An Efficient and Provable Secure Certificateless Identification Scheme in the Standard Model

Ji-Jian Chin[1] , Swee-Huay Heng[2] and Raphael C.-W. Phan[1]
[1] Faculty of Engineering, Multimedia University, Cyberjaya, Selangor, Malaysia
[2]Faculty of Information Science and Technology, Multimedia University, Melaka, Malaysia
[{e-mail: jjchin,shheng,raphael}@mmu.edu.my]
*Corresponding author: Ji-Jian Chin

Received November 8, 2013; revised April 8, 2014; accepted June 5, 2014; published July 29, 2014

Abstract

In Asiacrypt 2003, Al-Riyami and Paterson proposed the notion of certificateless cryptography, a technique to remove key escrow from traditional identity-based cryptography as well as circumvent the certificate management problem of traditional public key cryptography. Subsequently much research has been done in the realm of certificateless encryption and signature schemes, but little to no work has been done for the identification primitive until 2013 when Chin et al. rigorously defined certificateless identification and proposed a concrete scheme. However Chin et al.'s scheme was proven in the random oracle model and Canetti et al. has shown that certain schemes provable secure in the random oracle model can be insecure when random oracles are replaced with actual hash functions. Therefore while having a proof in the random oracle model is better than having no proof at all, a scheme to be proven in the standard model would provide stronger security guarantees. In this paper, we propose the first certificateless identification scheme that is both efficient and show our proof of security in the standard model, that is without having to assume random oracles exist.

Keywords: Cryptography, Privacy/Authorization/Authentication, Certificateless Identification, Standard Model, Proof of Security

http://dx.doi.org/10.3837/tiis.2014.07.019

1. Introduction

In traditional public key cryptography, a certificate authority is in charge of issuing certificates to users verifying that the public keys indeed belong to them. The down side of these certificates is that when the amount of users grows larger, the cost of managing these certificates increases as well. In identity-based cryptography, first proposed by Shamir in [1], a user can implicitly certify himself through the use of a unique identity-string binding him to his user secret key. However, the Trusted Authority who generates these keys still has access to the master secret key and therefore knows every user's secret. This key escrow, although desirable in certain situations, poses a security concern in others.

In the paradigm of certificateless cryptography, first proposed by Al-Riyami and Paterson in [2], the key generation center generates only a partial private key of the user. The user takes this partial secret key and combines it with a personal secret value to produce a full user secret key. This secret value is hidden from the key generation center and thus removes the key escrow. Identity-based cryptography also has the desired property of doing away with certificates, similar to certificateless cryptography. However, while there have been many advances in the realm of encryption and signature primitives for certificateless cryptography, the identification primitive has been virtually untouched.

An identification scheme allows a prover to prove himself to a verifier with the verifier learning nothing about the prover's secret key. Neven et al. in [3] and Kurosawa and Heng in [4] first pioneered the rigorous definition of identity-based identification, and their work has been incrementally improved upon over the most of the last decade in works such as [5-12]. However, to date little has been done to define and construct certificateless identification schemes. Some preliminary work has come in the form of [13] where the authors proposed a new and fairly effective scheme but without a proper definition of security models and proofs. [14] recently showed that the schemes in [13] are indeed insecure. A second but more comprehensive independent work has come in the form of [15] but the scheme is only provable secure in the random oracle model.

The random oracle model was first proposed by Bellare and Rogaway in [16]. Random oracles are treated as idealistic hash functions in a security proof where no mathematical parameters can be used to define the properties of random oracles. Open to honest and malicious parties alike, random oracles generate fully random responses to new queries while returning the same responses for queries that have been made before. However, since they are idealistic, random oracles cannot exist in the real world. In practice, regular hash functions are used to substitute these random oracles when implementing a cryptosystem. Canetti et al. in [17] showed that there are instances of cryptosystems where the cryptosystem can be broken if the random oracles are replaced by ordinary hash functions. Therefore it is our observation that while proofs of security in the random oracle model are better than having no proofs at all, it is more desirable to construct cryptosystems that are provable secure in the standard model.

In this paper, we provide the definitions of certificateless identification and proceed to construct a certificateless identification scheme that is efficient and provable secure in the standard model. This, as opposed to a proof in the random oracle model, is more desirable in rigorously defining security for a cryptosystem.

We divide our paper into the following sections: In Section 2 we introduce preliminary definitions required for certificateless identification schemes. In Section 3 we show the construction of our scheme. In Section 4 we provide four security proofs in the standard model - passive security against Type-1 and Type-2 adversaries, as well as active/concurrent security

against Type-1 and Type-2 adversaries. In Section 5 we show the operation costs of the scheme and conclude in Section 6.

2. Preliminaries

2.1. Bilinear Pairings

Let G_1 and G_2 be cyclic multiplicative groups of prime order q where the discrete logarithm problems are intractable. Then e: $G_1 \times G_1 \rightarrow G_T$ is an admissible bilinear map if it satisfies

1) Bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$ for all generators $g \in G$ and $a, b \in Z_q^*$
2) Non-degenaracy: $e(g, g) \neq 1$.
3) Computability: computation of the bilinear map *e* should be efficient.

2.2. Problems and Assumptions

We use the following hard problems to prove our certificateless identification scheme is secure:

a) Computational Diffie-Hellman problem (CDHP):Given g, g^a, g^b for some $a, b \in Z_q^*$, compute g^{ab}.

b) One-More computational Diffie-Hellman Problem (OMCDHP): The One-More Computational Diffie-Hellman Problem is modeled as a game played by an adversary where the adversary is given $1^k, G, G_T, g, g^a$ as input and access to two oracles *CHALL* and *CDH*. *CHALL* on any input returns a random point W_i, while *CDH* on any input h will return h^a. The adversary is required to find $W_0^a, \ldots, W_n^a$ while using the *CDH* oracle only $i < n$ times.
The OMCDHP was first proposed by [18] and subsequently used by [3] to prove security against impersonation under active/concurrent attacks for the pairing family schemes in their paper. Subsequent work utilized the one-more problems frequently to achieve active/concurrent level security for identification schemes.

The Computational Diffie-Hellman assumption and the One-More Computational Diffie-Hellman assumption state that there are no polynomial time algorithms for solving the discrete logarithm problem and the one-more discrete logarithm problems with non-negligible probability respectively.

2.3. The Knowledge of Exponent Assumption [19]

We use the Knowledge of Exponent Assumption for the proof against Type-1 adversaries, for the case when a target identity's public key is replaced. Let $k = \log |\langle g \rangle|$ be the security parameter of a prime order group where g is a generator. For any probabilistic polynomial time algorithm $\boldsymbol{A}$ that takes as input g and g^a, where a is chosen from $[0, |\langle g \rangle| - 1]$ uniformly at random, and which produces as output a pair of the form $(x, y), x \in \langle g \rangle$, there exists a probabilistic polynomial time extractor $\boldsymbol{E}$, which takes in the same input and outputs the pair (x, y) along with an exponent r such that for sufficiently large k, $\Pr[y = x^a \wedge g^r \neq x] \leq \frac{1}{Q^k}$ for any polynomial Q.

2.4. Definition for Certificateless Identification Schemes

A certificateless scheme consists of six probabilistic polynomial time algorithms **(Setup, Set-User-Key, Partial-Private-Key-Extract, Set-Private-Key, Prove and Verify)**.

1) **Setup** is run by the key generation center. Taking in the security parameter 1^k as input, it returns the master public key *MPK* and the master secret key *MSK*. It publishes *MPK* and

securely stores *MSK*.

2) **Set-User-Key** is run by the user when creating his own account. Taking in the security parameter 1^k and the user's identity *ID* as input, it generates the secret value for a user SV_{ID} as well as a corresponding public key UPK_{ID} which it publishes.
3) **Partial-Private-Key-Extract** is run by the key generation center upon a user's request for a partial private key. It takes in *MPK*, *MSK*, a user's identity *ID* and the user's public key UPK_{ID}. It returns the partial private key PPK_{ID} for the user via a secure channel.
4) **Set-Private-Key** is done by the user to combine the user's identity *ID*, partial private key PPK_{ID}, user public key UPK_{ID} and secret value SV_{ID} into the full private key. It returns the user private key as USK_{ID}.
5) **Identification Protocol** is the interactive protocol run by the 2 algorithms **Prove** and **Verify**. Both algorithms take in the master public key MPK, **Prove**'s identity string *ID*, user public key UPK_{ID}. **Prove** takes in the user private key USK_{ID} as auxiliary input. They perform the three-step canonical honest verifier zero knowledge proof of knowledge protocol with the following steps:
   (a) Commitment: **Prove** sends a commitment CMT to **Verify**.
   (b) Challenge: **Verify** sends to **Prove** a challenge CHA randomly chosen from a predefined set.
   (c) Response: **Prove** returns a response RSP where **Verify** will either choose to accept or reject.

2.5. Security Notion For Certificateless Identification Scheme

We consider four types of adversaries for the certificateless identification scheme:

1) The Type-1 passive impersonator (IMP-PA-1) and the active/concurrent impersonator (IMP-AA/CA-1) model malicious users attacking the scheme. Type-1 impersonators do not have access to the master secret key, but are able to request and replace public keys with values of their choosing.
2) The Type-2 passive impersonator (IMP-PA-2) and the active/concurrent impersonator (IMP-AA/CA-2) model malicious key generation centers attacking a particular user. Type-2 impersonators can generate partial private keys of users since they have access to the master secret key, and are able to replace public keys of users of their choice except the target user being attacked.

The difference in capability between passive and active impersonators is the passive impersonator can only eavesdrop on conversations between honest parties, while the active impersonator can act as a cheating verifier to gain knowledge from honest provers through interacting with them. The concurrent impersonator is an active impersonator who can run several instances of the protocol simultaneously.

We also classify adversary subtypes based on adversaries of certificateless signature schemes according to the definitions by [20,21]. These subtypes are the *Normal*, *Strong* and *Super* type adversary for Type 1 and Type 2 categories, which are differing in what parameters they have.

1) *Normal*-type adversaries cannot use a prover to converse with a verifier once its public key is replaced.
2) *Strong*-type adversaries can continue using a prover whose public key has been replaced, provided they supply the secret value corresponding to the replaced public key for the conversation.

3) *Super*-type adversaries can replace a prover's public key and still use it to correspond with a verifier without the new secret value.

The strength of these classifications are in increasing order, i.e. if a scheme is secure against super-type adversaries, it is secure against normal-type adversaries as well.

We describe the security model of CLI schemes against Type-1 and Type-2 impersonators in terms of the following games. We highlight the differences between each game with respect to the capabilities when making identification queries, for both passive and active/concurrent impersonators as well as *Normal*, *Strong* and *Super* adversaries.

Game I. The game played between a challenger C and the Type-1 impersonator I_1 for the CLI scheme is as follows:

1) **Setup**. C runs **Setup**, generates and passes the system parameters *params* to I_1 and keeps the master secret key *MSK*.
2) Phase 1: I_1 is allowed to make the following queries adaptively to C.
   a) **ExtrPartSK(*ID*)**. On request for the partial private key on user *ID*, C will run **Partial-Private-Key-Extract** and returns the user's partial private key PPK_{ID} to I_1.
   b) **ExtrFullSK(*ID*)**. On request for the full private key on user *ID*, C will run **Partial-Private-Key-Extract**, **Set-Secret-Value**, **Set-Private-Key** algorithms to generate the user's full private key and pass it to I_1.
   c) **RequestPK(*ID*)**. On request for the user public key on user *ID*, C will run **Set-User-Key** to generate the user's public key UPK_{ID} and pass it to I_1.
   d) **ReplacePK(*ID*,UPK'_{ID})**. I_1 will replace the user *ID*'s public key UPK_{ID} with the public key UPK'_{ID} chosen by him. The corresponding secret value is not required for public key replacement queries.
   e) **Identification(*ID*).** For passive I_1, C will generate a valid transcript between user *ID* and itself as the verifier and return the transcript to I_1. For active/concurrent I_1, C will play the role of the prover to interact with I_1 as the cheating verifier.
      i) *Normal*-type adversaries cannot make an identification query for a prover if its public key is replaced.
      ii) *Strong*-type adversaries have to additionally supply sv which is the corresponding secret value for the public key. If $sv = \perp$ then the user's public key is the original one. Otherwise C will use sv in the conversation for the replaced public key.
      iii) *Super*-type adversaries can continue to make identification queries, even for replaced public keys, without supplying sv.
3) **Phase 2**. I_1 will eventually output the challenge identity ID^* and then changes role to then play the role of the cheating prover. C, in turn, assumes the role of the verifier. I_1 wins the game if it manages to convince C to accept.

We say a CLI scheme is (t, q_I, ε)-secure under passive or active/concurrent attacks if for any passive or active/concurrent Type-1 impersonator I_1 who runs in time t, $\Pr[I_1\ can\ impersonate] < \varepsilon$, where I_1 can make at most q_I extract queries on full private keys.

Game II. The game played between a challenger C and the Type-2 Impersonator I_2 for the CLI scheme is as follows:

1) **Setup**. C runs **Setup** and passes both the system parameters *params* and the master secret key *MSK* to I_2.
2) **Phase 1**: I_2 will be allowed to make the following queries adaptively to C.
   a) **ExtrFullSK(*ID*)**.On request for the full private key USK_{ID} on user *ID*, C will run **Partial-Private-Key-Extract**, **Set-Secret-Value**, **Set-Private-Key** algorithms to generate the user's full private key. It passes the full private key to I_2.
   b) **RequestPK(*ID*)**. On request for the user public key on user *ID*, C will run **Set-User-Key** to generate the user's public key UPK_{ID} and pass it to I_2.
   c) **ReplacePK(ID,UPK'_{ID})**. I_2 is able to replace the user *ID*'s public key UPK_{ID} with the public key UPK'_{ID} chosen by him. Once again, the corresponding secret value is not required for public key replacement queries. The only exceptions are the targets *ID* and ID^*, otherwise it will be trivial to win the game.
   d) **Identification(*ID*)**. For passive I_2, C will generate a valid transcript between user *ID* and itself as the verifier and return the transcript to I_2. For active/concurrent I_2, C will play the role of the prover to interact with I_2 as the cheating verifier.
      i) *Normal*-type adversaries cannot make an identification query for a prover if its public key is replaced.
      ii) *Strong*-type adversaries have to additionally supply sv which is the corresponding secret value for the public key. If $sv = \perp$ then the public key is the original one. Otherwise C will use sv in the conversation for the replaced public key.
      iii) *Super*-type adversaries can continue to make identification queries, even for replaced public keys, without supplying sv.
3) **Phase 2**. I_2 will eventually output the challenge identity, ID^* and change roles to then play the role of the cheating prover. C will assume the role of the verifier. I_2 wins the game if it manages to convince C to accept.

Note that I_2 does not need to perform **ExtrPartSK** queries as it already has the master secret key and can generate partial private keys by itself. I_2 is also not allowed to replace the public key of the challenge identity, but is able to do so for any other user.

We say a CLI scheme is (t, q_I, ε)-secure under passive or active/concurrent attacks if for any passive or active/concurrent Type-2 impersonator I_2 who runs in time t, $\Pr[I_2\ can\ impersonate] < \varepsilon$, where I_2 can make at most q_I extract queries on full private keys.

3. Construction

In this section we show the construction of the new certificateless identification scheme. Let G and G_T be finite cyclic groups of large prime order q and let g be a generator of G. Use a collision-resistant hash function $H: \{0,1\}^* \rightarrow \{0,1\}^n$ to hash identity strings of arbitrary length to size n.

1. **Setup(1^k)**: Select $a \overset{\$}{\leftarrow} Z_q$, $g_2, u' \overset{\$}{\leftarrow} G$ and an n-length vector $\langle u \rangle$ consisting of elements $u_1, \ldots, u_n \in G$. Set $g_1 = g^a$ and publish master public key as $mpk = \langle G, G_T, e, g, g_1, u', \langle u \rangle, H \rangle$. The master secret key is $msk = g_2^a$.
2. **Set-User-Key(1^k, mpk)**: Select $s_{ID} \overset{\$}{\leftarrow} Z_q$ and set $upk_{ID} = \langle UPK_{1,ID}, UPK_{2,ID} \rangle =$

$\langle g^{s_{ID}}, g_1^{s_{ID}} \rangle$

3. **Partial-Private-Key-Extract($\boldsymbol{mpk}, \boldsymbol{msk}$, ID, $\boldsymbol{upk_{ID}}$)**: Parse **ID** as an n-bit identity string with d_i denoting the i-th bit of **ID**. Let $ID = \{1, \ldots, n\}$ be the set of all i in which $d_i = 1$. Define the Waters hash function as $U = (u' \prod_{i \in ID} u_i)$. Select $r \overset{\$}{\leftarrow} Z_q$ and construct the user partial private key as $ppk = \langle S_{ID}, R_{ID} \rangle = \langle g_2^a (U_{ID})^r, UPK_{1,ID}^r \rangle$.
4. Set-Private-Key($mpk, ID, s_{ID}, upk_{ID}, ppk_{ID}$):
First check if $e(S_{ID}, UPK_{1,ID}) = e(g_2, UPK_{2,ID}) e(U_{ID}, R_{ID})$. This is according to the equation:

$$\begin{aligned} e(S_{ID}, UPK_{1,ID}) &= e(g_2^a U_{ID}^r, g^{s_{ID}}) \\ &= e(g_2^a, g^{s_{ID}}) e(U_{ID}^r, g^{s_{ID}}) \\ &= e(g_2, g_1^{s_{ID}}) e(U_{ID}, g^{r s_{ID}}) \\ &= e(g_2, UPK_{2,ID}) e(U_{ID}, R_{ID}) \end{aligned} \tag{1}$$

If the partial private key is correct, then proceed to calculate $usk = \langle USK_{1,ID} = S_{ID}^{s_{ID}}, USK_{2,ID} = R_{ID} \rangle$
5. **Identification protocol** is run by **Prover($\boldsymbol{mpk}, \boldsymbol{ID}, \boldsymbol{upk_{ID}}, \boldsymbol{usk_{ID}}$)** and **Verifier**($\boldsymbol{mpk}, \boldsymbol{ID}, \boldsymbol{upk_{ID}}$) as such:
   a) **Prover** chooses a random $z \overset{\$}{\leftarrow} Z_q$, computes $X = e(U_{ID}, USK_{2,ID})^z, Y = g_2^z$ and sends $X, Y, USK_{2,ID}$ to **Verifier.**
   b) **Verifier** picks a random challenge $c \overset{\$}{\leftarrow} Z_q$ and sends it to **Prover**.
   c) **Prover** calculates $Z = USK_{1,ID}^{z+c}$ and sends Z as a response to **V**.

   V accepts if
   1) $e(g_1, UPK_{1,ID}) = e(UPK_{2,ID}, g)$ and
   2) $e(Z, g) = e(Y g_2^c, UPK_{1,ID}) X e(X U_{ID}^c, USK_{2,ID})$

To check for completeness:

$$\begin{aligned} e(g_1, UPK_{1,ID}) &= e(g^a, g^{s_{ID}}) \\ &= e(g^{a s_{ID}}, g) \\ &= e(UPK_{2,ID}, g) \end{aligned} \tag{2}$$

and

$$\begin{aligned} e(Z, g) &= e\left(S_{ID}^{s_{ID}(z+c)}, g\right) \\ &= e\left((g_2^a U_{ID}^r)^{s_{ID}(z+c)}, g\right) \\ &= e\left(g_2^{az+c} U_{ID}^{r s_{ID} z} U_{ID}^{r s_{ID} c}, g\right) \\ &= e(g_2^{az+c}, g) e\left(U_{ID}^{r s_{ID} z}, g\right) e\left(U_{ID}^{r s_{ID} c}, g\right) \\ &= e(g_2^z g_2^c, g^a) e(U_{ID}^z, g^{r s_{ID}}) e(U_{ID}^c, g^{r s_{ID}}) \\ &= e(Y g_2^c, g_1) X e\left(U_{ID}^c, USK_{2,ID}\right) \end{aligned} \tag{3}$$

4. Security Analysis

In this section, a security analysis on the certificateless identification scheme is presented. The scheme manages to achieve security against Super-Type-1 and Super-Type-2 adversaries for impersonation under passive attacks, and security against Strong-Type-1 and Strong-Type-2 adversaries for impersonation under active/concurrent attacks, all in the standard model.

4.1 Security Against Type-1 Impersonation under Passive Attacks

Theorem 1. *The certificateless identification scheme is* (t, q_I, ε)*-secure against Super-Type-1 impersonators under passive attack in the standard model if the CDHP is* (t', ε')*-hard where*

$$t' = t + O(\rho(2n(q_e) + \tau(q_I)), \tag{2}$$

$$\varepsilon \leq \sqrt{4q_e(n+1)\varepsilon'} + q^{-1} \tag{3}$$

where ρ *represents time taken to do a multiplication in* G, τ *is the time taken to do an exponentiation in* G, q_e *represents the number of extract queries made,* q_i *represents the number of transcript queries made and* $q_I = q_e + q_i$.

Proof. Suppose there exists an impersonator I_1 who (t, q_I, ε)-breaks the IBI scheme. Then we show an algorithm *M* which (t', ε')-breaks the CDH assumption by running I_1 as a subroutine. *M* is given a group G, a generator $g \in G$ and elements g^a, g^b. Without loss of generality, it can be assumed any **ExtrPartSK**, **RequestPK**, **ExtrFullSK** and **Identification** queries are preceded by a **CreateUser** query, while **Identification** and **ExtrFullSK** queries are preceded by a **RequestPK** query. *M* simulates the challenger for I_1 as follows:

1. **Setup(1^k).** Taking in the security parameter 1^k, *M* sets $l = 2q_e$ and randomly chooses $k \overset{\$}{\leftarrow} Z_n$. Assume that $l(n+1) < q$ for the given values of q_e and n. Furthermore, *M* randomly chooses $x' \overset{\$}{\leftarrow} Z_l$, a vector $\langle X \rangle$ of length n with $x_\iota \overset{\$}{\leftarrow} Z_l$ for all ι, a randomly selected $y' \overset{\$}{\leftarrow} Z_q$ and a vector $\langle y \rangle$ of length n with $y_\iota \overset{\$}{\leftarrow} Z_q$ for all ι. Define the following functions:

$$F(ID) = x' + \sum_{\iota \in ID} x_\iota - lk \tag{4}$$

$$J(ID) = y' + \sum_{\iota \in ID} y_\iota \tag{5}$$

M now sets $g_1 = g^a$ and $g_2 = g^b$. *M* also sets $u' = g_2^{x'-lk} g^{y'}$ and a vector $\langle u \rangle$ of length n consisting of n elements $u_\iota = g_2^{x_\iota} g^{y_\iota}$. *M* passes the system parameters mpk to I_1 as $\langle G, G_T, e, g, g_1, g_2, u', \langle u \rangle, H \rangle$ but does not have the master secret key $g_2^a = g^{ab}$. Note that with functions $F(ID)$ and $J(ID)$, we have:

$$U_{ID} = u' \prod_{\iota \in ID} u_\iota = g_2^{F(ID)} g^{J(ID)} \tag{6}$$

2. **CreateUser(ID_i) query:** For any **ExtrPartSK**, **RequestPK**,**ExtrFullSK** and **Identification** query, *M* first checks if the user ID_i is created. If not, *M* first selects $s_{ID_i} \overset{\$}{\leftarrow} Z_q$, pre-computes $U_{ID_i} = g_2^{F(ID_i)} g^{J(ID_i)}$ and the public key components $UPK_{1,ID_i} = g^{s_{ID_i}}$,$UPK_{2,ID_i} = g_1^{s_{ID_i}}$ and sets a flag $\varphi_{ID_i} = 1$ denoting that the public key is still the original for ID_i. *M* stores these values for future use.
3. **ExtrPartSK(ID_i) query.** When I_1 queries *M* for the partial private key of ID_i, *M* checks if $F(ID_i) = 0 \ mod \ l$ and aborts if it is. This is because given the assumption $l(n+1) < q$ implies $0 \leq lk \leq q$ and $0 \leq x' + \sum_{\iota \in ID_i} x_\iota \leq q$. Therefore $F(ID_i) = 0 \ mod \ q$ implies that $F(ID_i) = 0 \ mod \ l$ and the simulator aborts because it is unable to construct the

partial private key. Otherwise if $F(ID_i) \neq 0 \ mod \ l$, M constructs the partial private key by randomly selecting $r_{ID_i} \overset{\$}{\leftarrow} Z_q$ and computes the partial private key as: $\left(\tilde{S}_{ID_i} = g_1^{-\frac{J(ID_i)}{F(ID_i)}} \left(U_{ID_i}\right)^{r_{ID_i}}, \tilde{R}_{ID_i} = g_1^{-s_{ID_i}/F(ID_i)} g^{r_{ID_i} s_{ID_i}} \right)$

M returns $ppk_{ID_i} = \langle \tilde{S}_{ID_i}, \tilde{R}_{ID_i} \rangle$ to I_1.

4. **RequestPK(ID_i) query**. M finds $upk_{ID_i} = \langle UPK_{1,ID_i}, UPK_{2,ID_i} \rangle$ and returns it to I_1.
5. **ExtrFullSK(ID_i) query**. If $F(ID_i) = 0 \ mod \ l$ then M aborts the simulation. Otherwise M returns $usk_{ID_i} = \langle S_{ID_i}^{s_{ID_i}}, R_{ID_i} \rangle$ to I_1.
6. **ReplacePK($ID_i, upk'_{ID_i} = \langle UPK'_{1,ID_i}, UPK'_{2,ID_i} \rangle$) query.** If the new public key satisfies $e\left(g_1, UPK'_{1,ID_i}\right) = e(UPK'_{2,ID_i}, g)$, M then replaces $\langle UPK_{1,ID_i}, UPK_{2,ID_i} \rangle$ with $\langle UPK'_{1,ID_i}, UPK'_{2,ID_i} \rangle$ and sets $\varphi_{ID_i} = 0$. Note that the new corresponding secret value is not required here.
7. **Identification(ID_i) query.** I_1 will act as a cheating verifier to learn information from valid conversation transcripts from M. M retrieves ID_i's stored details to construct the transcript. Once again, note that $sv = s_{ID_i}$ is not necessary for passive attacks. M can create a valid transcript for each m-th query by picking $r_{ID_i} \overset{\$}{\leftarrow} Z_q$ (if not yet created) and $c_m, z_m \overset{\$}{\leftarrow} Z_q$. M sets the transcript as $\tilde{X}_m = e\left(U_{ID_i}, R_{ID_i}\right)^{z_m}, \tilde{Y}_m = g^{z_m} g_2^{-c_m}, \tilde{R}_{ID_i} = UPK_{1,ID_i}^{r_{ID_i}}, \widetilde{Z_J} = UPK_{2,ID_i}^{z_m} U_{ID_i}^{s_{ID_i} r_{ID_i}(z_m + c_m)}$ and passes the transcript to I_1. I_1 can check that this is a valid transcript since:

$$\begin{aligned} e\left(\widetilde{Z_m}, g\right) &= e\left(g_1^{s_{ID_i} z_m} U_{ID_i}^{s_{ID_i} r_{ID_i}(z_m + c_m)}, g\right) \\ &= e\left(g_1^{s_{ID_i} z_m}, g\right) e\left(U_{ID_i}^{s_{ID_i} r_{ID_i} z_m}, g\right) \left(e\left(U_{ID_i}^{s_{ID_i} r_{ID_i} c_m}, g\right)\right. \\ &= e\left(g^{z_m}, g_1^{s_{ID_i}}\right) e\left(U_{ID_i}^{z_m}, g^{s_{ID_i} r_{ID_i}}\right) e\left(U_{ID_i}^{c_m}, g^{s_{ID_i} r_{ID_i}}\right) \\ &= e\left(\frac{g^{z_m}}{g_2^{c_m}} g_2^{c_m}, g_1^{s_{ID_i}}\right) e\left(U_{ID_i}, R_{ID_i}\right)^{z_m} e\left(U_{ID_i}^{c_m}, R_{ID_i}\right) \\ &= e\left(\tilde{Y}_m g_2^{c_m}, UPK_{2,ID_i}\right) \tilde{X}_m e\left(U_{ID_i}^{c_m}, R_{ID_i}\right) \end{aligned} \tag{7}$$

Note that s_{ID_i} is not needed by M even for replaced public keys for ID_i while conducting **Identification** queries since the public keys need to hold for the verifier's first check equation $e\left(g_1, UPK_{1,ID_i}\right) = e(UPK_{2,ID_i}, g)$. In other words, the new public values of $UPK'_{1,ID_i}, UPK'_{2,ID_i}$ of all IDs must fulfill the check equation even with the replaced public keys, thus requiring I_1 to submit valid public key replacement values for **ReplacePK** queries.

Eventually I_1stops phase 1 and outputs the challenge identity, ID^*, on which it wishes to be challenged on. M checks if $F(ID^*) = 0 \ mod \ q$ then reports failure and aborts if not. Otherwise M runs I_1 now as a cheating prover on ID^*. M obtains (X, Y, R, c_1, z_1) then resets I_1 to its previous state where it just sent its commitment to obtain (X, Y, R, c_2, z_2). In both cases, it must hold that $e\left(g_1, UPK_{1,ID^*}\right) = e(UPK_{2,ID^*}, g)$ for all public values of

$UPK_{1,ID^*}, UPK_{2,ID^*}$ of ID^*. Based on the Reset Lemma [22], M is then able to extract the full private key as

$$\left(\frac{Z_1}{Z_2}\right)^{\frac{1}{(c_1-c_2)}} = \left(\frac{USK_{1,ID^*}^{z+c_1}}{USK_{1,ID^*}^{z+c_2}}\right)^{\frac{1}{(c_1-c_2)}} = \left(USK_{1,ID^*}^{c_1-c_2}\right)^{\frac{1}{(c_1-c_2)}} = USK_{1,ID^*} \tag{8}$$

By using the knowledge of exponent assumption from [19], M can either extract σ if $\langle UPK_{1,ID^*}, UPK_{2,ID^*}\rangle = \langle g^{\sigma}, g^{a\sigma}\rangle$ were generated from g, g^a, or extract ϱ if $\langle UPK_{1,ID^*}, UPK_{2,ID^*}\rangle = \langle g^{\sigma\varrho}, g^{a\sigma\varrho}\rangle$ were generated from $g^{\sigma}, g^{a\sigma}$.

For the first case, M calculates the solution to the CDH problem as:

$$\left(\frac{USK_{1,ID^*}}{R_{r_{ID^*}}^{J(ID^*)\sigma}}\right)^{\frac{1}{\sigma}} = \left(\frac{\left(g_2^a U_{ID^*}^{r_{ID^*}}\right)^{\sigma}}{g^{r_{ID^*}J(ID)\sigma}}\right)^{\frac{1}{\sigma}} = \left(\frac{g^{ab\sigma} g^{r_{ID^*}J(ID)\sigma}}{g^{r_{ID^*}J(ID)\sigma}}\right)^{\frac{1}{\sigma}} = (g^{ab\sigma})^{\frac{1}{\sigma}} = g^{ab} \tag{9}$$

For the second case, M calculates the solution to the CDH problem as:

$$\begin{aligned}\left(\frac{USK_{1,ID^*}}{R_{r_{ID^*}}^{J(ID^*)\varrho\sigma}}\right)^{\frac{1}{\varrho\sigma}} &= \left(\frac{\left(g_2^a U_{ID^*}^{r_{ID^*}}\right)^{\varrho\sigma}}{g^{r_{ID^*}J(ID)\varrho\sigma}}\right)^{\frac{1}{\sigma}} = \left(\frac{g^{ab\sigma} g^{r_{ID^*}J(ID)\varrho\sigma}}{g^{r_{ID^*}J(ID)\varrho\sigma}}\right)^{\frac{1}{\varrho\sigma}} \\ &= (g^{ab\varrho\sigma})^{\frac{1}{\varrho\sigma}} = g^{ab}\end{aligned} \tag{10}$$

It remains to calculate the probability of M solving the CDH problem and winning the game. The probability of M successfully extracting 2 valid transcripts from I_1 is given by $(\varepsilon - \frac{1}{q})^2$ as given by the Reset Lemma [21]. Upon extraction of USK_{1,ID^*}, M will be able to compute g^{ab}. We break down the probability of M winning the CDHP to:

$$\begin{aligned}\Pr[M\ wins\ CDHP] &= \Pr[M\ computes\ g^{ab} \wedge \neg abort] \\ &= \Pr[M\ computes\ g^{ab} | \neg abort]\ \Pr[\neg abort] \\ \varepsilon' &\geq (\varepsilon - q^{-1})^2 \Pr[\neg abort]\end{aligned} \tag{11}$$

Finally, calculate $\Pr[\neg abort]$. Define the following events:

1) Event A_i where M answers all queries $F(ID_i) \neq 0\ mod\ l$ and
2) Event A^* where I outputs the challenge identity ID^* where $F(ID) = 0\ mod\ q$.

Calculate the probability of A^* as:

$$\begin{aligned}\Pr[A^*] &= \Pr[F(ID^*) = 0\ mod\ q\ \vee F(ID^* = 0\ mod\ l\ ] \\ &= \Pr[F(ID^*) = 0\ mod\ l]\ \Pr[F(ID^*) = 0\ mod\ q\ |\ F(ID^*) = 0\ mod\ l] \\ &= \frac{1}{l}\left(\frac{1}{n+1}\right)\end{aligned} \tag{12}$$

Notice that:

$$\Pr[\bigcap_{i=1}^{q_e} A_i | A^*] = 1 - \Pr[\bigcup_{i=1}^{q_e} A_i | A^*] \tag{13}$$

$$= 1 - \sum_{i=1}^{q_e} \Pr[\neg A_i | A^*]$$
$$= 1 - \frac{q_e}{l}$$

Since $l = 2q_e$ in the simulation, therefore

$$\begin{aligned} \Pr[\neg abort] &= \left(1 - \frac{q_e}{l}\right)\left(\frac{1}{l}\right)\left(\frac{1}{n+1}\right) \\ &= \left(1 - \frac{q_e}{2q_e}\right)\left(\frac{1}{2q_e}\right)\left(\frac{1}{n+1}\right) \\ &= \left(\frac{1}{2}\right)\left(\frac{1}{2q_e}\right)\left(\frac{1}{n+1}\right) \\ &= \frac{1}{4q_e(n+1)} \end{aligned} \tag{14}$$

Putting them together, the advantage of *M* in breaking CDHP is:

$$\begin{aligned} \Pr[M\ computes\ g^{ab}] &\geq (\varepsilon - q^{-1})^2 \frac{1}{4q_e(n+1)} \\ \varepsilon' &\geq (\varepsilon - q^{-1})^2 \frac{1}{4q_e(n+1)} \\ \varepsilon &\leq \sqrt{4q_e(n+1)\varepsilon'} + q^{-1} \end{aligned} \tag{15}$$

■

4.2 Security Against Type-1 Active/Concurrent Attacks

Theorem 1. *The certificateless identification scheme is* (t, q_I, ε) *-secure against Strong-Type-1 impersonators under active/concurrent attack in the standard model if the OMCDHP is* $(t", q", \varepsilon")$*-hard where*

$$t' = t + O(\rho(2n(q_e) + \tau(q_I)), \tag{16}$$

$$\varepsilon \leq \sqrt{4q_e(n+1)\varepsilon"} + q^{-1} \tag{17}$$

where ρ *represents time taken to do a multiplication in* G, τ *is the time taken to do an exponentiation in* G, $-q_e$ *represents the number of extract queries made,* q_i *represents the number of transcript queries made and* $q_I = q_e + q_i$.

Proof. Define the following as the impersonation under active/concurrent attack (IMP-AA/CA-1) game. Assume that if the certificateless identification scheme is (t, q_I, ε) -breakable by an impersonator I_1, then we can show a simulator M that $(t", q", \varepsilon")$-breaks the OMCDHP. M is given a challenge oracle $CHALL$ which provides random points in $G_1$and a solution oracle CDH that upon an input h outputs h^a. In order to win the game, M has to provide the solutions to n queries to $CHALL$ by using strictly less queries to CDH. To begin, M is given $(g, g_1 = g^a)$. M then queries $CHALL$ for the initial challenge W_0 and runs the Type-1 impersonator I_1 as a subroutine. Without loss of generality, it can be assumed any **ExtrPartSK**, **RequestPK**, **ExtrFullSK** and **Identification** queries are preceded by a **CreateUser** query, while **Identification** and **ExtrFullSK** queries are preceded

by a **RequestPK** query. The way the environment is simulated for I_1 is similar to that of the IMP-PA-1 game, and hence only the differences are shown.

1. **Setup(1^k)** is similar to the one in the IMP-PA-1 game, except M sets $g_2 = W_0$ instead,
2. **CreateUser(ID_i) query** is similar to the one in the IMP-PA-1 game.
3. **ExtrPartSK(ID_i) query** is similar to the one in the IMP-PA-1 game.
4. **RequestPK(ID_i) query** is similar to the one in the IMP-PA-1 game.
5. **ExtrFullSK(ID_i) query** is similar to the one in the IMP-PA-1 game.
6. **ReplacePK($ID_i, upk'_{ID_i} = \langle UPK'_{1,ID_i}, UPK'_{2,ID_i} \rangle$) query** is similar to the one in the IMP-PA-1 game.
7. **Identification(ID_i) query.** This is where the game differs significantly from that of the IMP-PA-1 game. Once again, I_1 will act as a cheating verifier to learn information, but this time by requesting a valid conversation with M. M retrieves ID_i's stored details to conduct the conversation. If $\varphi = 0$, M then requests the new secret value $sv = s_{ID_i}$ and the replaced public key upk_{ID_i} from I_1. M starts with a counter $m = 0$ that is incremented every **Identification** query, then does the following:

   i) M queries $CHALL$ for a random W_m and additionally chooses $r_{ID_i} \overset{\$}{\leftarrow} Z_q$ (if not yet created) and $z_m \overset{\$}{\leftarrow} Z_q$. M sets $X_m = e(U_{ID_i}, R_{ID_i})^{z_m}, Y_m = W_m$ and $R_{ID_i} = UPK_{1,ID_i}^{r_{ID_i}} = g^{s_{ID_i} r_{ID_i}}$ and passes them as a commitment to to I_1.

   ii) I_1 selects a random challenge $c_m \overset{\$}{\leftarrow} Z_q$ and sends it back to M.

   iii) M responds by querying $(W_m W_0^{c_m})^{a s_{ID_j}} = CDH([W_m W_0^{c_m}]^{s_{ID_j}})$, sets $Z_m = (W_m W_0^{c_m})^{a s_{ID_j}} U_{ID_j}^{r_{ID_j} s_{ID_j} (z_m + c_m)}$ and sends it to I_1. One can check the validity of the second checking equation by:

$$\begin{aligned} e(\widetilde{Z_m}, g) &= e\left((W_m W_0^{c_m})^{a s_{ID_j}} U_{ID_j}^{r_{ID_j} s_{ID_j} (z_m + c_m)}, g\right) \\ &= e\left((W_m W_0^{c_m})^{a s_{ID_j}}, g\right) e(U_{ID_i}^{s_{ID_i} r_{ID_i} z_m}, g)\left(e(U_{ID_i}^{s_{ID_i} r_{ID_i} c_m}, g\right) \\ &= e\left(W_m W_0^{c_m}, g_1^{s_{ID_j}}\right) e\left(U_{ID_i}^{z_m}, g^{s_{ID_i} r_{ID_i}}\right) e\left(U_{ID_i}^{c_m}, g^{s_{ID_i} r_{ID_i}}\right) \\ &= e\left(\tilde{Y}_m g_2^{c_m}, UPK_{2,ID_i}\right) \tilde{X}_m e\left(U_{ID_i}^{c_m}, R_{ID_i}\right) \end{aligned} \tag{18}$$

   Additionally the verifier's first check equation $e(g_1, UPK_{1,ID_i}) = e(UPK_{2,ID_i}, g)$ for public values of $UPK'_{1,ID_i}, UPK'_{2,ID_i}$ of all IDs must hold even with the replaced public keys.

Eventually I_1 stops phase 1 and outputs the challenge identity, ID^*, on which it wishes to be challenged on. M checks if $F(ID^*) = 0 \bmod q$ then reports failure and aborts if not. Otherwise M runs I_1 now as a cheating prover on ID^*. M obtains (X, Y, R, c_1, z_1) then resets I_1 to its previous state where it just sent its commitment to obtain (X, Y, R, c_2, z_2). In both cases, it must hold that $e(g_1, UPK_{1,ID^*}) = e(UPK_{2,ID^*}, g)$ for all public values of $UPK_{1,ID^*}, UPK_{2,ID^*}$ of ID^*. Based on the Reset Lemma [22], M is then able to extract the full private key as

$$\left(\frac{Z_1}{Z_2}\right)^{\frac{1}{(c_1-c_2)}} = \left(\frac{USK_{1,ID^*}^{z+c_1}}{USK_{1,ID^*}^{z+c_2}}\right)^{\frac{1}{(c_1-c_2)}} = \left(USK_{1,ID^*}^{c_1-c_2}\right)^{\frac{1}{(c_1-c_2)}} = USK_{1,ID^*} \tag{19}$$

By using the knowledge of exponent assumption from [19], M can either extract σ if $\langle UPK_{1,ID^*}, UPK_{2,ID^*}\rangle = \langle g^{\sigma}, g^{a\sigma}\rangle$ were generated from g, g^a, or extract ϱ if $\langle UPK_{1,ID^*}, UPK_{2,ID^*}\rangle = \langle g^{\sigma\varrho}, g^{a\sigma\varrho}\rangle$ were generated from $g^{\sigma}, g^{a\sigma}$.

For the first case, M calculates the solution to the CDH problem as:

$$\left(\frac{USK_{1,ID^*}}{R_{r_{ID^*}}^{J(ID^*)\sigma}}\right)^{\frac{1}{\sigma}} = \left(\frac{\left(W_0^a U_{ID^*}^{r_{ID^*}}\right)^{\sigma}}{g^{r_{ID^*}J(ID)\sigma}}\right)^{\frac{1}{\sigma}} = \left(\frac{W_0^{a\sigma} g^{r_{ID^*}J(ID)\sigma}}{g^{r_{ID^*}J(ID)\sigma}}\right)^{\frac{1}{\sigma}} = (W_0^{a\sigma})^{\frac{1}{\sigma}} = W_0^a \tag{20}$$

For the second case, M calculates the solution to the CDH problem as:

$$\left(\frac{USK_{1,ID^*}}{R_{r_{ID^*}}^{J(ID^*)\varrho\sigma}}\right)^{\frac{1}{\varrho\sigma}} = \left(\frac{\left(W_0^a U_{ID^*}^{r_{ID^*}}\right)^{\varrho\sigma}}{g^{r_{ID^*}J(ID)\varrho\sigma}}\right)^{\frac{1}{\sigma}} = \left(\frac{W_0^{a\varrho\sigma} g^{r_{ID^*}J(ID)\varrho\sigma}}{g^{r_{ID^*}J(ID)\varrho\sigma}}\right)^{\frac{1}{\varrho\sigma}} = (W_0^{a\varrho\sigma})^{\frac{1}{\varrho\sigma}} = W_0^a \tag{21}$$

Recall that s_{ID_i} is provided by I_1 for every m-th **Identification** query for the corresponding public key of ID_i being used, both for original or replaced. M then proceeds to calculate the solutions for the challenges $W_1, \ldots, W_m$ as:

$$\left(\frac{Z_m}{W_0^{as_{ID_i}c_m} g^{J(ID_i)r_{ID_i}s_{ID_i}(z_m+c_m)}}\right)^{\frac{1}{s_{ID_i}}} = \left(\frac{W_m^{as_{ID_i}} W_0^{as_{ID_i}c_m} g^{J(ID_i)r_{ID_i}s_{ID_i}(z_m+c_m)}}{W_0^{as_{ID_i}c_m} g^{J(ID_i)r_{ID_i}s_{ID_i}(z_m+c_m)}}\right)^{\frac{1}{s_{ID_i}}} = W_m^a \tag{22}$$

The probability of M winning the OMCDHP is the same as in the IMP-PA-1 game, except that ε', the advantage of M in solving the CDH problem is substituted with ε", the advantage of M in solving the OMCDHP game.

■

4.3 Security Against Type-2 Impersonation under Passive Attacks

Theorem 1. *The certificateless identification scheme is* (t, q_I, ε) *-secure against Strong-Type-2 impersonators under passive attack in the standard model if the CDHP is* (t', ε')*-hard where*

$$t' = t + O(\rho(2n(q_e) + \tau(q_I)), \tag{23}$$

$$\varepsilon \leq \sqrt{4q_e(n+1)\varepsilon'} + q^{-1} \tag{24}$$

where ρ *represents time taken to do a multiplication in* G, τ *is the time taken to do an*

exponentiation in G , q_e represents the number of extract queries made, q_i represents the number of transcript queries made and $q_I = q_e + q_i$.

Proof. Suppose there exists an impersonator I_2 who (t, q_I, ε)-breaks the IBI scheme. Then we show an algorithm *M* which (t', ε')-breaks the CDH problem by running I_2 as a subroutine. *M* is given a group G, a generator $g \in G$ and, to keep with the consistency of scheme definitions, elements defined as g^s, g^b. Without loss of generality, it can be assumed any **RequestPK**, **ExtrFullSK** and **Identification** queries are preceded by a **CreateUser** query, while **Identification** and **ExtrFullSK** queries are preceded by a **RequestPK** query. *M* simulates the challenger for I_2 as follows:

1. **Setup(1^k).** Taking in the security parameter 1^k, *M* sets $l = 2q_e$ and randomly chooses $k \xleftarrow{\$} Z_n$. Assume that $l(n+1) < q$ for the given values of q_e and n. Furthermore, *M* randomly chooses $x' \xleftarrow{\$} Z_l$, a vector $\langle X \rangle$ of length n with $x_\iota \xleftarrow{\$} Z_l$ for all ι, a randomly selected $y' \xleftarrow{\$} Z_q$ and a vector $\langle y \rangle$ of length n with $y_\iota \xleftarrow{\$} Z_q$ for all ι. Define the following functions:

$$F(ID) = x' + \sum_{\iota \in ID} x_\iota - lk \tag{25}$$

$$J(ID) = y' + \sum_{\iota \in ID} y_\iota \tag{26}$$

   M now selects $a \xleftarrow{\$} Z_q$, sets $g_1 = g^a$ and $g_2 = g^b$. *M* also sets $u' = g_2^{x'-lk} g^{y'}$ and a vector $\langle u \rangle$ of length n consisting of n elements $u_\iota = g_2^{x_\iota} g^{y_\iota}$. *M* passes the system parameters mpk to I_2 as $\langle G, G_T, e, g, g_1, g_2, u', \langle u \rangle, H \rangle$ but does not have the master secret key $g_2^a = g^{ab}$. Note that with functions $F(ID)$ and $J(ID)$, we have:

$$U_{ID} = u' \prod_{\iota \in ID} u_\iota = g_2^{F(ID)} g^{J(ID)} \tag{27}$$

   In Type-2 games, *M* knows the master secret key $g_2^a = g^{ab}$ and is able to create partial private keys, therefore **ExtrPartSK** queries are not necessary.
2. **CreateUser(ID_i) query:** For any **RequestPK**,**ExtrFullSK** and **Identification** query, *M* first checks if the user ID_i is created. If $F(ID_i) = 0 \; mod \; l$, *M* sets $UPK_{1,ID_i} = g^s$ and $UPK_{2,ID_i} = g_1^s$. If not, *M* first selects $s_{ID_i} \xleftarrow{\$} Z_q$, pre-computes $U_{ID_i} = g_2^{F(ID_i)} g^{J(ID_i)}$ and the public key components $UPK_{1,ID_i} = g^{s_{ID_i}}$, $UPK_{2,ID_i} = g_1^{s_{ID_i}}$ and sets a flag $\varphi_{ID_i} = 1$ denoting that the public key is still the original for ID_i. *M* stores these values for future use.
3. **RequestPK(ID_i) query**. *M* finds $upk_{ID_i} = \langle UPK_{1,ID_i}, UPK_{2,ID_i} \rangle$ and returns it to I_2.
4. **ExtrFullSK(ID_i) query**. If $F(ID_i) = 0 \; mod \; l$ then *M* aborts the simulation. Otherwise, *M* retrieves s_{ID_i}, selects $r_{ID_i} \xleftarrow{\$} Z_q$ and calculates $USK_{1,ID_i} = g_1^{\frac{-J(ID_i)s_{ID_i}}{F(ID_i)}} U_{ID_i}^{r_{ID_i} s_{ID_i}}$ and $USK_{2,ID_i} = g_1^{\frac{-s_{ID_i}}{F(ID_i)}} g^{r_{ID_i} s_{ID_i}}$. *M* returns $usk_{ID_i} = \langle S_{ID_i}^{s_{ID_i}}, R_{ID_i} \rangle$ to I_2.

5. **ReplacePK($ID_i, upk'_{ID_i} = \langle UPK'_{1,ID_i}, UPK'_{2,ID_i} \rangle$) query.** For $F(ID_i) \neq 0 \; mod \; l$, M checks if the new public key satisfies $e(g_1, UPK'_{1,ID_i}) = e(UPK'_{2,ID_i}, g), \; and$ replaces $\langle UPK_{1,ID_i}, UPK_{2,ID_i} \rangle$ with $\langle UPK'_{1,ID_i}, UPK'_{2,ID_i} \rangle$ and sets $\varphi_{ID_i} = 0$ if true. Note that the new corresponding secret value is not required here. Otherwise for $F(ID_i) = 0 \; mod \; l$, M replies with $\perp$ since I_2 is not allowed to replace the challenge identity's public key.
6. **Identification(ID_i) query.** I_2 will act as a cheating verifier to learn information from valid conversation transcripts from M. M retrieves ID_i's stored details to construct the transcript. For Strong-Type-2 attacks, I_2 needs to supply $sv = s_{ID_i}$ for replaced public keys. M can create a valid transcript for each of the following cases:

a) If $F(ID_i) = 0 \; mod \; l$, then $U_{ID_i} = g^{J(ID_i)}$ and M picks $r_{ID_i} \overset{\$}{\leftarrow} Z_q$ (if not yet created) and $c_m, z_m \overset{\$}{\leftarrow} Z_q$, sets the transcript as $\tilde{X}_m = e(U_{ID_i}, R_{ID_i})^{z_m}, \tilde{Y}_m = g^{z_m} g_2^{-c_m}, \tilde{R}_{ID_i} = UPK_{1,ID_i}^{r_{ID_i}}, \widetilde{Z_m} = UPK_{2,ID_i}^{z_m} UPK_{1,ID_i}^{J(ID_i) r_{ID_i}(z_m + c_m)}$ and passes the transcript to I_2. I_2 can check that this is a valid transcript since:

$$\begin{aligned} e(\widetilde{Z_m}, g) &= e\left(UPK_{2,ID_i}^{z_m} UPK_{1,ID_i}^{J(ID_i) r_{ID_i}(z_m + c_m)}, g\right) \\ &= e\left(g^{asz_m} g^{sJ(ID_i) r_{ID_i} z_m} g^{sJ(ID_i) c_m}, g\right) \\ &= e\left(g^{asz_m} g^{sJ(ID_i) r_{ID_i} z_m} g^{sJ(ID_i) c_m}, g\right) \\ &= e(g^{z_m}, g^{as}) e\left(g^{J(ID_i) z_m}, g^{s r_{ID_i}}\right) e\left(g^{J(ID_i) c_m}, g^{s r_{ID_i}}\right) \\ &= e\left(\frac{g^{z_m}}{g_2^{c_m}} g_2^{c_m}, g^{as}\right) e\left(U_{ID_i}, R_{ID_i}\right)^{z_m} e\left(U_{ID_i}^{c_m}, R_{ID_i}\right) \\ &= e\left(\tilde{Y}_m g_2^{c_m}, UPK_{2,ID_i}\right) \tilde{X}_m e\left(U_{ID_i}^{c_m}, R_{ID_i}\right) \end{aligned} \tag{28}$$

b) If $F(ID_i) \neq 0 \; mod \; l$, then $U_{ID_i} = g_2^{F(ID_i)} g^{J(ID_i)}$. For this case M can build the full private key for any user other than those with $F(ID_i) = 0 \; mod \; l$ since it has access to the msk and s_{ID_i}. If $\varphi = 0$ then M requests the new secret value $sv = s_{ID_i}$ for the replaced public key from I_2. M then constructs the transcript by picking $r_{ID_i} \overset{\$}{\leftarrow} Z_q$ (if not yet created), $c_m, z_m \overset{\$}{\leftarrow} Z_q$ and setting $\tilde{X}_m = e(U_{ID_i}, R_{ID_i})^{z_m}$, $\tilde{Y}_m = g^{z_m}$, $R_{ID_i} = UPK_{1,ID_i}^{r_{ID_i}}, \widetilde{Z_m} = \left(g_2^a U_{ID_i}^{r_{ID_i}}\right)^{s_{ID_i}(z_m + c_m)}$ as the transcript. Once again, I_2 can check the second verifying equation:

$$\begin{aligned} e(\tilde{Z}_m, g) &= e\left(\left(g_2^a \left[g_2^{F(ID_j)} g^{J(ID_i)}\right]^{r_{ID_i}}\right)^{s_{ID_i}(z_m + c_m)}, g\right) \\ &= e\left(g_2^{as_{ID_i}(z_m + c_m)} \left[g_2^{F(ID_j)} g^{J(ID_i)}\right]^{r_{ID_i} s_{ID_i} z_m} \left[g_2^{F(ID_j)} g^{J(ID_i)}\right]^{r_{ID_i} s_{ID_i} c_m}, g\right) \\ &= e\left(g_2^{as_{ID_i}(z_m + c_m)}, g\right) e\left(U_{ID_i}^{s_{ID_i} r_{ID_i} z_m}, g\right) e\left(U_{ID_i}^{s_{ID_i} r_{ID_i} c_m}, g\right) \\ &= e\left(\frac{g^{z_m}}{g_2^{c_m}} g_2^{c_m}, g^{as}\right) e\left(U_{ID_i}, R_{ID_i}\right)^{z_m} e\left(U_{ID_i}^{c_m}, R_{ID_i}\right) \\ &= e\left(\tilde{Y}_m g_2^{c_m}, UPK_{2,ID_i}\right) \tilde{X}_m e\left(U_{ID_i}^{c_m}, R_{ID_i}\right) \end{aligned} \tag{29}$$

Note that for both cases the public keys need to hold for the verifier's first check equation $e(g_1, UPK_{1,ID_i}) = e(UPK_{2,ID_i}, g)$. In other words, the new public values of

$UPK'_{1,ID_i}, UPK'_{2,ID_i}$ of all IDs must fulfill the check equation even with the replaced public keys, thus requiring I_2 to submit valid public key replacement values for **ReplacePK** queries and a valid sv for **Identification** queries.

Eventually I_2 stops phase 1 and outputs the challenge identity, ID^*, on which it wishes to be challenged on. M checks if $F(ID^*) = 0 \ mod \ q$ then reports failure and aborts if not. Otherwise M runs I_2 now as a cheating prover on ID^*. M obtains (X, Y, R, c_1, z_1) then resets I_2 to its previous state where it just sent its commitment to obtain (X, Y, R, c_2, z_2). In both cases, it must hold that $e(g_1, UPK_{1,ID^*}) = e(UPK_{2,ID^*}, g)$ for all public values of $UPK_{1,ID^*}, UPK_{2,ID^*}$ of ID^*. Based on the Reset Lemma [22], M is then able to extract the full private key as

$$\left(\frac{Z_1}{Z_2}\right)^{\frac{1}{(c_1-c_2)}} = \left(\frac{USK_{1,ID^*}^{z+c_1}}{USK_{1,ID^*}^{z+c_2}}\right)^{\frac{1}{(c_1-c_2)}} = \left(USK_{1,ID^*}^{c_1-c_2}\right)^{\frac{1}{(c_1-c_2)}} = USK_{1,ID^*} \tag{30}$$

Since I_2 is not allowed to replace the public key of ID^*, M calculates the solution to the CDH problem as:

$$\left(\frac{USK_{1,ID^*}}{R^{J(ID^*)s}}\right)^{\frac{1}{a}} = \left(\frac{\left(g_2^a U_{ID^*}^{r_{ID^*}}\right)^s}{g^{r_{ID^*}J(IID^*)s}}\right)^{\frac{1}{a}} = \left(\frac{g^{abs} g^{r_{ID^*}J(ID^*)s}}{g^{r_{ID^*}J(ID^*)s}}\right)^{\frac{1}{a}} = (g^{abs})^{\frac{1}{a}} = g^{bs} \tag{31}$$

For the second case, M calculates the solution to the CDH problem as:

It remains to calculate the probability of M solving the CDH problem and winning the game. The probability of M successfully extracting 2 valid transcripts from I_1 is given by $(\varepsilon - \frac{1}{q})^2$ as given by the Reset Lemma [21]. Upon extraction of USK_{1,ID^*}, M will be able to compute g^{bs}. We break down the probability of M winning the CDHP to:

$$\begin{aligned} \Pr[M \ wins \ CDHP] &= \Pr[M \ computes \ g^{bs} \wedge \neg abort] \\ &= \Pr[M \ computes \ g^{bs} | \neg abort] \Pr[\neg abort] \\ \varepsilon' &\geq (\varepsilon - q^{-1})^2 \Pr[\neg abort] \end{aligned} \tag{32}$$

Finally, calculate $\Pr[\neg abort]$. Define the following events:

3) Event A_i where M answers all queries $F(ID_i) \neq 0 \ mod \ l$ and
4) Event A^* where I outputs the challenge identity ID^* where $F(ID) = 0 \ mod \ q$.

Calculate the probability of A^* as:

$$\begin{aligned} \Pr[A^*] &= \Pr[F(ID^*) = 0 \ mod \ q \ \vee F(ID^* = 0 \ mod \ l \ ] \\ &= \Pr[F(ID^*) = 0 \ mod \ l] \Pr[F(ID^*) = 0 \ mod \ q \mid F(ID^*) = 0 \ mod \ l] \\ &= \frac{1}{l}\left(\frac{1}{n+1}\right) \end{aligned} \tag{33}$$

Notice that:

$$\begin{aligned}\Pr[\bigcap_{i=1}^{q_e} A_i|A^*] &= 1 - \Pr[\bigcup_{i=1}^{q_e} A_i|A^*] \\ &= 1 - \sum_{i=1}^{q_e} \Pr[\neg A_i|A^*] \\ &= 1 - \frac{q_e}{l}\end{aligned} \tag{34}$$

Since $l = 2q_e$ in the simulation, therefore

$$\begin{aligned}\Pr[\neg abort] &= \left(1 - \frac{q_e}{l}\right)\left(\frac{1}{l}\right)\left(\frac{1}{n+1}\right) \\ &= \left(1 - \frac{q_e}{2q_e}\right)\left(\frac{1}{2q_e}\right)\left(\frac{1}{n+1}\right) \\ &= \left(\frac{1}{2}\right)\left(\frac{1}{2q_e}\right)\left(\frac{1}{n+1}\right) \\ &= \frac{1}{4q_e(n+1)}\end{aligned} \tag{35}$$

Putting them together, the advantage of *M* in breaking CDHP is:

$$\begin{aligned}\Pr[M\ computes\ g^{bs}] &\geq (\varepsilon - q^{-1})^2 \frac{1}{4q_e(n+1)} \\ \varepsilon' &\geq (\varepsilon - q^{-1})^2 \frac{1}{4q_e(n+1)} \\ \varepsilon &\leq \sqrt{4q_e(n+1)\varepsilon'} + q^{-1}\end{aligned} \tag{36}$$

■

4.4 Security Against Type-2 Active/Concurrent Attacks

Theorem 1. *The certificateless identification scheme is* (t, q_I, ε) *-secure against Strong-Type-2 impersonators under active/concurrent attack in the standard model if the OMCDHP is* $(t", q", \varepsilon")$*-hard where*

$$t' = t + O(\rho\big(2n(q_e) + \tau(q_I)\big), \tag{37}$$

$$\varepsilon \leq \sqrt{4q_e(n+1)\varepsilon"} + q^{-1} \tag{38}$$

where ρ *represents time taken to do a multiplication in* G, τ *is the time taken to do an exponentiation in* G, q_e *represents the number of extract queries made,* q_i *represents the number of transcript queries made and* $q_I = q_e + q_i$.

Proof. Define the following game as the impersonation under active/concurrent attack (IMP-AA/CA-2) game. Assume that the certificateless identification scheme is (t, q_I, ε) -breakable by an impersonator I_2, then we can show a simulator M that $(t", q", \varepsilon")$-breaks the OMCDHP. M is given a challenge oracle $CHALL$ which provides random points in $G_1$and a solution oracle CDH that upon input h outputs h^a. M has to provide the solutions to n queries to $CHALL$ by using strictly less queries to CDH in order to win the game. To begin, M is given $(g, g_1 = g^a)$. M then queries $CHALL$ for the initial challenge W_0

and runs the Type-2 impersonator I_2 as a subroutine. Without loss of generality, it can be assumed any **RequestPK**, **ExtrFullSK** and **Identification** queries are preceded by a **CreateUser** query, while **Identification** and **ExtrFullSK** queries are preceded by a **RequestPK** query. The way the environment is simulated for I_2 is similar to that of the IMP-PA-2 game, and hence only the differences are shown.

1. **Setup(1^k)** is similar to the one in the IMP-PA-2 game.
2. **CreateUser(ID_i) query** is similar to the one in the IMP-PA-2 game, except for $F(ID_i) = 0 \, mod \, l$, M sets $UPK_{1,ID_i} = W_0$ instead.
3. **RequestPK(ID_i) query** is similar to the one in the IMP-PA-2 game.
4. **ExtrFullSK(ID_i) query** is similar to the one in the IMP-PA-2 game.
5. **ReplacePK($ID_i, upk'_{ID_i} = \langle UPK'_{1,ID_i}, UPK'_{2,ID_i} \rangle$) query** is similar to the one in the IMP-PA-2 game.
6. **Identification(ID_i) query.** This is where the game differs significantly from the one in the IMP-PA-2 game. Once again, I_2 will act as a cheating verifier but this time by requesting a valid conversation with M. M retrieves ID_i's stored details to conduct the conversation. There are two cases to consider:
   a) If $F(ID_i) = 0 \, mod \, l$, then $U_{ID_i} = g^{J(ID_i)}$. M starts with a counter $m = 0$ that is incremented every **Identification** query for $F(ID_i) = 0 \, mod \, l$, then does the following:
      i) Mqueries $CHALL$ for a random W_m and additionally chooses $r_{ID_i} \overset{\$}{\leftarrow} Z_q$ (if not yet created) and $z_m \overset{\$}{\leftarrow} Z_q$. M sets $X_m = e(U_{ID_i}, R_{ID_i})^{z_m}, Y_m = W_m$ and $R_{ID_i} = UPK_{1,ID_i}^{r_{ID_i}}$ and passes them as a commitment to to I_2.
      ii) I_2 selects a random challenge $c_m \overset{\$}{\leftarrow} Z_q$ and sends it back to M.
      iii) M responds by querying $\left(W_m^a W_0^{ac_m} g^{J(ID_i) r_{ID_j (z_m + c_m)}}\right)^s = CDH(W_m^a W_0^{ac_m} g^{J(ID_i) r_{ID_j (z_m + c_m)}})$, sets $Z_m = \left(W_m^a W_0^{ac_m} g^{J(ID_i) r_{ID_j (z_m + c_m)}}\right)^s$ and sends it to I_2. I_2 can check the validity of the second checking equation by:

$$\begin{aligned} e(\widetilde{Z_m}, g) &= e\left(\left(W_m^a W_0^{ac_m} g^{J(ID_i) r_{ID_i (z_m + c_m)}}\right)^s, g\right) \\ &= e\left(\left(W_m^a W_0^{ac_m}\right)^s \left(g^{sJ(ID_i) r_{ID_i} z_m}\right)\left(g^{sJ(ID_i) r_{ID_i} c_m}\right), g\right) \\ &= e(W_m^{as} W_0^{asc_m}, g) e\left(\left(g^{sJ(ID_i) r_{ID_i} z_m}\right), g\right) e\left(g^{sJ(ID_i) r_{ID_i} c_m}, g\right) \\ &= e(W_m W_0^{c_m}, g^{as}) e\left(U_{ID_i}, g^{s r_{ID_i}}\right)^{z_m} e\left(U_{ID_i}^{c_m}, g^{s r_{ID_i}}\right) \\ &= e(Y_m g_2^{c_m}, UPK_{1,ID_i}) X_m e\left(U_{ID_i}^{c_m}, USK_{2,ID_i}\right) \end{aligned} \quad (39)$$

   b) If $F(ID_i) = 0 \, mod \, l$, then $U_{ID_i} = g_2^{F(ID_i)} g^{J(ID_i)}$. If $\varphi = 0$ then M requests the new secret value $sv = s_{ID_i}$ and the replaced public key upk'_{ID_i} from I_2, otherwise M just retrieves the original s_{ID_i} value. Since M can then construct the full usk_{ID_i}, M just builds $USK_{1,ID_i} = \left(g_2^a \left[g_2^{F(ID_i)} g^{J(ID_i)}\right]^{r_{ID_i}}\right)^{s_{ID_i}}$ and $USK_{2,ID_i} = g^{s_{ID_i} r_{ID_i}}$ and uses it to run the **Identification** query. For both cases of φ, the check equation $e(g, UPK_{2,ID_i}) = e(g_1, UPK_{1,ID_i})$ must hold.

Eventually I_2 stops phase 1 and outputs the challenge identity, ID^*, on which it wishes to be challenged on. M checks if $F(ID^*) = 0 \, mod \, q$ then reports failure and aborts if not. Otherwise M runs I_2 now as a cheating prover on ID^*. M obtains (X, Y, R, c_1, z_1) then resets I_2 to its previous state where it just sent its commitment to obtain (X, Y, R, c_2, z_2). In both cases, it must hold that $e(g_1, UPK_{1,ID^*}) = e(UPK_{2,ID^*}, g)$ for all public values of $UPK_{1,ID^*}, UPK_{2,ID^*}$ of ID^*. Based on the Reset Lemma [22], M is then able to extract the full private key as

$$\left(\frac{Z_1}{Z_2}\right)^{\frac{1}{(c_1-c_2)}} = \left(\frac{USK_{1,ID^*}^{z+c_1}}{USK_{1,ID^*}^{z+c_2}}\right)^{\frac{1}{(c_1-c_2)}} = \left(USK_{1,ID^*}^{c_1-c_2}\right)^{\frac{1}{(c_1-c_2)}} = USK_{1,ID^*} \tag{40}$$

Since I_2 is not allowed to replace the public key of ID^*, M calculates the solution to the OMCDHP as:

$$\left(\frac{USK_{1,ID^*}}{R^{J(ID^*)s}}\right)^{\frac{1}{a}} = \left(\frac{\left(W_0^a U_{ID^*}^{r_{ID^*}}\right)^s}{g^{r_{ID^*}J(ID)s}}\right)^{\frac{1}{a}} = \left(\frac{W_0^{as} g^{r_{ID^*}J(ID)s}}{g^{r_{ID^*}J(ID)s}}\right)^{\frac{1}{a}} = (W_0^{as})^{\frac{1}{a}} = W_0^s \tag{41}$$

M then proceeds to calculate the solutions for the challenges $W_1, \ldots, W_m$ as:

$$\begin{aligned} &\left(\frac{Z_m}{W_0^{asc_m} g^{J(ID_i) r_{ID_i} s(z_m + c_m)}}\right)^{\frac{1}{a}} \\ &= \left(\frac{W_m^{as} W_0^{asc_m} g^{J(ID_i) r_{ID_i} s(z_m + c_m)}}{W_0^{asc_m} g^{J(ID_i) r_{ID_i} s(z_m + c_m)}}\right)^{\frac{1}{a}} \\ &= W_m^s \end{aligned} \tag{42}$$

The probability of M winning the OMCDHP is the same as IMP-PA-2 game, except that ε', the advantage of M in solving the CDH problem is substituted with ε", the advantage of M in solving the OMCDHP game.

■

5. Efficiency Analysis

We give the operational cost of the certificateless identification scheme in **Table 1**.

Table 1. Operation Costs for the Certificateless Identification Scheme

Algorithm	Multiplication in G_1	Exponentiation in G_1	Multiplication in G_2	Exponentiation in G_2	Pairing
Setup	0	2	0	0	0
Set-User-Key	0	2	0	0	0
PPK-Extract	$n+2$	3	0	0	0
Set-Priv-Key	$n+1$	1	1	0	3
Prover	$n+1$	2	0	1	1
Verifier	$n+2$	2	2	0	5

Since the certificateless identification scheme is constructed based on an extension of the identity-based identification scheme from [23] to the certificateless setting, similar pre-computations are able to be conducted in order to reduce operation costs.

One can pre-compute the value of $\tilde{U} = e(U_{ID}, USK_{2,ID})$ beforehand, since this value is fixed, then calculate $X = \tilde{U}^z$ for **Prover** each time the protocol is run. This will reduce up to $n + 1$ times of multiplication in G_1 for both **Prover** and **Verifier,** and one pairing operation on **Prover**.
Another pre-computation operation available is to pre-compute and store $U = (u' \prod_{\iota \in ID} u_\iota)$ within **Prover**. This can later be sent as part of the commitment to **Verifier** so that **Verifier** does not require a second calculation. This saves another $n + 1$ multiplications in G_1 for **Verifier**.
The operation costs of **Prover** and **Verifier** with pre-computation is given in **Table 2**.

Table 2. Operation Costs for the Identification Protocol with Pre-computation

Algorithm	Multiplication in G_1	Exponentiation in G_1	Multiplication in G_2	Exponentiation in G_2	Pairing
Prover	0	2	0	1	0
Verifier	1	2	2	0	5

6. Conclusion

In this paper, we proposed a certificateless identification scheme with provable security in the standard model. This scheme provides a stronger security guarantee due to its non-reliance on the existence of random oracles. It is also the first ceritficateless identification scheme to have provable security in the standard model. The scheme is provable secure against both Type-1 and Type-2 impersonators, both passive and active/concurrent alike assuming the CDHP and OMCDHP is hard. It is secure against Super-Type-1 and Strong-Type-2 adversaries with regard to passive adversaries and secure against Strong-Type-1 and Strong-Type-2 adversaries with regard to active/concurrent security.

One interesting problem is to increase the security even more to propose a certificateless identification scheme provable secure in the standard model against Super-Type adversaries for active/concurrent attacks. Another direction the research on certificateless identification can take is to apply formal methods for proving certificateless identification schemes secure.

Acknowledgement

The authors would like to thank Ministry of Higher Education for aiding this research financially through the Exploratory Research Grant Scheme (ERGS/1/2011/PK/MMU/03/1) and the Fundamental Research Grant Scheme (FRGS/2/2013/ICT07/MMU/03/5). We also thank the anonymous reviewers for their kind comments.

References

[1] Shamir, A.. “Identity-based cryptosystems and signature schemes,” *Advances in cryptology*, Springer Berlin Heidelberg, pp. 47-53, January, 1985. Article (CrossRef Link)
[2] Al-Riyami, S. S., & Paterson, K. G., “Certificateless public key cryptography,” *Advances in Cryptology-ASIACRYPT 2003*, Springer Berlin Heidelberg, pp. 452-473, 2003. Article (CrossRef Link)
[3] Bellare, M., Namprempre, C., & Neven, G., “Security proofs for identity-based identification and signature schemes,” *Journal of Cryptology*, 22(1), 1-61, 2009. Article (CrossRef Link)
[4] Kurosawa, K., & Heng, S. H., “From digital signature to ID-based identification/signature” *Public Key Cryptography–PKC 2004*, Springer Berlin Heidelberg, pp. 248-261, 2004. Article (CrossRef Link)

[5] Kurosawa, K., & Heng, S. H., "Identity-based identification without random oracles," *Computational Science and Its Applications–ICCSA 2005*, Springer Berlin Heidelberg, pp. 603-613, 2005. Article (CrossRef Link)
[6] Kurosawa, K., & Heng, S. H., "The power of identification schemes," *Public Key Cryptography-PKC 2006*, Springer Berlin Heidelberg, pp. 364-377, 2006. Article (CrossRef Link)
[7] Yang, G., Chen, J., Wong, D. S., Deng, X., & Wang, D., "A new framework for the design and analysis of identity-based identification schemes," *Theoretical Computer Science*, 407(1), 370-388, 2008. Article (CrossRef Link)
[8] Chin, J. J., Heng, S. H., & Goi, B. M., "An efficient and provable secure identity-based identification scheme in the standard model," *Public Key Infrastructure*, Springer Berlin Heidelberg, pp. 60-73, 2008. Article (CrossRef Link)
[9] Chin, J. J., Heng, S. H., & Goi, B. M., "Hierarchical identity-based identification schemes," *Security Technology*, Springer Berlin Heidelberg, pp. 93-99, 2009. Article (CrossRef Link)
[10] Thorncharoensri, P., Susilo, W., & Mu, Y., "Identity-based identification scheme secure against concurrent-reset attacks without random oracles," *Information Security Applications*, Springer Berlin Heidelberg, pp. 94-108, 2009. Article (CrossRef Link)
[11] Fujioka, A., Saito, T., & Xagawa, K., "Security enhancements by OR-proof in identity-based identification," *Applied Cryptography and Network Security*, Springer Berlin Heidelberg, pp. 135-152, January, 2012. Article (CrossRef Link)
[12] Fujioka, A., Saito, T., & Xagawa, K., "Security enhancement of identity-based identification with reversibility," *Information and Communications Security*, Springer Berlin Heidelberg, pp. 202-213, 2012. Article (CrossRef Link)
[13] Dehkordi, M. H., & Alimoradi, R., "Certificateless identification protocols from super singular elliptic curve," *Security and Communication Networks*, 2013. Article (CrossRef Link)
[14] Chin, J. J., Behnia, R., Heng, S. H. and Phan, R. P. C., "Cryptanalysis of a certificateless identification scheme," *Security and Communication Networks*, 2014. Article (CrossRef Link)
[15] Chin, J. J., Heng, S. H., Phan, R. P. C & Behnia, R., "An Efficient and Provably Secure Certificateless Identification Scheme," in *Proc. of Proceedings of the 10th International Conference on Security and Cryptography*, SECRYPT , pp. 371-378, 2013.
[16] Bellare, M., & Rogaway, P., "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. of Proceedings of the 1st ACM conference on Computer and communications security*, ACM, pp. 62-73, December, 1993. Article (CrossRef Link)
[17] Canetti, R., Goldreich, O., & Halevi, S., "The random oracle methodology, revisited," *Journal of the ACM (JACM)*, 51(4), 557-594, 2004. Article (CrossRef Link)
[18] Boldyreva, A., "Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme," *Public key cryptography—PKC 2003*, Springer Berlin Heidelberg, pp. 31-46. 2002. Article (CrossRef Link)
[19] Damgård, I., "Towards practical public key systems secure against chosen ciphertext attacks," *Advances in Cryptology—CRYPTO'91*, Springer Berlin Heidelberg, pp. 445-456, January, 1992. Article (CrossRef Link)
[20] Huang, X., Mu, Y., Susilo, W., Wong, D. S., & Wu, W., "Certificateless signature revisited," *Information Security and Privacy*, Springer Berlin Heidelberg, pp. 308-322, January, 2007. Article (CrossRef Link)
[21] Huang, X., Mu, Y., Susilo, W., Wong, D. S., & Wu, W., "Certificateless signatures: new schemes and security models," *The Computer Journal*, 55(4), 457-474, 2012. Article (CrossRef Link)
[22] Bellare, M., & Palacio, A., "GQ and Schnorr identification schemes: Proofs of security against impersonation under active/concurrent attacks," *Advances in Cryptology—CRYPTO 2002*, Springer Berlin Heidelberg, pp. 162-177, 2002. Article (CrossRef Link)
[23] Tan, S. Y., Chin, J. J., Heng, S. H., & Goi, B. M., "An improved efficient provable secure identity-Based identification scheme in the standard model," *KSII Transactions on Internet and Information Systems (TIIS)*, 7(4), 910-922, 2013.

Ji-Jian Chin received his Bachelor of Science majoring in Computer Science and C omputational Mathematics from Campbell University and his Master of Engineering Science from Multimedia University. He is currently is a PhD student at the Faculty o f Information Science and Technology, Multimedia University while also teaching at t he Faculty of Engineering, Multimedia University. His research interest is in cryptog raphy, particularly in the area of public key cryptography.

Swee-Huay Heng received her B.Sc (Hons) and M.Sc degrees from University Put ra Malaysia (UPM), and her Doctor of Engineering degree from the Tokyo Institute o f Technology, Japan. She is currently the Dean and a Professor in the Faculty of Infor mation Science and Technology, Multimedia University, Malaysia. Her research inte rests include Cryptography and Information Security. She was the Programme Chair o f ProvSec 2010 and CANS 2010. She has been actively involved in technical Progra mme Committees of several international security conferences.

Raphael Phan received his B.Eng (Hons), M.Eng.Sc, and PhD degrees from Multi media University and currently holds a professor position at the Faculty of Engineeri ng, Multimedia University. Each year he serves in numerous technical Program Com mittees of international security conferences. He researches on diverse security and p rivacy topics ranging from cryptology to human-involved processes.