (R_{USER_{ID^*}} R_{SEM_{ID^*}})^{c_{m,ID^*}} X^{H(ID^*)c_{m,ID^*}} \\
 &= g^{z_{m,USER_{ID^*}}} g^{z_{m,SEM_{ID^*}}} \left(\frac{S_{FULL_{ID^*}}}{S_{SEM_{ID^*}}} S_{SEM_{ID^*}} \right)^{-c_{m,ID^*}} \\
 &\quad \times \left(\frac{R_{FULL_{ID^*}}}{R_{SEM_{ID^*}}} R_{SEM_{ID^*}} \right)^{c_{m,ID^*}} X^{H(ID^*)c_{m,ID^*}}
 \end{aligned}$$

$$\begin{aligned}
&= g^{z_{m,FULL_{ID^*}} A^{-c_{m,ID^*}} A^{c_{m,ID^*}} X^{-H(ID^*)c_{m,ID^*}}} \\
&\quad \times X^{H(ID^*)c_{m,ID^*}} \\
&= g^{z_{m,FULL_{ID^*}}}.
\end{aligned} \tag{21}$$

Eventually, I stops Phase 1 and outputs the challenge ID , ID^* on which it wishes to be challenged on. M checks if $ID = ID_l$ from L_H and aborts if not. Otherwise, M runs I now as a cheating prover on ID^* by obtaining its commitment, $Y_{FULL_{ID^*}}, S_{FULL_{ID^*}}, R_{FULL_{ID^*}}$ selecting a challenge $c_1 \xleftarrow{\$} \mathbb{Z}_q^*$ and obtaining the response $z_{FULL_{ID^*},1}$ from I . M then resets I to the step whereby I just sent $Y_{FULL_{ID^*}}, S_{FULL_{ID^*}}, R_{FULL_{ID^*}}$, selects a second challenge $c_2 \xleftarrow{\$} \mathbb{Z}_q^*$, and receives $z_{FULL_{ID^*},2}$ as response. M is then able to extract the full user private key as follows:

$$\begin{aligned}
&\frac{z_{FULL_{ID^*},1} - z_{FULL_{ID^*},2}}{c_1 - c_2} \\
&= \frac{y_{FULL_{ID^*}} + c_1 a - y_{FULL_{ID^*}} + c_2 a}{c_1 - c_2} \\
&= \frac{(c_1 - c_2) a}{c_1 - c_2} \\
&= a.
\end{aligned} \tag{22}$$

M outputs a as the discrete log solution.

It remains to calculate the probability of M solving the CDH problem and winning the game. The probability of M successfully extracting two valid conversation transcripts from I is bounded by $(\epsilon_{SMIBI} - (1/q))^2$ as given by the reset lemma [20]:

$$\begin{aligned}
&\Pr [M \text{ wins DL}] \\
&= \Pr [M \text{ computes } a \mid \neg \text{abort}] \\
&= \Pr [M \text{ computes } a \mid \neg \text{abort}] \Pr [\neg \text{abort}] \\
&\epsilon_{DL} \geq \left(\epsilon_{SMIBI} - \frac{1}{q} \right)^2 \Pr [\neg \text{abort}].
\end{aligned} \tag{23}$$

Finally, let δ be the probability that I issues a *SEM-Extract* and a *USER-Extract* query on ID^* and that I makes a total of q_{SMIBI} of such queries. The probability of M answering all the extraction queries is δ_l^q . In Phase 2, the probability of M not aborting is if I outputs the challenge identity ID^* that was not queried before. This is given by the probability $1 - \delta$. Compiling them, the probability of M not aborting is $\delta^{q_{SMIBI}} (1 - \delta)$. This value is maximised at $\delta_{opt} = 1 - (1/(q_{SMIBI} + 1))$. Using δ_{opt} , The probability that M does not abort is at least $1/e(q_{SMIBI} + 1)$ because the value $(1 - (1/(q_{SMIBI} + 1)))^{q_{SMIBI}}$

approaches $1/e$ for large q_{SMIBI} . Therefore, the advantage of M , ϵ_{DL} , and the bound of the simulation is given as follows:

$$\begin{aligned}
\epsilon_{DL} &\geq \left(\epsilon_{SMIBI} - \frac{1}{q} \right)^2 \frac{1}{e(q_{SMIBI} + 1)} \\
\epsilon_{DL} e(q_{SMIBI} + 1) &\geq \left(\epsilon_{SMIBI} - \frac{1}{q} \right)^2 \\
\epsilon_{SMIBI} &\leq \sqrt{\epsilon_{DL} e(q_{SMIBI} + 1)} + \frac{1}{q}.
\end{aligned} \tag{24}$$

□

4.2. Security Analysis: Impersonation under Active/Concurrent Attack

Theorem 8. *The BNN-SM-IBI scheme is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -secure against impersonation under active/concurrent attacks in the random oracle if the OMCDH Problem is $(t_{OMCDH}, q_{OMCDH}, \epsilon_{OMCDH})$ -hard where*

$$\epsilon_{SMIBI} \leq \sqrt{\epsilon_{OMCDH} e(q_{SMIBI} + 1)} + \frac{1}{q}. \tag{25}$$

Proof. Assume the BNN-SM-IBI scheme is $(t_{SMIBI}, q_{SMIBI}, \epsilon_{SMIBI})$ -breakable; then a simulator M that $(t_{OMCDH}, q_{OMCDH}, \epsilon_{OMCDH})$ -breaks the OMCDH Problem can be shown. M takes in input (g, g^a) , is given access to *CHALL* and *DLOG* oracles, and runs the impersonator I as a subroutine. Without loss of generality, it can be assumed that any *SEM-Extract*, *User-Extract*, *SEM-Identification*, *User-Identification*, and *Full-Identification* queries are preceded by a *Create-User* query. To avoid collision and consistently respond to these queries, M maintains a list $L_H = \langle ID_i, Q_{ID_i}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, r_{USER_{ID_i}}, R_{FULL_{ID_i}}, R_{SEM_{ID_i}}, R_{USER_{ID_i}} \rangle$ which is initially empty. The following shows how M simulates the environment and oracle queries for I .

- (1) *Setup.* M sets the system parameters as $\text{mpk} = \langle G, q, g, X = g^x, H \rangle$ and keeps master private key x secret. It sends mpk to I .
- (2) *Create-User (ID_i).* M chooses $l \in \{1, \dots, q_H\}$ randomly and lets $ID_l = ID^*$ at this point. Whenever I makes a H query on ID_i , consider the following.

- (i) If $i = l$, M chooses $Q_{ID^*} \xleftarrow{\$} \mathbb{Z}_q$, queries *CHALL* for W_0 , and sets $S_{FULL_{ID^*}} = W_0$ and $R_{FULL_{ID^*}} = W_0 X^{Q_{ID^*}}$. M returns Q_{ID^*} to I . M adds $L_H = \langle ID_l, Q_{ID^*}, S_{FULL_{ID^*}} = \perp, S_{SEM_{ID^*}} = \perp, S_{USER_{ID^*}} = \perp, S_{FULL_{ID^*}} = \perp, S_{SEM_{ID^*}} = \perp, S_{USER_{ID^*}} = \perp, r_{FULL_{ID^*}} = \perp, r_{SEM_{ID^*}} = \perp, r_{USER_{ID^*}} = \perp, R_{FULL_{ID^*}}, R_{SEM_{ID^*}} = \perp, R_{USER_{ID^*}} = \perp \rangle$ to L_H .

- (ii) Otherwise, for $i \neq l$, M chooses $Q_{ID_i} \xleftarrow{\$} \mathbb{Z}_q^*$ and returns Q_{ID_i} . It then chooses $r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, s_{SEM_{ID_i}} \xleftarrow{\$} \mathbb{Z}_q^*$, calculates

$s_{FULL_{ID_i}} = r_{FULL_{ID_i}} + xQ$, and sets $r_{USER_{ID_i}} = r_{FULL_{ID_i}} - r_{SEM_{ID_i}}$ and $s_{USER_{ID_i}} = s_{FULL_{ID_i}} - s_{SEM_{ID_i}}$. M adds $L_H = \langle ID_i, Q_{ID_i}, S_{FULL_{ID_i}}, S_{SEM_{ID_i}}, S_{USER_{ID_i}}, s_{FULL_{ID_i}}, s_{SEM_{ID_i}}, s_{USER_{ID_i}}, r_{FULL_{ID_i}}, r_{SEM_{ID_i}}, r_{USER_{ID_i}}, R_{FULL_{ID_i}}, R_{SEM_{ID_i}}, R_{USER_{ID_i}} \rangle$ to L_H .

(3) *SEM-Extract* (ID_i). If $ID_i = ID^*$ and *USER-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *USER-Extract* (ID^*) has not been queried before, M will choose $r_{SEM_{ID^*}}, s_{SEM_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q$ and sets $R_{SEM_{ID^*}} = g^{r_{SEM_{ID^*}}}$ and $S_{SEM_{ID^*}} = g^{s_{SEM_{ID^*}}}$. M also sets $S_{USER_{ID^*}} = S_{FULL_{ID^*}}/S_{SEM_{ID^*}}$ and $R_{USER_{ID^*}} = R_{FULL_{ID^*}}/R_{SEM_{ID^*}}$, saves these values in L_H , and returns them to I . These values will be used in the event of an *Identification* query later on. Otherwise for all other $ID_i \neq ID^*$, M just retrieves $\langle s_{SEM_{ID_i}}, R_{SEM_{ID_i}} \rangle$ in L_H and returns it to I .

(4) *USER-Extract* (ID_i). If $ID_i = ID^*$ and *SEM-Extract* (ID^*) has been queried before, M aborts. If $ID_i = ID^*$, but *SEM-Extract* (ID^*) has not been queried before, M will choose $r_{USER_{ID^*}}, s_{USER_{ID^*}} \xleftarrow{\$} \mathbb{Z}_q$ and sets $R_{USER_{ID^*}} = g^{r_{USER_{ID^*}}}$ and $S_{USER_{ID^*}} = g^{s_{USER_{ID^*}}}$. M also sets $S_{SEM_{ID^*}} = S_{FULL_{ID^*}}/S_{USER_{ID^*}}$ and $R_{SEM_{ID^*}} = R_{FULL_{ID^*}}/R_{USER_{ID^*}}$, saves these values in L_H , and returns them to I . These values will be used in the event of an *Identification* query later on. Otherwise for all other $ID_i \neq ID^*$, M just retrieves $\langle s_{USER_{ID_i}}, R_{USER_{ID_i}} \rangle$ in L_H and returns it to I .

(5) *Identification* (ID_j , session). I will act as the cheating verifier to learn information from valid conversations from M . If $ID_j \neq ID^*$, M just retrieves ID_j 's entry in L_H and runs the identification protocol for either the SEM's interactions, the user's interactions, or both combined as in the protocol.

Otherwise, for $ID_i = ID^*$, M then creates a full valid conversation for each m th query by querying *CHALL* for W_m and setting $Y_{m,FULL_{ID^*}} = W_m$. M also retrieves. Additionally, M chooses $Y_{m,SEM_{ID_j}} \xleftarrow{\$} G$ and sets $Y_{m,USER_{ID_j}} = W_m/Y_{m,SEM_{ID_j}}$. M also retrieves $S_{m,FULL_{ID_j}}, S_{m,SEM_{ID_j}}, S_{m,USER_{ID_j}}, R_{m,FULL_{ID_j}}, R_{m,SEM_{ID_j}}$, and $R_{m,USER_{ID_j}}$ from L_H . Upon receiving c_{m,ID_j} from I , M will query $z_{m,FULL_{ID_j}} = DLOG(W_m W_0^{c_{m,ID_j}})$, selects $z_{m,SEM_{ID_j}} \xleftarrow{\$} \mathbb{Z}_q$, and sets $z_{m,USER_{ID_j}} = z_{m,FULL_{ID_j}} - z_{m,SEM_{ID_j}}$. M tags the session with *session* = m .

(a) If I queries *SEM-Identification* (ID_j , m), M will commit $Y_{m,SEM_{ID_j}}, S_{SEM_{ID_j}}, R_{SEM_{ID_j}}$ and respond with $z_{m,SEM_{ID_j}}$ upon receiving c_{m,ID_j} .

(b) If I queries *User-Identification* (ID_j , m), M will commit $Y_{m,USER_{ID_j}}, S_{USER_{ID_j}}, R_{USER_{ID_j}}$ and respond with $z_{m,USER_{ID_j}}$ upon receiving c_{m,ID_j} .

(c) If I queries *Full-Identification* (ID_j , m), M will commit $Y_{m,FULL_{ID_j}}, S_{FULL_{ID_j}}, R_{FULL_{ID_j}}$ and respond with $z_{m,FULL_{ID_j}}$ upon receiving c_{m,ID_j} .

If no session is specified for the query, M just returns the next session in sequence. One can see that *SEM-Identification* (ID_j , m) and *User-Identification* (ID^* , m) can be combined to create a full and valid conversation on session m :

$$\begin{aligned}
 & Y_{m,FULL_{ID_j}} R_{FULL_{ID_j}}^{c_{m,ID_j}} X^{H(ID_j)c_{m,ID_j}} \\
 &= Y_{m,USER_{ID_j}} Y_{m,SEM_{ID_j}} R_{USER_{ID_j}}^{c_{m,ID_j}} R_{SEM_{ID_j}}^{c_{m,ID_j}} X^{H(ID_j)c_{m,ID_j}} \\
 &= \frac{W_m}{Y_{m,SEM_{ID_j}}} Y_{m,SEM_{ID_j}} \left(\frac{W_0 X^{-H(ID)}}{R_{SEM_{ID_j}}} \right)^{c_{m,ID_j}} \\
 &\quad \times R_{SEM_{ID_j}}^{c_{m,ID_j}} X^{H(ID_j)c_{m,ID_j}} \\
 &= W_m \left(W_0^{c_{m,ID_j}} \right) X^{-H(ID)c_{m,ID_j}} X^{H(ID_j)c_{m,ID_j}} \\
 &= W_m \left(W_0^{c_{m,ID_j}} \right) \\
 &= g^{DLOG(W_m (W_0^{c_{m,ID_j}}))} \\
 &= g^{z_{m,FULL_{ID_j}}}.
 \end{aligned} \tag{26}$$

Eventually, I stops Phase 1 and outputs the challenge ID , ID^* on which it wishes to be challenged on. M checks if $ID^* = ID_i$ from L_H and aborts if not. Otherwise, M runs I now as a cheating prover on ID^* . M runs by obtaining its commitment, $Y_{FULL_{ID^*}}, S_{FULL_{ID^*}}, R_{FULL_{ID^*}}$, selects a challenge $c_1 \xleftarrow{\$} \mathbb{Z}_q^*$, and obtains the response $z_{FULL_{ID^*},1}$ from I . M then resets I to the step whereby I just sent $Y_{FULL_{ID^*}}, S_{FULL_{ID^*}}, R_{FULL_{ID^*}}$, selects a second challenge $c_2 \xleftarrow{\$} \mathbb{Z}_q^*$, and receives $z_{FULL_{ID^*},2}$ as response. M is then able to extract the full user private key as follows:

$$\begin{aligned}
 & \frac{z_{FULL_{ID^*},1} - z_{FULL_{ID^*},2}}{c_1 - c_2} \\
 &= \frac{y_{FULL_{ID^*}} + c_1 w_0 - y_{FULL_{ID^*}} + c_2 w_0}{c_1 - c_2} \\
 &= \frac{(c_1 - c_2) w_0}{c_1 - c_2} \\
 &= w_0.
 \end{aligned} \tag{27}$$

M outputs w_0 as the initial discrete log challenge solution.

TABLE 1: Operation costs for the GQ-SM-IBI scheme.

Algorithm	A	M1	M2	E
Setup	0	0	0	0
Extract	1	0	0	3
SEM-Prove	0	0	1	2
User-Prove	0	0	3	2
Verify	0	0	1	2

A: addition in $\mathbb{Z}_{\phi(N)}$, M1: multiplication in $\mathbb{Z}_{\phi(N)}$, M2: multiplication in $\mathbb{Z}_N^*$, and E: exponentiation in $\mathbb{Z}_N^*$.

TABLE 2: Communication costs between algorithms for the GQ-SM-IBI scheme.

Algorithms	Bitstring ID	Elements of $ e - 1 $	Elements in $\mathbb{Z}_N^*$
TA-User (Extract)	1	0	1
TA-SEM (Extract)	1	0	1
SEM-User (Identification)	1	1	2
User-Verifier (Identification)	1	1	2

M proceeds to calculate the solutions to the other challenges as follows:

$$\begin{aligned}
 & z_{m, \text{FULL}_{ID_j}} - w_0 c_{m, ID_j} \\
 &= DLOG \left(W_m W_0^{c_{m, ID_j}} \right) - w_0 c_{m, ID_j} \\
 &= \left[w_m + w_0 c_{m, ID_j} \right] - w_0 c_{m, ID_j} \\
 &= w_m.
 \end{aligned} \tag{28}$$

The probability study for the simulation above is similar to that of the impersonation under passive attack game and is therefore omitted. $\square$

5. Efficiency Analysis

In this section we provide the breakdown of operation costs for both the GQ-SM-IBI scheme and the BNN-SM-IBI scheme.

We measure the operation costs of the GQ-SM-IBI scheme in terms of addition operations in $\mathbb{Z}_{\phi(N)}$, multiplication operations in $\mathbb{Z}_{\phi(N)}$ and $\mathbb{Z}_N^*$, and exponentiation operations in $\mathbb{Z}_N^*$. Overall, the operational costs of the GQ-SM-IBI scheme are given in Table 1.

We also provide the communication costs of the GQ-SM-IBI scheme in Table 2.

As for the BNN-SM-IBI scheme, we measure the operational costs in terms of group operational costs of addition modulo q , multiplication modulo q , multiplication in group G , and exponentiations modulo q . Overall, the operational costs of the BNN-SM-IBI scheme are given in Table 3.

We also provide the communication costs of the BNN-SM-IBI scheme in Table 4.

TABLE 3: Operation costs for the BNN-SM-IBI scheme.

Algorithm	A	M	GM	E
Setup	0	0	0	1
Extract	3	1	0	3
SEM-Prover	1	1	0	2
User-Prover	2	1	3	2
Verifier	0	1	2	3

H: hash operation, A: addition mod q , M: multiplication mod q , GM: group multiplication, and E: exponentiation mod q .

TABLE 4: Communication costs between algorithms for the BNN-SM-IBI scheme.

Algorithms	Bitstring	Elements in G_1	Elements in $\mathbb{Z}_q$
TA-User (Extract)	1	1	1
TA-SEM (Extract)	1	1	1
SEM-User (Identification)	1	3	2
User-Verifier (Identification)	1	3	2

TABLE 5: Comparison of average simulated runtimes for SM-IBI schemes.

	Identification (ns)
GQ-SM-IBI (1024-bits)	2,158,747
BNN-SM-IBI (1024-bits)	14,064,652
BLS-SM-IBI (512-bits)	104,664,826
GQ-SM-IBI (2048-bits)	7,706,801
BNN-SM-IBI (2048-bits)	82,456,452
BLS-SM-IBI (1024-bits)	487,682,172
GQ-SM-IBI (3072-bits)	17,027,899
BNN-SM-IBI (3072-bits)	182,153,886
BLS-SM-IBI (1536-bits)	1,188,023,987

However, it is difficult to compare all these operational costs since the schemes are using operations defined in different fields and groups. Therefore, we built a simulator to generate running time results using a common platform.

We compare the running time of only the identification protocol, since the protocol is invoked every time a prover wishes to perform an interaction with a verifier. We compare the running times of the GQ-SM-IBI scheme and BNN-SM-IBI scheme as well as the original pairing-based scheme by [19]. We name the scheme from [19] as BLS-SM-IBI since its design follows the BLS signature scheme by Boneh et al. from [22]. For GQ-SM-IBI and BNN-SM-IBI the level of security used was 1024-bits, 2048-bits, and 3072-bits. We also compared the results with the equivalent level of security for BLS-SM-IBI pairing-based scheme of 512-bits, 1024-bits, and 1536-bits, respectively.

The simulation was run on a i7-2630QM platform with 4 GB RAM running 64-bit Windows 7. The library used was Java Cryptography Extension. We ran the simulation 100

iterations for each algorithm and took the average running time, measured in nanoseconds as presented in Table 5.

From the results, one can see that the GQ-SM-IBI scheme has the fastest identification protocol running time with the BNN-SM-IBI scheme trailing behind. However, both pairing-free schemes vastly outperform the BLS-SM-IBI that is pairing-based with at least an approximate factor of 6 to 7 for all three security levels. Therefore we obtain faster SM-IBI schemes from pairing-free alternatives.

6. Conclusion

We proposed two SM-IBI schemes that have an instant revocation feature and are very efficient. Our schemes outperform the only pairing-based SM-IBI currently known and are provably secure in the random oracle model against both passive and active/concurrent attackers.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to acknowledge the Ministry of Education, Malaysia, for financially aiding this research through the Fundamental Research Grant Scheme FRGS/2/2013/ICT07/MMU/03/5.

References

- [1] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, G. R. Blakley and D. Chaum, Eds., vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, Berlin, Germany, 1984.
- [2] D. Boneh, X. Ding, G. Tsudik, and C. M. Wong, "A method for fast revocation of public key certificates and security capabilities," in *Proceedings of the 10th Conference on USENIX Security Symposium (SSYM '01)*, D. S. Wallach, Ed., vol. 10, p. 22, Berkeley, Calif, USA, 2001.
- [3] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," in *Advances in Cryptology-CRYPTO '86*, A. M. Odlyzko, Ed., vol. 263 of *Lecture Notes in Computer Science*, pp. 186–194, Springer, Berlin, Germany, 1986.
- [4] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," in *Advances in Cryptology-EUROCRYPT 2004*, C. Cachin and J. Camenisch, Eds., vol. 3027 of *Lecture Notes in Computer Science*, pp. 268–286, Springer, Berlin, Germany, 2004.
- [5] K. Kurosawa and S. H. Heng, "From digital signature to ID-based identification/signature," in *Public Key Cryptography-PKC 2004*, F. Bao, R. H. Deng, and J. Zhou, Eds., vol. 2947 of *Lecture Notes in Computer Science*, pp. 248–261, Springer, Berlin, Germany, 2004.
- [6] J. J. Chin, S. H. Heng, and B. M. Goi, "An efficient and provable secure identity-based identification scheme in the standard model," in *Public Key Infrastructure*, S. F. Mjølne, S. Mauw, and S. K. Katsikas, Eds., vol. 5057 of *Lecture Notes in Computer Science*, pp. 60–73, Springer, Berlin, Germany, 2008.
- [7] J. J. Chin and S. H. Heng, "An adaptive-secure k -resilient identity-based identification scheme in the standard model," in *Proceedings of the FTRA-ACSA Summer*, Vancouver, Canada, June 2012.
- [8] J. J. Chin, S. H. Heng, and B. M. Goi, "Hierarchical identity-based identification schemes," in *Security Technology*, D. Slezak, T. H. Kim, W. C. Fang, and K. P. Arnett, Eds., vol. 58 of *Communications in Computer and Information Science*, pp. 93–99, Springer, Berlin, Germany, 2009.
- [9] A. Fujioka, T. Saito, and K. Xagawa, "Secure hierarchical identity-based identification without random oracles," in *Information Security*, D. Gollmann and F. C. Freiling, Eds., vol. 7483 of *Lecture Notes in Computer Science*, pp. 258–273, Springer, Berlin, Germany, 2012.
- [10] A. Fujioka, T. Saito, and K. Xagawa, "Applicability of OR-proof techniques to hierarchical identity-based identification," in *Cryptology and Network Security*, J. Pieprzyk, A. R. Sadeghi, and M. Manulis, Eds., vol. 7712 of *Lecture Notes in Computer Science*, pp. 169–184, Springer, Berlin, Germany, 2012.
- [11] J. J. Chin, R. C. W. Phan, R. Behnia, and S. H. Heng, "An efficient and provably secure certificateless identification scheme," in *SECURITY 2013*, P. Samarati, Ed., pp. 371–378, SciTePress, 2013.
- [12] X. Ding and G. Tsudik, "Simple identity-based cryptography with mediated RSA," in *Topics in Cryptology-CT-RSA 2003*, M. Joye, Ed., vol. 2612 of *Lecture Notes in Computer Science*, pp. 193–210, Springer, Berlin, Germany, 2003.
- [13] B. Libert and J. J. Quisquater, "Efficient revocation and threshold pairing based cryptosystems," in *Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing (PODC '03)*, E. Borowsky and S. Rajsbaum, Eds., pp. 163–171, ACM, Boston, Mass, USA, July 2003.
- [14] X. Cheng, L. Guo, and X. Wang, "An identity-based mediated signature scheme from bilinear pairing," *International Journal of Network Security*, vol. 2, no. 1, pp. 29–33, 2006.
- [15] S. S. M. Chow, C. Boyd, and J. M. G. Nieto, "Security-mediated certificateless cryptography," in *Public Key Cryptography-PKC 2006*, M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, Eds., vol. 3958 of *Lecture Notes in Computer Science*, pp. 508–524, Springer, Berlin, Germany, 2006.
- [16] W. S. Yap, S. S. M. Chow, S. H. Heng, and B. M. Goi, "Security mediated certificateless signatures," in *Applied Cryptography and Network Security*, J. Katz and M. Yung, Eds., vol. 4521 of *Lecture Notes in Computer Science*, pp. 459–477, Springer, Berlin, Germany, 2007.
- [17] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology-ASIACRYPT 2003*, C.-S. Lai, Ed., vol. 2894 of *Lecture Notes in Computer Science*, pp. 452–473, Springer, Berlin, Germany, 2003.
- [18] J. J. Chin, R. Behnia, S. H. Heng, and R. C. W. Phan, "Cryptanalysis of a certificateless identification scheme," *Security and Communication Networks*, 2014.
- [19] J. J. Chin, R. Behnia, S. H. Heng, and R. C. Phan, "An efficient and provable secure security-mediated identity-based identification scheme," in *Proceedings of the 8th Asia Joint Conference on Information Security (Asia JCIS '13)*, pp. 27–32, IEEE, Seoul, South Korea, 2013.
- [20] M. Bellare and A. Palacio, "GQ and Schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks," in *Advances in Cryptology-CRYPTO*

2002, M. Yung, Ed., vol. 2442 of *Lecture Notes in Computer Science*, pp. 162–177, Springer, Berlin, Germany, 2002.

- [21] S. Y. Tan, S. H. Heng, R. C. W. Phan, and B. M. Goi, “A variant of Schnorr identity-based identification scheme with tight reduction,” in *Future Generation Information Technology*, T. H. Kim, H. Adeli, D. Slezak et al., Eds., vol. 7105 of *Lecture Notes in Computer Science*, pp. 361–370, Springer, Berlin, Germany, 2011.
- [22] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the weil pairing,” *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.