Malaysian Journal of Mathematical Sciences 13(S) August: 93–109 (2019)
Special Issue: The 6th International Cryptology and Information Security Conference (CRYPTOLOGY2018)

MALAYSIAN JOURNAL OF MATHEMATICAL SCIENCES

Journal homepage: http://einspem.upm.edu.my/journal

A Hierarchical Identity-Based Identification Scheme Without Pairing

Vangujar, A.K. *[1], Chin, J.J.[1], Tan, S.Y.[2], and Ng, T.S.[2]

[1] *Faculty of Engineering, Multimedia University, Malaysia*
[2] *Faculty of Information Science and Technology, Multimedia University, Malaysia*

E-mail: apurva710@gmail.com
* *Corresponding author*

ABSTRACT

In 2015, Chin et al. proposed an extension to the Schnorr IBI scheme using two secret keys to tighten the security based on the discrete logarithmic assumption, namely the Twin-Schnorr IBI. Twin-Schnorr IBI works without pairing operation and this helps to increase the efficiency of the scheme as well as strengthening it's security. In this paper, we extend Chin et al.'s scheme to accommodate hierarchies, namely the Hierarchical Identity-based identification (HIBI). Our scheme uses no pairings and is able to operate faster than pairing based HIBIs.

Keywords: Security attacks, Hierarchical Identity-based identification scheme (HIBI), Discrete logarithmic assumption, Twin-Schnorr IBI scheme.

1. Introduction

1.1 Identification Scheme

Public key cryptography uses the recipient's public key for encryption and the recipient's private key for decryption to recover the original message. The public key cryptography is further sub-divided into digital signature schemes and identification schemes.

An identification scheme consists of two parties, namely prover and a verifier in order to perform a challenge response protocol. An identification scheme allows a prover to prove himself to a verifier without revealing any information about himself. The traditional cryptographic scheme which includes identification schemes, requires the use of certificates issued by a certificate authority (CA) in order to authenticate user's public key. Maintaining certificates in large numbers in itself is a major issue.

Identification schemes first proposed in (Shamir, 1984) was built based on three-move protocol using zero-knowledge proof results into higher efficiency. In Identity-based cryptography Shamir, the certificate requirement is abolished by replacing the public key with an identity string Fiat and Shamir (1986). It is the simplest form of cryptographic primitive without relying on certificates. Conventional IBI schemes only allow single user interaction with the verifier.

1.2 Related Work

In 1989, Schnorr described the first scheme based on the discrete logarithm assumption and it is particularly suited for the smart cards. The key generation algorithm is faster and more secure than Shamir (1984) using an efficient algorithm to pre-process the exponentiation of random numbers.

Boneh and Franklin (2003) proposed the first identity-based encryption scheme, which lead to the booming of identity-based cryptography. Later years, IBI schemes were more secure and efficient formalized in Bellare et al. (2009). Subsequently, Tan et al. (2011) proposed a variant of Schnorr IBI scheme and direct proof with tight security reduction. He described the security against impersonator under passive, active and concurrent attack based on the Decisional Diffie Hellman (DDH) assumption in the random oracle model. Separately, Barapatre and Rangan (2013) also proposed another IBI scheme from ID Key Encapsulation Mechanisms. Finally Chin et al. (2015) introduced Twin-Schnorr IBI scheme. The authors proposed to generate two secret keys

in key generation algorithm. The authors prove that this method tightens the security for Schnorr IBI scheme to the discrete logarithm assumption only, at little additional cost Tan et al. (2011).

The first idea of Hierarchical identity-based encryption (HIBE) was first proposed by Horwitz and Lynn (2002). For hierarchical IBI (HIBI), Chin et al. (2009) extended Horwitz and Lynn (2002)'s construct in the first hierarchical IBI scheme. Both schemes use pairings. Fujioka et al. (2012) and subsequently Fujioka et al. (2014) extended the work of HIBI using constructions that utilize the RSA and CDH assumption. Fujioka's IBI scheme is proven secure in the standard security model whereas Chin et al. (2009) is proven efficient with random oracle. However, their HIBI without random oracle has increased communication cost and key size as compared to HIBI with random oracle.

This paper focuses on the Hierarchical IBI scheme without pairing and it's security proof of passive, active and concurrent attack respectively. HIBI has root PKG as the first-level and n lower-level PKG where n is defined by users. Each node is connected to other node and communicates with each other by three move protocol. The advantages of HIBI are listed as the following.

1. It is an efficient as there is no database needed for identities.
2. It has improved scalability.
3. It solves the key escrow problem with delegated key feature.

In this paper, we propose a Hierarchical version of the of Twin-Schnorr IBI scheme without pairing. The Hierarchical Twin-Schnorr IBI scheme without pairing has a Public Key Generator (PKG) which will distribute the secret key once and then partially creates multiple PKG.

1.3 Organization

The paper is organized as follows. In Section 2, we begin with some preliminaries including assumptions, groups, and security definitions for IBI schemes. In Section 3, we define the Hierarchical Twin-Schnorr IBI scheme. Section 5 tells us more detail about the Hierarchical IBI without pairing with JAVA code. We define the security proof against impersonation under active and concurrent attack for the Hierarchical IBI scheme without pairing in Section 4. In Section 6, we calculate the efficiency analysis of Hierarchical IBI scheme without pairing in comparison with other IBI schemes. We conclude this paper in Section 7.

2. Preliminaries

2.1 Discrete Logarithm Assumption

We adopt the definition of the discrete logarithm assumption from Kurosawa and Heng (2004), Bellare and Palacio (2002) Ioannidis et al. (2005) follows:

Definition 2.1. *Let G be a finite cyclic group of order n. Let α be a generator of G, and let $\beta \in G$. The discrete logarithm of β to the base α, denoted $log_\alpha \beta$, is the unique integer x, $0 \leq x \leq n-1$, such that $\beta = \alpha^x$.*

2.2 Formal Definition of IBI Schemes

Definition 2.2. *An identity-based identification (IBI) scheme is based on the four probabilistic algorithms.*

$$ID = (\mathcal{S}, \mathcal{E}, \mathcal{P}, \mathcal{V})$$

- ***Key Setup*** $(\mathcal{S})$. *It takes the input as 1^k and generates output as $(param, masterkey)$.*
- ***Extract*** $(\mathcal{E})$. *An extract oracle is used to extract the private key. Input $(masterkey, ID)$ and returns the private key d.*
- ***Identification Protocol*** $(\mathcal{P}$ *and* $\mathcal{V})$. *In this phase, the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ communicates with each other. $\mathcal{P}$ takes input as $(param, ID, d)$ whereas the $\mathcal{V}$ takes input as $(param, ID)$. $\mathcal{P}$ and $\mathcal{V}$ communicates with each other with the help of (CMT, CH, RSP) and gives output in boolean decision 0 (rejects) or 1 (accepts). The canonical protocol acts in four steps as following :*

  1. *$\mathcal{P}$ sends commitment (CMT) to $\mathcal{V}$.*
  2. *$\mathcal{V}$ provides challenge (CH) which is randomly chosen.*
  3. *$\mathcal{P}$ calculates the response (RSP) to $\mathcal{V}$ as per challenge.*
  4. *$\mathcal{V}$ verifies $(param, ID, CMT, CH, RSP)$ is DH tuple.*

2.3 HIBI Schemes

Definition 2.3. *An HIBI scheme is based on the four probabilistic algorithms. Gentry and Silverberg (2002) Chin et al. (2009)*

$$ID = (\mathcal{S}, \mathcal{E}, \mathcal{P}, \mathcal{V})$$

- ***Root Setup*** $(\mathcal{S})$. *This algorithm selects the random generator and a secret key and generates the output pair of param and secret key.*

    1. ***Lower Level Setup***. *This level in which all identities at lower level sets random parameter keeping it secret.*

- ***Extract*** $(\mathcal{E})$. *For any identity, it calculates user secret key with the help of ancestor secret key.*

- ***Identification Protocol*** $(\mathcal{P}, \mathcal{V})$. *In this phase, the prover* $\mathcal{P}$ *and the verifier* $\mathcal{V}$ *communicate in three steps as following.*

    1. $\mathcal{P}$ *chooses random variable to calculate value and send to* $\mathcal{V}$.
    2. $\mathcal{V}$ *generates the random challenge and forwards to* $\mathcal{P}$.
    3. $\mathcal{P}$ *accepts the challenge and generates response base on the challenge.*
    4. $\mathcal{V}$ *accepts if and only if, it verifies the final equation.*

An impersonator focuses to impersonate an honest user. The following two section states the types of adversary.

- A passive adversary. This is the attack where an adversary obtains the communication transcript between the real prover and a verifier. An adversary only can steal the information but doesn't affect the communication line between the prover and the verifier. This is the weakest attack.

- An active adversary and concurrent adversary. An adversary can directly communicate with the prover playing the role of a cheating verifier actively. The adversary in the active attack can drop, change and configure the information. It threatens authentication and integrity of data. An adversary can concurrently communicate with communication protocol the prover playing the role of the cheating verifier and an adversary can do changes in between ongoing process Katz and Lindell (2014).

  We adopt the security model for IBI scheme from Chin et al. (2009). An impersonation attack between an impersonator $\mathcal{I}$ and a challenger $\mathcal{C}$ is described as a two-phased game as follows:

    1. **Setup (S)**. $\mathcal{C}$ takes input 1 and runs algorithm S. The result of system parameters mpk is given to $\mathcal{I}$ while msk is kept to itself.
    2. **Phase 1**: Learning Phase. $\mathcal{I}$ issues some extract queries ID_i to $\mathcal{C}$. $\mathcal{C}$ responds by running the extract algorithm to generate and return the private key usk corresponding to the identity ID_i to $\mathcal{I}$.

The queries may be asked adaptively. $\mathcal{I}$ issues transcript queries for passive attacks or requests to act as a cheating verifier corresponding to some ID_i for active/concurrent attacks.

3. **Phase 2**: Impersonation Phase. Finally, outputs a challenge identity ID which it wishes to impersonate whereby $\mathcal{I}$ now acts as a cheating prover to convince the verifier $\mathcal{C}$ based on information gathered in Phase 1. $\mathcal{I}$ wins the game if it is successful in convincing the verifier.

2.4 Security Model for HIBI Schemes

We describe the security of a HIBI scheme with the following game between an impersonator $\mathcal{I}$ and a challenger $\mathcal{C}$.Chin et al. (2009)

1. **Setup** ($\mathcal{S}$). The challenger first takes in a security parameter 1^k and gives the resulting *params* to the $\mathcal{I}$. It keeps *rlmsk* root-level master secret key to itself.

2. **Phase 1**. $\mathcal{I}$ can issue queries $(q_i, ..., q_m)$ where q_i is one of:

   (a) **Extract Key Query**($\mathcal{E}$). Upon being queried with the public key of ID_i, returns usk_i to $\mathcal{I}$.

   (b) **Transcript/Identification Query**($\mathcal{P}$ and $\mathcal{V}$). For passive $\mathcal{I}$, $\mathcal{C}$ responds with a transcript for the interaction between the prover and a verifier. For active/concurrent, $\mathcal{C}$ acts as the prover while $\mathcal{I}$ takes the role of a cheating verifier.

3. **Challenge** ($\mathcal{C}$). $\mathcal{I}$ outputs $ID^* \neq ID_i$ wishes to impersonate. ID^* is the targeted identity by impersonator among $(ID_1, ..., ID_i)$.

4. **Phase 2**.

   (a) **Extract Key Query**($\mathcal{E}$). $\mathcal{I}$ can continue to query the private keys of ID_i as long as ID_i is not an ancestor of $ID^* \neq ID_i$.

   (b) **transcripts/Identification Query**($\mathcal{P}$ and $\mathcal{V}$). $\mathcal{I}$ can continue to query either transcripts for passive $\mathcal{I}$ or identification interactions for active/concurrent $\mathcal{I}$ for ID^* or any ancestor of ID^*.

5. **Impersonation**. $\mathcal{I}$ takes the role of the cheating prover and tries to convince the verifier. $\mathcal{I}$ wins the game if it succeeds in convincing the verifier to accept with non-negligible probability.

Definition 2.4. *We say an HIBI scheme is* $(t_{HIBI}, q_{HIBI}, \varepsilon_{HIBI})$ *-secure under passive or active/concurrent attacks if for any passive/active/concurrent* $\mathcal{I}$ *who runs in time* t_{HIBI}*,* $Pr[\mathcal{I}\text{can impersonate}] \leq \varepsilon_{HIBI}$*, where* $\mathcal{I}$ *can make at most* q_{HIBI} *extract queries and transcripts/Identification Query.*

3. The Hierarchical IBI Scheme Without Pairing

The Hierarchical IBI scheme without pairing which is based on the Twin-Schnorr IBI scheme by Chin et al. (2015).

Root level consists of ID_0 identity. The hierarchy proceed for $level_1$ having identities $(ID_1, ID_2, ..., ID_i, ..., ID_m)$ where m represent the last identity of that level. ID_i is the targeted identity which can exist in a such a way that $(ID_1 \leq ID_i \geq ID_m)$. The construction of the Hierarchical IBI scheme without pairing algorithms $(\mathcal{S}, \mathcal{E}, \mathcal{P}, \mathcal{V})$ are as follows Chin et al. (2009).

1. **Key Setup** $(\mathcal{S})$. It takes 1^k where k is the security parameter and generates $\mathbb{G}$ the group of order q. It picks random generators $g_1, g_2 \in \mathbb{G}$ and two random integers $x_1, x_2 \in \mathbb{Z}_q$. It sets $X = g_1^{-x_1} g_2^{-x_2}$. It chooses a hash function $H : (0,1)^* \times \mathbb{G} \times \mathbb{G} \Rightarrow \mathbb{Z}_q$. It publishes pair of (mpk, msk) where $mpk = \langle \mathbb{G}, q, g_1, g_2, X \rangle$ and $msk = \langle x_1, x_2 \rangle$.

2. **Extract** $(\mathcal{E})$. For ID_0 root level, (mpk, msk, ID_0) is the input. It calculates $R = g_1^{x_1} g_2^{x_2}$ and sets $\alpha_0 = \mathbb{H}(ID_0, R, X)$. later, It picks two random integers $r_{0,1}, r_{0,2} \in Z_q$ and calculates $S_{0,1} = r_{0,1} + x_1\alpha_0$, $S_{0,2} = r_{0,2} + x_2\alpha_0$. Finally, it sets $usk_{ID_0} = \langle S_{0,1}, S_{0,2}, \alpha_0 \rangle$. It passes usk_{ID_0} to next level.

   For ID_1 level 1, It picks two random integers $r_{1,1}, r_{1,2} \in Z_q$ and to calculate $(S_{1,1}, S_{1,2})$ where $S_{1,1} = r_{1,1} + \alpha_1 + S_{0,1}$ and $S_{1,2} = r_{1,2} + \alpha_1 + S_{0,2}$, it uses the $(S_{0,1}, S_{0,2})$ as ancestor user secret key. Therefore, $usk_{ID_1} = \langle S_{1,1}, S_{1,2}, \alpha_1 \rangle$.

   For ID_i level i, it takes input $mpk = \langle \mathbb{G}, q, g_1, g_2, X \rangle$, $usk_{ID_{i-1}} = \langle S_{i-1}, S_{i-2} \rangle$ and user identity string $(ID_0, ..., ID_i)$. It picks two random integers $r_{i,1}, r_{i,2} \in Z_q$, calculates $V_i = g_1^{r_{i,1}+S_{i-1,1}} g_2^{r_{i,2}+S_{i-1,2}}$ and sets $\alpha_i = \mathbb{H}(ID_0||...||ID_i, V_i, X)$. Next, calculates $S_{i,1} = r_{i,1} + \alpha_i + S_{i-1,1}$ and $S_{i,2} = r_{i,2} + \alpha_i + S_{i-1,2}$ and sets $usk_{ID_i} = \langle S_{i,1}, S_{i,2}, \alpha_i \rangle$.

3. **Identification Protocol** ($\mathcal{P}$ and $\mathcal{V}$) in which prover takes in mpk, ID_i and usk_{ID_i} while $\mathcal{V}$ takes in mpk and ID_i. They run an identification protocol as follows.

- $\mathcal{P}$ begins by picking two random integers $y_{i,1}, y_{i,2} \in \mathbb{Z}_q$ and sets $Y = g_1^{y_{i,1}} g_2^{y_{i,2}}$. $\mathcal{P}$ additionally sets $V_i = g_1{}^{S_{i,1}} g_2{}^{S_{i,2}} X^{\alpha_i}$ and sends Y, V_i to $\mathcal{V}$.
- $\mathcal{V}$ picks a random challenge $c \in Z_q$ and sends it to $\mathcal{P}$.
- $\mathcal{P}$ responds by setting $z_{i,1} = y_{i,1} + cS_{i,1}$ and $z_{i,2} = y_{i,2} + cS_{i,2}$ and sends $z_{i,1}, z_{i,2}$ to $\mathcal{V}$ as it's response.

  $\mathcal{V}$ calculates and accepts if the following equation holds for each i:

$$g_1^{z_{i,1}} g_2^{z_{i,2}} = Y \left(\frac{V_i}{X^{\alpha_i^{'}}} \right)^c$$

where $\alpha_i^{'} = H(ID_i, V_i, X)$ VERIFY can calculate $\alpha_i^{'} = H(ID_i, V_i, X)$ by itself since

$$\begin{aligned} g_1^{S_{i,1}} g_2^{S_{i,2}} X_i^{\alpha} &= g_1^{r_{i,1}+\alpha_i} g_2^{r_{i,2}+\alpha_i} g_1^{-\alpha_i} g_2^{-\alpha_i} \\ &= g_1^{r_{i,1}} g_2^{r_{i,2}} \\ &= R \end{aligned}$$

The correctness of the identification protocol can be proven as such:

$$\begin{aligned} Y \left(\frac{V_i}{X^{\alpha_i}} \right)^c &= g_1^{y_{i,1}} g_2^{y_{i,2}} \left(\frac{g_1^{S_{i,1}} g_2^{S_{i,2}} X^{\alpha_i}}{X^{\alpha_i}} \right)^c \\ &= (g_1^{y_{i,1}} g_2^{y_{i,2}}) \left(g_1^{S_{i,1}} g_2^{S_{i,2}} \right)^c \\ &= (g_1^{y_{i,1}} g_2^{y_{i,2}}) \left(g_1^{cS_{i,1}} g_2^{cS_{i,2}} \right) \\ &= g_1^{y_{i,1}+cS_{i,1}} g_2^{y_{i,2}+cS_{i,2}} \\ &= g_1^{z_{i,1}} g_2^{z_{i,2}} \end{aligned}$$

4. Security Analysis

We describe the security of the Hierarchical IBI scheme without pairing the following game between an impersonator $\mathcal{I}$ and a challenger $\mathcal{C}$.

Theorem 4.1. *Hierarchical IBI scheme without pairing is secure against impersonation under active and concurrent attack if the discrete logarithm problem is hard in group* $\mathbb{G}$*, where*

$$\varepsilon_{HIBIwithoutpairing}^{imp_{pa}} = \sqrt[l]{\varepsilon_{G,C}^{DLOG}(k) + \frac{1}{2^k} + \frac{1}{2^k}}$$

Proof. Let $\mathcal{I}$ be an impersonator who (t, q_i, ϵ) breaks the security of Hierarchical IBI scheme without pairing. $\mathcal{C}$ is a simulator that find out the value of a according to discrete logarithm assumption. $\mathcal{C}$ will be given a group G, generators $(g_1 = g, g_2 = g^a) \in G$, $\mathcal{C}$ will simulate for $\mathcal{I}$ as follows.

1. **Setup**$(\mathcal{S})$. $\mathcal{C}$ takes 1^k and returns $mpk = \langle \mathbb{G}, q, g_1, g_2, X \rangle$ to $\mathcal{I}$.
2. **Phase 1**. $\mathcal{I}$ can issue queries $(q_0, .., q_i, ..., q_m)$ where q_i is for ID_i. There are q_m queries in total as there is m number of queries. In training phase, $\mathcal{I}$ tries to learn from the $\mathcal{C}$. It will forge the user secret key and runs transcript. It is considered as a hierarchical version of the Twin-Schnorr IBI scheme without pairing for $(ID_1, ..., ID_m)$, where $(ID_1, ..., ID_i)$ for $1 \leq i \leq m$ and $(level_1, level_2, ..., level_l)$ where $(level_1, ..., level_j)$ for $1 \leq j \leq l$ to define hierarchy.
   (a) Case 1.
      i. **Extract Query** $(\mathcal{E})$. For $ID_i \neq ID^*$, $\mathcal{C}$ takes master public key and identity string as the input. Upon being queried with the public key of ID_i and returns $usk_{ID_i} = (S_{i,1}, S_{i,2})$ to $\mathcal{I}$. To calculate usk_{ID_i} with the help of ancestor $usk_{ID_{i-1}}$ can be done.
      ii. **Identification query** ($\mathcal{P}$ and $\mathcal{V}$). For $\mathcal{I}$, $\mathcal{C}$ responds with a transcript for the interaction between the prover and a verifier. In the simulation, Prover takes input (mpk, ID_i, usk_{ID_i}) where the verifier takes input (mpk, ID_i). Prover generates (Y, V_i). $\mathcal{C}$ generates random challenge $c \in Z_q$. On the basis of challenge prover calculates $z_{i,1}, z_{i,2}$ to $\mathcal{V}$ as its response. Lastly $\mathcal{V}$ verifies $g_1^{z_{i,1}} g_2^{z_{i,2}} = Y \left(\frac{V_i}{X^{\alpha_i'}}\right)^c$
   (b) Case 2.
      i. **Extract Query** $(\mathcal{E})$. For $ID^* = ID_i$, the ancestor of usk_{ID^*} is unknown. But, the root secret key is known. Therefore, the algorithm aborts. There is ID string where all ID are defined as parent and child node according to hierarchy. Parent helps to generate usk of child node. Child node's usk is generated only in case it has parent usk defined. $\mathcal{C}$ takes master public key and identity string as the input. Upon being queried with the public key of ID^* and returns $usk_{ID^*} = (S_{*,1}, S_{*,2})$ to $\mathcal{I}$.
      ii. **Identification query** ($\mathcal{P}$ and $\mathcal{V}$). When transcript will create even if not yet queried before as an extract query. Prover participate in transcript and add in the set. We will not able to issue transcript for the already corrupted user. Prover and

verifier communicates in this phase. ID^* is targeted identity and verifier needs to verify it. $ID_i = ID_*$, $\mathcal{I}$ act as the cheater $\mathcal{V}$ and $\mathcal{C}$ does not have user secret key of ID_*, however it needs to create it again to run an identification protocol. When $\mathcal{I}$ tries to forge ID^* then he should know the previous (ID_{*-1}). We can perform transcript as many times as number of queries does not exceed. Prover takes input (mpk, ID^*, usk_{ID^*}) where the verifier takes input (mpk, ID^*). Prover generates (Y, V_*). $\mathcal{C}$ generates random challenge $c \in Z_q$ where c corresponds to ID^*. On the basis of challenge prover calculates $z_{*,1}, z_{*,2}$ to $\mathcal{V}$ as its response. Lastly $\mathcal{V}$ verifies $g_1^{z_{*,1}} g_2^{z_{*,2}} = Y \left(\frac{V_*}{X^{\alpha'_*}}\right)^c$.

(c) **Challenge** $(\mathcal{C})$. $\mathcal{I}$ outputs an $ID_i \neq ID^*$ that it wishes to impersonate.

3. **Phase 2**. Breaking phase calculates as follows:

$[y_{i,1}, c_1, V_i, z_{i,1}]$ and $[y_{i,2}, c_2, V_i, z_{i,2}]$ from $\mathcal{I}$ where $c_1 \neq c_2$. From here, $\mathcal{C}$ extracts $\widetilde{S_{i,1}} = (z_{i,1} - z_{i,2})/(c_2 - c_1)$ and $\widetilde{S_{i,2}} = (z_{i,1} - z_{i,2})/(c_2 - c_1)$.

If $S_{i,1} = \widetilde{S_{i,1}}$ and $S_{i,2} = \widetilde{S_{i,2}}$ then $\mathcal{C}$ aborts.

$$
\begin{aligned}
g^{S_{i,1}} g^{S_{i,2}} &= g^{\widetilde{S_{i,1}}} g^{\widetilde{S_{i,2}}} \\
g^{S_{i,1} + aS_{i,2}} &= g^{\widetilde{S_{i,1}} + a\widetilde{S_{i,2}}} \\
g^{aS_{i,2}} - g^{a\widetilde{S_{i,2}}} &= g^{\widetilde{S_{i,1}}} - g^{S_{i,1}} \\
g^a &= g^{(\widetilde{S_{i,1}} - S_{i,1})(S_{i,2} - \widetilde{S_{i,2}})} \\
a &= -\frac{\widetilde{S_{i,1}} - S_{i,1}}{S_{i,2} - \widetilde{S_{i,2}}}
\end{aligned}
$$

□

To calculate the probability of $\mathcal{C}$ winning the game to solve the discrete logarithm problem. By the Reset Lemma, will successfully extract 2 valid conversations to derive $(S_{i,1}, S_{i,2})$ and calculating a with the probability $\varepsilon^{imp_{aa/ca}}_{HIBI_{w}ithoutpairing} - (-\frac{1}{2^k} - \frac{1}{2^k})^l$ Assume $\mathcal{C}$ solves the discrete logarithm assumption. $\mathcal{C}$ which computes correct value of a then event is A and not aborting event is B. Winning

probability can be given as following.

$$\mathcal{C} = Pr[A \wedge B]$$
$$\mathcal{C} = Pr[A|B]Pr[B]$$
$$\varepsilon_{G,C}^{DLOG}(k) \geq (\varepsilon - \frac{1}{2^k})^l Pr[B]$$

The probability of $\mathcal{C}$ aborting when event B is $S_{i,1} = \widetilde{S_{i,1}}$ and $S_{i,2} = \widetilde{S_{i,2}}$. Therefore probability of winning $\mathcal{C}$ is,

$$\varepsilon_{G,C}^{DLOG}(k) \geq (\varepsilon_{HIBIwithoutpairing}^{imp_{ca/aa}} - \frac{1}{2^k})^l - \frac{1}{2^k}$$
$$\varepsilon_{G,C}^{DLOG}(k) + \frac{1}{2^k} \geq (\varepsilon_{THIBIwithoutpairing}^{imp_{ca/aa}} - \frac{1}{2^k})^l$$
$$\varepsilon_{HIBIwithoutpairing}^{imp_{pa}} \leq \sqrt[l]{\varepsilon_{G,C}^{DLOG}(k) + (\frac{1}{2^k} + \frac{1}{2^k})}$$

5. Implementation of JAVA code for Hierarchical IBI without Pairing

In this section, we show our implementation of the Hierarchical IBI without Pairing simulator in Java. We used NetBeans IDE 8.2 as the front end and JDK Bundle as back end. Previously Schnorr, Tight Schnorr, Twin Schnorr, RS Twin Schnorr IBI Scheme have been implemented in (Kam et al., 2015). We extend this work by adding in the Hierarchical Twin-Schnorr IBI without pairings to the Schnorr Suite prototype. A step by step implementation of the JAVA code to simulate the Hierarchical IBI without pairing is given in figures. /par

Figure 1 shows the interface of the implemented code. It shows the list of schemes,input text box and algorithms. Users need to select Hierarchical IBI without pairing Scheme from given list of Schemes and enter the ID-string along with number of iteration given in Figure 2.

Figure 3 tells us more about the generation of master public and master secret key for the Hierarchical IBI without pairing. For 100 iterations, 48.115 milliseconds is the average time taken.

The User secret key is calculated in Figure 4. We conduct simulations for up to 4 levels of hierarchy. For 100 iterations, We measure the average time in milliseconds to run extraction for the user secret key at each level. Root level (similar to conventional IBI) takes 3.173, level 1 takes 3.210, level 2 takes 3.231, level 3 takes 3.253 and finally level 4 takes 3.274 milliseconds.
Figure 5 elaborates the communication between a prover and the verifier for Hierarchical IBI without pairing for one iteration. For 100 iterations, identification protocol takes average times to run as follows: root level takes 1.279, level 1 takes 2.552, level 2 takes 3.848, level 3 takes 5.135 and finally level 4 takes 6.399 milliseconds. This is consistent that the time taken will increase slightly with each level of the hierarchy added to the protocol.

Figure 1: Default Demo page for Hierarchical IBI without pairing.

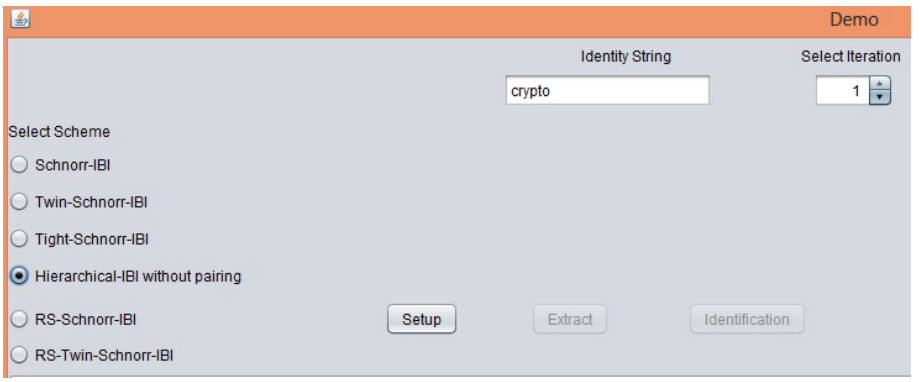

Figure 2: Selection for Scheme,ID-string and Iteration.

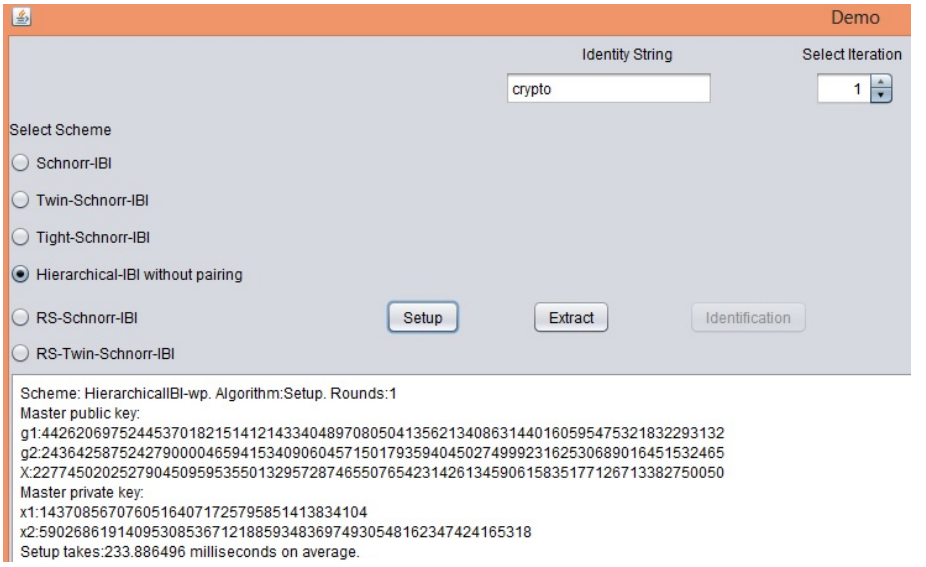

Figure 3: Setup Algorithm for Hierarchical IBI without pairing

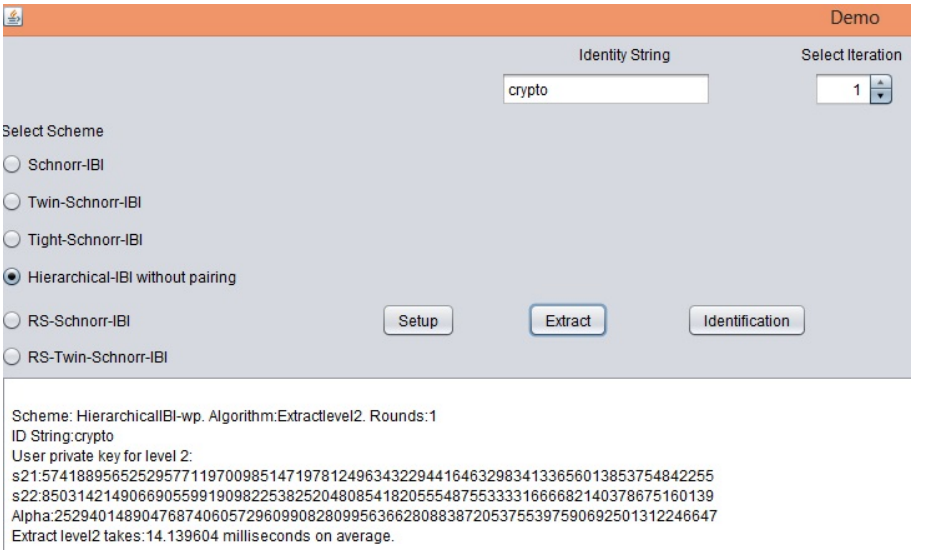

Figure 4: Extract Algorithm for Hierarchical IBI without pairing

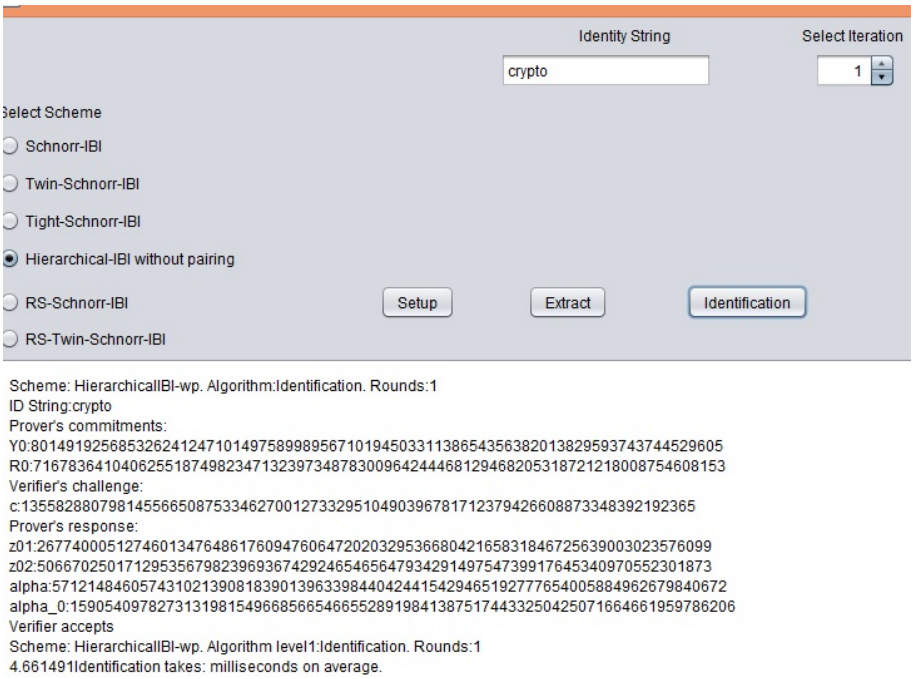

Figure 5: Identification Protocol for Hierarchical IBI without pairing

6. Efficiency Analysis

In this section, we provide the efficiency cost of the Hierarchical IBI scheme without pairing in Table 1. We consider pairings (P), exponentiation (E), multiplications in group $\mathbb{G}$ (MG), multiplications in $\mathbb{Z}_q$ (MZ) and additions in $\mathbb{Z}_q$ (A) in terms to define the efficiency in order.

We consider other schemes in order to calculate the identification cost in Table 2. The Twin-Schnorr IBI is slightly superior in terms of efficiency and

Table 1: Efficiency analysis for the Hierarchical IBI scheme without pairing

Algorithm	E	MG	MZ	A
SETUP	2	1	0	0
EXTRACT	4	2	4	4
PROVE	5	3	2	2
VERIFY	4	3	0	0

Table 2: Comparison of the identification protocol with other HIBI schemes, where l is the number of hierarchy levels

Scheme	P	E	MG	MZ	A	Assumption
HIBI by Chin et al. (2009)	l+1	l	l	0	0	CDH,OMCDH
Waters-HIBI by Fujioka et al. (2014)	4	6	3	0	0	Prime order bilinear group
Hess-HIBI by Fujioka et al. (2014)	4	6	3	0	0	Composite order bilinear group
RSA-HIBI by Fujioka et al. (2014)	0	3	0	3	4	RSA
HIBI without pairing	0	9	6	2	2	DLP

security compared to the HIBI scheme proposed in Fujioka et al. (2014). We are considering the Hierarchical IBI scheme without pairing which is efficient scheme in case of targeted identity.

According to communication cost calculation, the Hierarchical IBI without pairing is more efficient and secure compared to the other HIBI schemes with the exception of Fujioka et al's RSA-scheme, since pairing operations are costly.

7. Concluding Remarks

In this paper, we upgraded the Twin-Schnorr IBI scheme into the Hierarchical IBI scheme without pairing. Our proposed Hierarchical IBI scheme without pairing is designed to prove many identification and verification at a time. The proposed scheme is efficient as it is pairing-free and secure based the discrete logarithmic assumption.In this paper, we upgraded the Twin-Schnorr IBI scheme into the Hierarchical IBI scheme without pairing. Our proposed Hierarchical IBI scheme without pairing is designed to prove many identification and verification at a time. The proposed scheme is efficient as it is pairing-free and secure based the discrete logarithmic assumption.

References

Barapatre, P. and Rangan, C. P. (2013). Identity-based identification schemes from id-kems. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 111–129. Springer.

Bellare, M., Namprempre, C., and Neven, G. (2009). Security proofs for identity-based identification and signature schemes. *Journal of Cryptology*, 22(1):1–61.

Bellare, M. and Palacio, A. (2002). Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *Annual International Cryptology Conference*, pages 162–177. Springer.

Boneh, D. and Franklin, M. (2003). Identity-based encryption from the weil pairing. *SIAM journal on computing*, 32(3):586–615.

Chin, J.-J., Heng, S.-H., and Goi, B.-M. (2009). Hierarchical identity-based identification schemes. In *International Conference on Security Technology*, pages 93–99. Springer.

Chin, J.-J., Tan, S.-Y., Heng, S.-H., and Phan, R. C.-W. (2015). Twin-schnorr: a security upgrade for the schnorr identity-based identification scheme. *The Scientific World Journal*, 2015.

Fiat, A. and Shamir, A. (1986). How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 186–194. Springer.

Fujioka, A., Saito, T., and Xagawa, K. (2012). Security enhancements by or-proof in identity-based identification. In *International Conference on Applied Cryptography and Network Security*, pages 135–152. Springer.

Fujioka, A., Saito, T., and Xagawa, K. (2014). Secure hierarchical identity-based identification without random oracles. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 97(6):1307–1317.

Gentry, C. and Silverberg, A. (2002). Hierarchical id-based cryptography. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 548–566. Springer.

Horwitz, J. and Lynn, B. (2002). Toward hierarchical identity-based encryption. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 466–481. Springer.

Ioannidis, J., Keromytis, A., and Yung, M. (2005). *Applied cryptography and network security*. Springer Berlin/Heidelberg.

Kam, Y. H. S., Chin, J. J., and Tan, S. Y. (2015). The schnorr-suite: Simulation of pairing-free identity-based identification schemes using java. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 13–18. SEKEIE 2015.

Katz, J. and Lindell, Y. (2014). *Introduction to modern cryptography*. CRC press.

Kurosawa, K. and Heng, S.-H. (2004). From digital signature to id-based identification/signature. In *International Workshop on Public Key Cryptography*, pages 248–261. Springer.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques*, pages 47–53. Springer.

Tan, S.-Y., Heng, S.-H., Phan, R. C.-W., and Goi, B.-M. (2011). A variant of schnorr identity-based identification scheme with tight reduction. In *International Conference on Future Generation Information Technology*, pages 361–370. Springer.