INFORMATICA, 2014, Vol. 25, No. 2, 299–326

RHIBE: Constructing Revocable Hierarchical ID-Based Encryption from HIBE

Tung-Tso TSAI[1], Yuh-Min TSENG[2] *, Tsu-Yang WU[2]

[1] *Department of Mathematics, National Changhua University of Education*
Jin-De Campus, Chang-Hua City 500, Taiwan, R.O.C.
[2] *School of Computer Science and Technology, Shenzhen Graduate School*
Harbin Institute of Technology, Shenzhen 518055, P.R. China
e-mail: ymtseng@cc.ncue.edu.tw

Received: August 2012; accepted: December 2012

Abstract. Up to now, there was very little work on studying the revocation problem in existing hierarchical ID-based encryption (HIBE) systems. Certainly, all existing HIBE systems may inherit the revocation method suggested by Boneh and Franklin to revoke illegal or expired users, in which non-revoked users must periodically update their private keys using secure channels by contacting their ancestors in hierarchical structures. In this paper, we propose the first HIBE scheme with public revocation mechanism, called revocable HIBE (RHIBE), which is extended from Lewko and Waters's unbounded HIBE scheme presented in Eurocrypt 2011. We demonstrate that the proposed RHIBE scheme is fully secure while removing the requirement of secure channels for private key updating in Boneh and Franklin's revocation method. The public revocation mechanism is an exciting alternative to the existing revocation methods. Finally, we discuss the transformation technique from a HIBE scheme to a RHIBE scheme and employ it to another well-known HIBE scheme.

Key words: revocation, hierarchical identity-based encryption, full security, bilinear pairing, public channel.

1. Introduction

In order to eliminate the need of certificates that make publicly available the mapping between identities and public keys in traditional public key systems, Shamir (1984) presented a good idea for public key systems, called identity (ID)-based public key system. In an ID-based public key system, a user's identity (e.g. name, e-mail address or social security number) is viewed as the user's public key. However, Shamir's system is not easy to be realized in practice. In 2001, Boneh and Franklin (2001) used Shamir's idea to propose the first practical ID-based encryption (IBE) scheme. In their IBE scheme, there are two roles: a trusted private key generator (PKG) and users at the same level. All users need to authenticate themselves to the PKG and then the PKG generates the corresponding private keys to the users. Following Boneh and Franklin's IBE system, a large number of literatures in the ID-based cryptography have been published such as Paterson (2002), Cha and Cheon

*Corresponding author.

(2003), Paterson and Schuldt (2006), Chen *et al.* (2007), Tseng *et al.* (2008), Wu, *et al.* (2011), Wu and Tseng (2012), Chen *et al.* (2012), in particular IBE (Boneh and Boyen, 2004a; Waters, 2005; Gentry, 2006; Boneh and Hamburg, 2008; Boldyreva *et al.*, 2008; Bellare *et al.*, 2011; Tseng and Tsai, 2012; Tsai *et al.*, 2012). However, many organizations have hierarchical structures, and users are distributed in different levels. Using Boneh and Franklin's IBE system cannot offer these organizations with hierarchical structure the flexibility of levels in sharing and managing sensitive data. For solving this problem, the PKG needs to delegate private keys to its subordinates. In addition, these subordinates in turn can keep delegating private keys further down the hierarchy to the users. Horwitz and Lynn (2002) first introduced a hierarchical IBE (HIBE) to meet the requirements above. Gentry and Silverberg (2002) then proposed the first fully functional HIBE system. Subsequently, the design of hierarchical ID-based encryption has received much attention from cryptographic researchers such as Boneh *et al.* (2005a), Okamoto and Takashima (2009), Waters (2009), Lewko *et al.* (2010), Lewko and Waters (2010, 2011).

However, any public key system must provide a revocation mechanism to remove illegal or expired users from the system. For the revocation problem, Boneh and Franklin (2001) also suggested a revocation mechanism for ID-based public key systems, in which the PKG must generate new private keys for all non-revoked users in each period, and a secure channel must be established between the PKG and each non-revoked user to transmit the periodic private keys. To our best knowledge, no revocation mechanism is proposed for dedicating to the existing HIBE systems. Indeed, all existing HIBE systems may inherit the revocation method suggested by Boneh and Franklin to revoke illegal or expired users. As a result, non-revoked users must periodically update their private keys using secure channels by contacting their ancestors (or the root PKG) in hierarchical structures. In such a case, their ancestors and these non-revoked users must encrypt and decrypt periodic private keys, respectively.

In this paper, we address the revocation problem in HIBE systems. We will propose the first revocable HIBE (RHIBE) scheme with public revocation mechanism in the standard model, which is extended from Lewko and Waters's unbounded HIBE scheme (2011). The involved public revocation mechanism is an exciting alternative to the existing revocation methods. In our proposed RHIBE scheme, non-revoked users and their ancestors do not require secure channels to transmit periodic private keys so that it can avoid enormous computation costs of encrypting/decrypting these periodic private keys. Finally, we discuss the transformation technique from a HIBE scheme to a RHIBE scheme. We believe that our transformation technique is suitable to construct RHIBE schemes from most of the existing HIBE schemes. Meanwhile, we employ the transformation technique to present another RHIBE scheme with public revocation mechanism in the random oracle model, which is extended from Gentry and Silverberg's HIBE scheme (2002).

1.1. *Related Work*

Boneh and Franklin (2001) used Shamir's idea to propose the first practical IBE system in the random oracle model. Although the IBE schemes based on the random oracle model can offer better performance, the resulting schemes could be insecure when random oracles are instantiated with concrete hash functions (Bellare *et al.*, 2004). For the IBE construction in the standard model, Canetti *et al.* (2003) presented an IBE scheme that is proven selectively secure in the standard model (without random oracle model). Subsequently, Boneh and Boyen (2004a) provided a more practical IBE scheme that is also proven selectively secure. Boneh and Boyen (2004a) furthermore presented a fully secure IBE scheme that is proven adaptively secure in the standard model. For improving the efficiency of Boneh and Boyen's IBE scheme, Waters (2005) also proposed an efficient IBE scheme in the standard model. However, Waters's IBE scheme had public parameters consisting of $O(\lambda)$ group elements, where λ is the security parameter. Furthermore, Gentry (2006) proposed a new fully secure IBE scheme to reduce the number of public parameters required in Waters's construction, but it relied on a "q-type" assumption (meaning that the number of terms in the assumption depending on the number of queries q made by an attacker), called the augmented bilinear Diffie–Hellman exponent (ABDHE) assumption (Gentry, 2006). Waters (2009) provided a dual system encryption IBE scheme with short public parameters that is proven fully secure under the decisional linear and decisional bilinear Diffie–Hellman assumptions.

Following the IBE system proposed by Boneh and Franklin (2001), Horwitz and Lynn (2002) first introduced a hierarchical ID-based encryption (HIBE). Afterward, Gentry and Silverberg (2002) proposed the first fully functional HIBE system in the random oracle model. Boneh and Boyen (2004a) proposed a HIBE scheme that is proven selectively secure in the standard model, but in their HIBE scheme ciphertext size, private key size and the computational time required for decryption and encryption, grow linearly in the depth of the hierarchy. For reducing the ciphertext size, Boneh *et al.* (2005a) presented a HIBE scheme with constant size ciphertext. They proved that their HIBE scheme is selective-ID secure in the standard model and fully secure in the random oracle model. Boyen and Waters (2006) presented an anonymous hierarchical identity-based encryption in the standard model that features fully anonymous ciphertexts and realizes fully anonymous HIBE at all levels in the hierarchy. Gentry and Halevi (2009) presented a fully secure HIBE construction for polynomial depth, but it relies on a complex assumption. Under simple assumptions (the decisional linear and decisional bilinear Diffie–Hellman assumptions), Waters (2009) proposed a fully secure dual system encryption HIBE scheme in the standard model. However, the ciphtext size of Waters's construction is dependent on the depth of the hierarchy. Note that the construction of Boneh *et al.* (2005a) has constant size ciphertext, but it was proven to be secure under a non-static assumption, which depended on the depth of the hierarchy. Recently, Lewko and Waters (2010) applied the dual system encryption concept proposed by Waters (2009) to present a construction with short ciphertext. They presented the HIBE system in composite order groups of three primes and achieved full security under static assumptions.

In the existing HIBE constructions, a maximum hierarchy depth had to be fixed at setup phase except the HIBE scheme proposed by Lewko and Waters (2011). They presented a HIBE scheme, which is "unbounded" in the sense that the public parameters do not impose additional limitations on the functionality of the system. They employed a secret-sharing technique to overcome the limitations of the previous HIBE constructions. We use

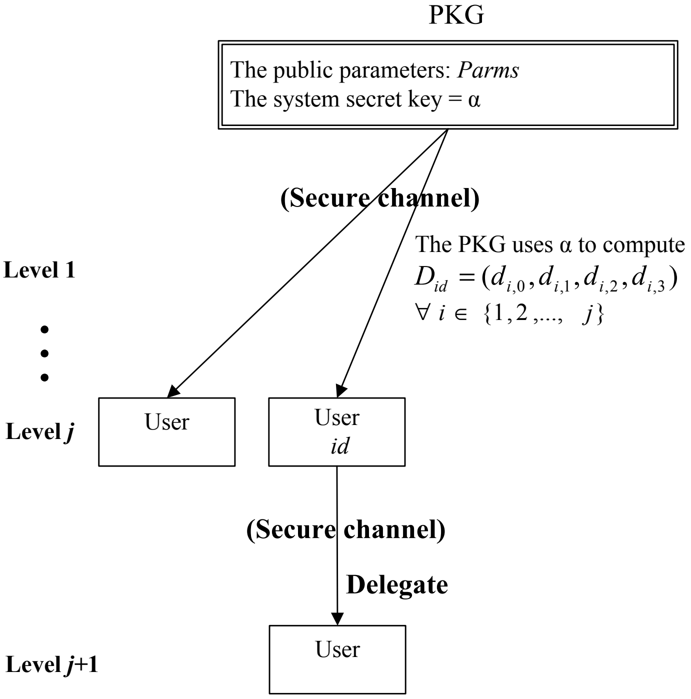
PKG
The public parameters: *Parms*
The system secret key = α
(Secure channel)
The PKG uses α to compute
$D_{id} = (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3})$
$\forall\ i \in \{1, 2, \ldots, j\}$
Level 1
Level *j*
User
User *id*
(Secure channel)
Delegate
Level *j*+1
User

Fig. 1. The decryption key generation of HIBE.

Fig. 1 to illustrate Lewko and Waters's HIBE scheme, in which α is the system's secret key. A user *id* at level j in their HIBE scheme can obtain its associated private key D_{id} from the PKG or its ancestors. Then the user *id* at level j can use its associated private key D_{id} to delegate private keys to its subordinates. Obviously, any user in their HIBE scheme can revoke its illegal or expired subordinates when this user stops to delegate new private keys to these subordinates. Thus, a secure channel needs to be established to transmit its subordinates' new private keys for each period.

As we all know, any public key system must provide a revocation mechanism to remove illegal or expired users from the system. For this problem, Boneh and Franklin (2001) suggested that the PKG generates all non-revoked users' new private keys of each period and then uses secure channels to transmit these periodic private keys to non-revoked users. For reducing the PKG and user's periodic computational workload, Boldyreva *et al.* (2008) proposed an IBE scheme with efficient revocation that used a binary tree structure to reduce the key update size to logarithmic in the number of users. Their IBE scheme is proven selectively secure in the standard model. Later on, Libert and Vergnaud (2009) improved Boldyreva *et al.*'s scheme to present another IBE scheme with efficient revocation that is proven adaptively secure in the standard model. However, in both schemes, each user must keep $3 \log n$ private keys while the PKG needs to maintain a binary tree data structure of n leaf nodes, where n denotes the total number of users.

Up to now, there was little work on studying the revocation problem of HIBE systems. Certainly, all existing IBE systems can inherit the revocation method suggested by

Boneh and Franklin to revoke illegal or expired users. However, the HIBE constructions using Boneh and Franklin's revocation method will require enormous computational cost between users and their ancestors in each period because computational workload for encrypting and decrypting the new private keys are required for each period.

1.2. *Contribution*

In this paper, we will propose the first revocable HIBE (RHIBE) scheme with public revocation mechanism in the standard model, which is extended from Lewko and Waters's unbounded HIBE scheme (2011). We first present the framework of the new RHIBE construction with public revocation mechanism. For security model, we consider all adversarial capabilities of the standard HIBE security notions. We define the RHIBE's security notions based on the complete security definitions presented in Shi and Waters (2008), which keep track of how keys are generated and delegated. Following the framework of RHIBE with public revocation mechanism, we propose a concrete RHIBE scheme from bilinear pairings in the standard model. For the security of proposed RHIBE scheme, we use the identical technique of the nested dual system encryption in Lewko and Waters (2011) to prove that it is fully secure while removing the requirement of secure channels for private key updating, and remaining the merits of Lewko and Waters's HIBE scheme.

We employ a transformation technique to convert Lewko and Waters's unbounded HIBE scheme (2011) to our RHIBE scheme, in which a user's private key (decryption key) is divided into two components including a fixed initial secret key and a changed time update key along with periods. The point is that any user needs to get both the fixed initial secret key and the changed time update key to decrypt his/her ciphertext. If a receiver obtains only either the fixed initial secret key or the changed time update key, the receiver cannot get the complete private key. For increasing the flexibility of the proposed RHIBE system in revoking illegal or expired users, a delegated revocation authority (DRA) is involved into the proposed RHIBE scheme and it can assist the PKG to revoke any users in the whole system, while remaining the ability of users at high level revoking their subordinates.

Without loss of generality, we use Fig. 2 to illustrate our proposed RHIBE scheme, in which α and β are the system's secret keys. The PKG gives the DRA the secret key β via a secure channel. A new user *id* at level j can obtain its associated initial secret key D_{id} from the PKG or its ancestors via a secure channel. In the present period, a non-revoked user *id* at level j can also get the current time update key $T_{id,t}$ from the DRA or its ancestors via a public channel (e.g. E-mail system). The user *id* then can use its initial secret key D_{id} and time update key $T_{id,t}$ to compute its current decryption key. Obviously, if the DRA or its ancestors stop to issue the current time update key for the user *id* at level j, it means that the user *id* has been revoked. As compared to Boneh and Franklin's revocation method, our revocation mechanism can reduce the computational workload of encryption and decryption procedures for periodic key updating.

Although involving a DRA into the RHIBE scheme can provide the revocation flexibility, it could also increase the computational cost as compared to the original HIBE

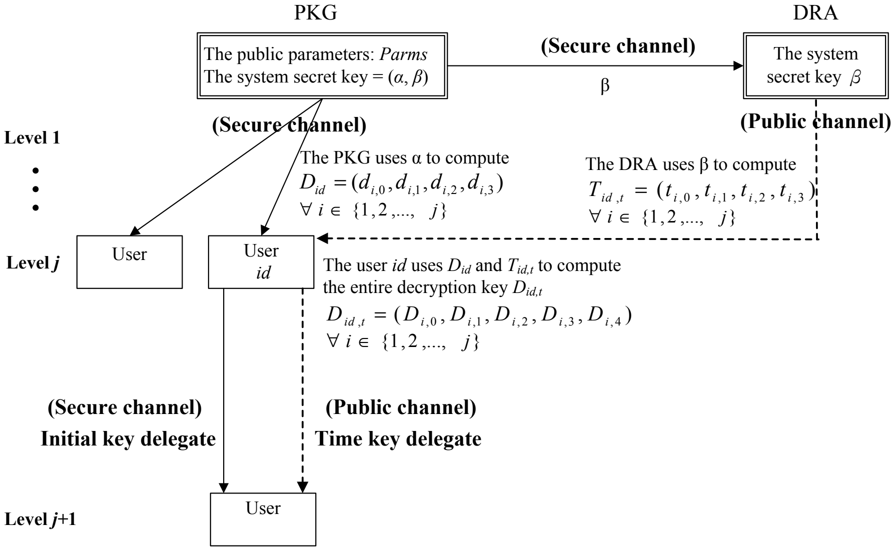
PKG
DRA
The public parameters: *Parms*
The system secret key = (α, β)
(Secure channel)
β
The system secret key β
Level 1
(Secure channel)
(Public channel)
The PKG uses α to compute
$D_{id} = (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3})$
$\forall\, i \in \{1,2,\ldots,\ j\}$
The DRA uses β to compute
$T_{id,t} = (t_{i,0}, t_{i,1}, t_{i,2}, t_{i,3})$
$\forall\, i \in \{1,2,\ldots,\ j\}$
Level *j*
User
User *id*
The user *id* uses D_{id} and $T_{id,t}$ to compute the entire decryption key $D_{id,t}$
$D_{id,t} = (D_{i,0}, D_{i,1}, D_{i,2}, D_{i,3}, D_{i,4})$
$\forall\, i \in \{1,2,\ldots,\ j\}$
(Secure channel)
Initial key delegate
(Public channel)
Time key delegate
Level *j*+1
User

Fig. 2. The decryption key generation of RHIBE.

scheme. If we remove the DRA role from the RHIBE scheme, the work of the DRA must be performed by the PKG. As a result, the system can remain the efficiency of encryption and decryption procedures. For showing the generality of the transformation technique, we also present another RHIBE scheme without a DRA in the random oracle model, which is extended from a famous HIBE scheme proposed by Gentry and Silverberg (2002). The proposed RHIBE scheme without a DRA remains the efficiency of encryption and decryption procedures as compared to Gentry and Silverberg's HIBE scheme (2002). Indeed, we believe that the transformation technique is suitable to construct RHIBE schemes with/without the DRA from some existing HIBE schemes.

Here, we summarize our concrete work and contributions in this paper:

(1) The framework and security notions of RHIBE with public revocation mechanism are defined. Meanwhile, the first RHIBE scheme is proposed and proved to be fully secure in the standard model.
(2) A DRA role is added to assist the PKG to revoke illegal or expired users in the whole system. As a result, it may provide the flexibility for revoking illegal or expired users.
(3) We present a transformation technique to convert a HIBE scheme to a RHIBE scheme. Finally, we employ it to present another concrete RHIBE scheme without a DRA in the random oracle model.

1.3. *Organization*

The remainder of the paper is organized as follows. Preliminaries are given in Section 2. In Section 3, we formally present the framework and security notions of RHIBE with public revocation mechanism. The concrete RHIBE scheme is proposed in Section 4. We analyze the security of the proposed RHIBE scheme in Section 5. We discuss the transformation technique from a HIBE scheme to a RHIBE scheme and employ it to present another RHIBE scheme in Section 6. Section 7 demonstrates comparisons and performance analysis. Conclusions are given in Section 8.

2. Preliminaries

In this section, we briefly introduce the concept of bilinear pairings and the related assumptions (Boneh *et al.*, 2005a; Lewko and Waters, 2011). Meanwhile, we review Lewko and Waters's HIBE scheme (2011) and the concept of dual system encryption.

2.1. *Bilinear Pairings*

Let G_1 and G_2 be two cyclic groups of order $N = p_1 p_2 p_3$, where p_1, p_2, p_3 are three distinct primes. In particular, the composite order bilinear group was first introduced in Boneh *et al.* (2005b). A bilinear map $\hat{e} : G_1 \times G_1 \to G_2$ must satisfy the following properties:

(1) Bilinear: $\hat{e}(P^a, Q^b) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_N^*$.
(2) Non-degenerate: there exists a value $P \in G_1$ such that $\hat{e}(P, P)$ has order N in G_2.
(3) Computable: the group operations in both G_1 and G_2, and the map $\hat{e}$ are computable in polynomial time with respect to a security parameter.

Let G_{p_i} denote the subgroup of order p_i in G_1, where $i = 1, 2, 3$, and $G_{p_i p_j}$ denote the subgroup of order $p_i p_j$ in G_1, where $i, j = 1, 2, 3$ and $i \neq j$. Note that these subgroups G_{p_i} are "orthogonal" each other under the bilinear map $\hat{e}$. That is, if $h_i \in G_{p_i}$ and $h_j \in G_{p_j}$ for $i \neq j$, then $\hat{e}(h_i, h_j)$ is the identity element in G_2. Assume that g_i is a generator of G_{p_i}, where $i = 1, 2, 3$. We then have that every element in G_1 can be expressed as $g_1^x g_2^y g_3^z$ for some values $x, y, z \in Z_N$.

2.2. *Complexity Assumptions*

Our RHIBE scheme is extended from Lewko and Waters's unbounded HIBE scheme (2011). The security of Lewko and Waters's unbounded HIBE scheme is based on special cases of the general subgroup decision assumption defined in Bellare *et al.* (2011). The security of our RHIBE scheme has identical assumptions as in Lewko and Waters (2011) except Assumption 2. We first define the notation $X \overset{R}{\leftarrow} S$ to express that X is chosen uniformly randomly from the finite set S.

Assumption 1. Let $\mathcal{G}$ be a group generator, we define the following distribution: Given $G = (N = p_1 p_2 p_3, \hat{e}, G_1, G_2) \overset{R}{\leftarrow} \mathcal{G}$, $g \overset{R}{\leftarrow} G_{p_1}$, $T_1 \overset{R}{\leftarrow} G_{p_1 p_2}$, $T_2 \overset{R}{\leftarrow} G_{p_1}$, the distribution is $D = (G, g)$. The successful probability (advantage) of the adversary $\mathcal{A}$ in breaking Assumption 1 is presented as $Adv1_{\mathcal{G},\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$, where the probability is over the random choice consumed by the probabilistic polynomial-time (PPT) adversary $\mathcal{A}$. We say that $\mathcal{G}$ satisfies Assumption 1 if $Adv1_{\mathcal{G},\mathcal{A}}$ is a negligible function for any PPT adversary $\mathcal{A}$.

Assumption 2. Let $\mathcal{G}$ be a group generator, we define the following distribution: Given $G = (N = p_1p_2p_3, \hat{e}, G_1, G_2) \overset{R}{\leftarrow} \mathcal{G}$, $g \overset{R}{\leftarrow} G_{p_1}$, $g_2, X_2, Y_2 \overset{R}{\leftarrow} G_{p_2}$, $g_3 \overset{R}{\leftarrow} G_{p_3}$, $\alpha, \beta, s \overset{R}{\leftarrow} Z_N$, $T_1 = \hat{e}(g, g)^{(\alpha+\beta)\cdot s}$, $T_2 \overset{R}{\leftarrow} G_2$, the distribution is $D = (G, g, g_2, g_3, g^{\alpha}X_2, g^{\beta}X_2, g^{s}Y_2)$. The successful probability (advantage) of the adversary $\mathcal{A}$ in breaking Assumption 2 is presented as $Adv2_{\mathcal{G},\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$, where the probability is over the random choice consumed by the PPT adversary $\mathcal{A}$. We say that $\mathcal{G}$ satisfies Assumption 2 if $Adv2_{\mathcal{G},\mathcal{A}}$ is a negligible function for any PPT adversary $\mathcal{A}$.

Assumption 3. Let $\mathcal{G}$ be a group generator, we define the following distribution: Given $G = (N = p_1p_2p_3, \hat{e}, G_1, G_2) \overset{R}{\leftarrow} \mathcal{G}$, $g, X_1 \overset{R}{\leftarrow} G_{p_1}$, $g_2 \overset{R}{\leftarrow} G_{p_2}$, $X_3 \overset{R}{\leftarrow} G_{p_3}$, $T_1 \overset{R}{\leftarrow} G_{p_1}$, $T_2 \overset{R}{\leftarrow} G_{p_1p_3}$, the distribution is $D = (G, g, g_2, X_1, X_3)$. The successful probability (advantage) of the adversary $\mathcal{A}$ in breaking Assumption 3 is presented as $Adv3_{\mathcal{G},\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$, where the probability is over the random choice consumed by the PPT adversary $\mathcal{A}$. We say that $\mathcal{G}$ satisfies Assumption 3 if $Adv3_{\mathcal{G},\mathcal{A}}$ is a negligible function for any PPT adversary $\mathcal{A}$.

Assumption 4. Let $\mathcal{G}$ be a group generator, we define the following distribution: Given $G = (N = p_1p_2p_3, \hat{e}, G_1, G_2) \overset{R}{\leftarrow} \mathcal{G}$, g, $X_1 \overset{R}{\leftarrow} G_{p_1}$, X_2, $Y_2 \overset{R}{\leftarrow} G_{p_2}$, g_3, $Y_3 \overset{R}{\leftarrow} G_{p_3}$, $T_1 \overset{R}{\leftarrow} G_{p_1p_3}$, $T_2 \overset{R}{\leftarrow} G_1$, the distribution is $D = (G, g, g_3, X_1X_2, Y_2Y_3)$. The successful probability (advantage) of the adversary $\mathcal{A}$ in breaking Assumption 4 is presented as $Adv4_{\mathcal{G},\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$, where the probability is over the random choice consumed by the PPT adversary $\mathcal{A}$. We say that $\mathcal{G}$ satisfies Assumption 4 if $Adv4_{\mathcal{G},\mathcal{A}}$ is a negligible function for any PPT adversary $\mathcal{A}$.

2.3. *Lewko and Waters's Hierarchical Identity-Based Encryption*

Lewko and Waters's HIBE scheme includes five algorithms: *System setup*, *Encryption*, *Key extract*, *Delegate* and *Decryption* algorithms. They assumed that identity vectors are encoded such that if the identity vector $id = (id_1, \ldots, id_j)$ is not a prefix of identity vector $id' = (id'_1, \ldots, id'_i)$, then $id_j \neq id'_k$ for all $k \in \{1, 2, \ldots, i\}$.

- ***System setup*:** A trusted private key generation (PKG) takes a security parameter l as input of the *System setup* algorithm. The algorithm generates two groups G_1 and G_2 of order $N = p_1p_2p_3$, where p_1, p_2, $p_3$are distinct primes, and an admissible bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$. Let G_{p_i} denote the subgroup of p_i in G_1. Let g_2 denote a generator of G_{p_2} and g_3 denote a generator of G_{p_3}. The algorithm randomly chooses g, u, h, v, w from G_{p_1}, and α from Z_N. Finally, the algorithm returns the system secret key α to the PKG and publishes the public parameters $Parms = (N, G_1, G_2, \hat{e}, g, u, h, v, w, \hat{e}(g, g)^{\alpha})$.

- ***Encryption*:** Given a message M and a receiver $id = (id_1, \ldots, id_j)$, a sender chooses random values $s, m_1, \ldots, m_j \in Z_N$ and computes the ciphertext as

$$\begin{aligned} C &= (C_0, C_1, C_{i,2}, C_{i,3}, C_{i,4}), \quad \forall i \in \{1, \ldots, j\} \\ &= \left(\hat{e}(g, g)^{\alpha \cdot s} \cdot M, g^s, w^s v^{m_i}, g^{m_i}, \left(u^{id_i} h\right)^{m_i}\right), \quad \forall i \in \{1, \ldots, j\}. \end{aligned}$$

- ***Key extract*:** Given a user's identity $id = (id_1, \ldots, id_j)$, the PKG chooses random values $r_1, \ldots, r_j, y_1, \ldots, y_j$ from Z_N and random values $\lambda_1, \ldots, \lambda_j \in Z_N$ subject to the constraint that $\alpha = \lambda_1 + \lambda_2 + \cdots + \lambda_j$. Finally, the PKG computes the private key $D_{id} = (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3}) = (g^{\lambda_i} w^{y_i}, g^{y_i}, v^{y_i} (u^{id_i} h)^{r_i}, g^{r_i})$, $\forall i \in \{1, \ldots, j\}$ and transmits D_{id} to the user via a secure channel.
- ***Delegate*:** Given a private key $D_{id} = (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3})$, $\forall i \in \{1, \ldots, j\}$ for a user's identity $id = (id_1, \ldots, id_j)$ and a level $j + 1$ identity $id' = (id_1, \ldots, id_j, id_{j+1})$, this algorithm chooses random values $r_1], \ldots, r'_j, r'_{j+1}, y'_1, \ldots, y'_j, y'_{j+1}$ from Z_N and random values $\lambda'_1, \ldots, \lambda'_j, \lambda'_{j+1} \in Z_N$ subject to the constraint that $\lambda'_1 + \lambda'_2 + \cdots + \lambda'_j + \lambda'_{j+1} = 0$. Finally, this algorithm computes the secret key $D'_{id} = (d'_{i,0}, d'_{i,1}, d'_{i,2}, d'_{i,3}) = (d_{i,0} \cdot g^{\lambda'_i} w^{y'_i}, d_{i,1} \cdot g^{y'_i}, d_{i,2} \cdot v^{y'_i} (u^{id_i} h)^{r'_i}, d_{i,3} \cdot g^{r'_i})$, $\forall i \in \{1, \ldots, j + 1\}$, where $d_{j+1,0}, d_{j+1,1}, d_{j+1,2}$ and $d_{j+1,3}$ are defined to be the identity element in G_1.
- ***Decryption*:** Given a ciphertext $C = (C_0, C_1, C_{i,2}, C_{i,3}, C_{i,4})$, $\forall i \in \{1, \ldots, j\}$, the receiver can use the private key $D_{id} = (D_{i,0}, D_{i,1}, D_{i,2}, D_{i,3})$, $\forall i \in \{1, \ldots, j\}$ for $id = (id_1, \ldots, id_j)$ to decrypt C as follows:

$$\begin{aligned} \text{Compute } B &= \prod_{i=1}^{j} \frac{\hat{e}(C_1, D_{i,0}) \cdot \hat{e}(C_{i,3}, D_{i,2})}{\hat{e}(C_{i,2}, D_{i,1}) \cdot \hat{e}(C_{i,4}, D_{i,3})}, \quad \forall i \in \{1, \ldots, j\} \\ &= \hat{e}(g, g)^{\alpha \cdot s}, \end{aligned}$$

then

$$M = \frac{C_0}{B} = \frac{\hat{e}(g, g)^{\alpha \cdot s} \cdot M}{\hat{e}(g, g)^{\alpha \cdot s}}.$$

Lewko and Waters (2011) used the nested dual system encryption argument (Waters, 2009) to prove full security of their HIBE scheme. Beside the five algorithms above, a dual system encryption HIBE scheme has two extra algorithms *Encryption SF* and *Key extract SF*, which produces semi-functional ciphertexts and keys, respectively. Note that the *encryption SF* and *key extract SF* algorithms are used only for the security proof and not for the normal operation of the scheme.

- ***Encryption SF*:** The semi-functional *encryption algorithm* first calls the encryption algorithm to obtain a normal ciphertext $C = (C_0, C_1, C_{i,2}, C_{i,3}, C_{i,4})$, $\forall i \in \{1, \ldots, j\}$. This algorithm chooses random values γ, $\delta \in Z_N$ and computes the semi-functional ciphertext as

$$\begin{aligned} \tilde{C} &= (\tilde{C}_0, \tilde{C}_1, \tilde{C}_{i,2}, \tilde{C}_{i,3}, \tilde{C}_{i,4}), \quad \forall i \in \{1, \ldots, j\} \\ &= (C_0, C_1 \cdot g_2^{\gamma}, C_{i,2} \cdot g_2^{\delta}, C_{i,3}, C_{i,4}), \quad \forall i \in \{1, \ldots, j\}. \end{aligned}$$

- ***Key extract SF*:** The semi-functional key extract algorithm first calls the *key extract algorithm* to obtain a normal private key $D_{id} = (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3}), \forall i \in \{1, \dots, j\}$. This algorithm chooses random values ζ, η, $\hat{y} \in Z_N$ and computes the semi-functional private key as

$$\begin{aligned}\tilde{D}_{id} &= (\tilde{d}_{i,0}, \tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,3}), \quad \text{for } i \in \{1, \dots, j\},\\ &= \begin{cases} (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3}), & \text{if } i \in \{1, \dots, j-1\},\\ (d_{i,0} \cdot (g_2 g_3)^{\eta \cdot \hat{y}_j}, d_{i,1} \cdot (g_2 g_3)^{\hat{y}_j}, d_{i,2} \cdot (g_2 g_3)^{\zeta \cdot \hat{y}_j}, d_{i,3}), & \text{if } i = j. \end{cases}\end{aligned}$$

In the dual system encryption HIBE scheme, ciphertexts and keys can take two forms: normal and semi-functional. By running *Encryption* and *Key extract* algorithms, we can get the normal ciphertexts and keys, respectively. By running *Encryption SF* and *Key extract SF* algorithms, we can get the semi-functional ciphertexts and keys, respectively. Normal keys can decrypt both normal and semi-functional ciphertexts, while semi-functional keys can decrypt only normal ciphertexts.

3. Framework and Security Notions of RHIBE

In this section, we formally define the framework and security notions of revocable hierarchical ID-based encryption (RHIBE) with public revocation mechanism. Following the framework and security notions of HIBE presented by Lewko and Waters (2011), we redefine the framework of RHIBE by adding Time key update and Time key delegate algorithms. We define the new framework and security notions of RHIBE in the following subsections.

3.1. *Framework of RHIBE*

A RHIBE scheme consists of seven algorithms: *System setup*, *Encryption*, *Initial key extract*, *Initial key delegate*, *Time key update*, *Time key delegate* and *Decryption* algorithms:

- ***System setup*:** This algorithm is a probabilistic algorithm which takes as input a security parameter l and the total number $\mathcal{T}$ of all periods, it returns a system secret key α, a time secret key β and the public parameters *Parms*. The public parameters *Parms* are made public and implicitly inputted to all the following algorithms.
- ***Encryption*:** Take an identity vector *id* and a message M as input, the algorithm generates a ciphertext C.
- ***Initial key extract*:** Take the system secret key α and a user's identity vector *id* as input, the algorithm returns the user's initial secret key D_{id}.
- ***Initial key delegate*:** Take the initial secret key D_{id} for the identity vector $id = (id_1, \dots, id_j)$ and a user's identity $id' = (id_1, \dots, id_j, id_{j+1})$ as input, the algorithm returns the user's initial secret key D'_{id} for the identity vector id'. Here, the vector $id = (id_1, \dots, id_j)$ is the user's identity at level j and the vector $id' = (id_1, \dots, id_j, id_{j+1})$ is the user's identity at level $j + 1$.

- ***Time key update*:** For a period t, take the time secret key β and a user's identity vector id as input, the algorithm returns the user's time update key $T_{id,t}$. Note that the non-revoked user can use the initial secret key D_{id} and the time update key $T_{id,t}$ to obtain the decryption key $D_{id,t}$.
- ***Time key delegate*:** For a period t, take the time update key $T_{id,t}$ for identity vector $id = (id_1, \dots, id_j)$ and a user's identity $id' = (id_1, \dots, id_j, id_{j+1})$ as input, the algorithm returns the user's time update key $T_{id',t}$ for identity vector id'.
- ***Decryption*:** The algorithm takes a ciphertext C and the user's decryption key $D_{id,t}$ as input. If the identity vector of the secret key id is a prefix of the identity vector used to encrypt the ciphertext, and the key and ciphertext are not both semi-functional, the algorithm returns a plaintext M.

As mentioned in Section 2.3, we also use the nested dual system encryption argument to prove full security of the dual system encryption RHIBE scheme. We define three semi-functional algorithms as follows:

- ***Encryption SF*:** Take an identity vector id and a message M as input, the algorithm generates a semi-functional ciphertext $\tilde{C}$.
- ***Initial key extract SF*:** Take the system secret key α and a user's identity vector id as input, the algorithm returns the user's semi-functional initial secret key $\tilde{D}_{id}$.
- ***Time key update SF***: For a period t, take the time secret key β and a user's identity vector id as input, the algorithm returns the user's semi-functional time update key $\tilde{T}_{id,t}$.

3.2. *Security Model of RHIBE*

We extend the security definition of Lewko and Waters's HIBE scheme (2011) to define the security game of RHIBE scheme, called Game RHIBE. We also use the complete security definition presented in Shi and Waters (2008), which keeps track of how keys are generated and delegated.

Definition 1. We say that a RHIBE scheme is secure if no PPT adversary $\mathcal{A}$ has a non-negligible advantage in the following Game RHIBE played with a challenger $\mathcal{B}$.

- *Phase* 1. The challenger $\mathcal{B}$ runs *System setup* algorithm of RHIBE to generate a system secret key α and a time secret key β, and produce the public parameters *Parms*. Then the challenger $\mathcal{B}$ gives the adversary $\mathcal{A}$ the *Parms* and keeps the system secret key α and time secret key β to itself. We let S and V denote the sets of initial secret keys and time update keys, respectively, that the challenger has created but not yet given to the adversary. We initialize $S = \Phi$ and $V = \Phi$.
- *Phase* 2. The adversary $\mathcal{A}$ may issue a number of different queries to $\mathcal{B}$ as follows:
  - *Initial key create query*. Upon receiving this query with an identity vector $id = (id_1, \dots, id_j)$ at level j, the challenger $\mathcal{B}$ runs *Initial key extract* algorithm to generate the initial secret key D_{id} and places this key in the set S. The challenger $\mathcal{B}$ gives the adversary $\mathcal{A}$ only a reference to this initial secret key D_{id}, not the key D_{id} itself.

- *Initial key delegate query*. Upon receiving this query with an initial secret key D_{id} in the set S and an identity vector $id' = (id_1, \dots, id_j, id_{j+1})$ at level $j+1$, the challenger $\mathcal{B}$ runs *Initial key delegate* algorithm to generate an initial secret key $D_{id'}$ for the identity vector id'. The challenger $\mathcal{B}$ adds this key to the set S and gives the adversary $\mathcal{A}$ only a reference not the actual key.
- *Initial key reveal query*. Upon receiving this query with an element of the set S, the challenger $\mathcal{B}$ gives this initial secret key to the adversary $\mathcal{A}$ and removes it from the set S. We note that the adversary $\mathcal{A}$ needs no longer to make any initial key delegation queries for this initial secret key because it can run *Initial key delegate* algorithm on the revealed key for itself.
- *Time key create query*. Upon receiving this query with an identity vector $id = (id_1, \dots, id_j)$ at level j for a period t, the challenger $\mathcal{B}$ runs *Time key update* algorithm to generate the time update key $T_{id,t}$ and places this key in the set V. The challenger $\mathcal{B}$ gives the adversary $\mathcal{A}$ only a reference to this time update key $T_{id,t}$, not the key$T_{id,t}$ itself.
- *Time key delegate query*. Upon receiving this query with a time update key $T_{id,t}$ in the set V and an identity vector $id' = (id_1, \dots, id_j, id_{j+1})$ at level $j+1$ for a period t, the challenger $\mathcal{B}$ runs *Time key delegate* algorithm to generate time update key $T_{id',t}$ for this new identity vector id'. The challenger $\mathcal{B}$ adds this key to the set V and gives the adversary $\mathcal{A}$ only a reference not the actual key.
- *Time key reveal query*. Upon receiving this query with an element of the set V, the challenger $\mathcal{B}$ gives this time update key to the adversary $\mathcal{A}$ and removes it from the set V. We note that the adversary $\mathcal{A}$ needs no longer to make any time key delegate queries for this time update key because it can run *Time key delegate* algorithm on the revealed key for itself.

- *Phase* 3. The adversary $\mathcal{A}$ gives a target identity vector id^*, a period t^* and a plaintext pair (M_0^*, M_1^*) to $\mathcal{B}$. A restriction here is that either id^* or (id^*, t^*) did not appear in *Phase* 2. This identity vector must satisfy the property that no revealed identity in *Phase* 2 was a prefix of it. Then the challenger $\mathcal{B}$ chooses a random $\mathfrak{b} \in \{0, 1\}$ and computes C^* by running the *Encryption algorithm*. Then $\mathcal{B}$ sends C^* to $\mathcal{A}$.
- *Phase* 4. The adversary $\mathcal{A}$ may issue more queries as in *Phase* 2. A restriction here is that either id^* or (id^*, t^*) is disallowed to be queried in *Phase* 2. Any revealed identity in *Phase* 2 was not a prefix of id^*.
- *Phase* 5. The adversary $\mathcal{A}$ outputs $\mathfrak{b}' \in \{0, 1\}$ and wins this game if $\mathfrak{b}' = \mathfrak{b}$.

We define the adversary $\mathcal{A}$'s advantage in attacking the RHIBE scheme in the security game as $Adv_{\mathcal{A}}^{\text{RHIBE}}(l) = |\Pr[\mathfrak{b} = \mathfrak{b}'] - 1/2|$.

4. Concrete Dual System Encryption RHIBE Scheme

In this section, we extend Lewko and Waters's dual system encryption HIBE scheme (2011) to propose the concrete dual system encryption RHIBE scheme. We involve a delegated revocation authority (DRA) into our RHIBE scheme, which is different from the dual system encryption HIBE scheme. In the proposed RHIBE scheme, there are three roles: a trusted PKG, a trusted DRA and users. The trusted PKG can delegate the time secret key to the DRA, which can assist the PKG to revoke misbehaving/compromised users from the system. It is obvious that the PKG can also perform the same work.

Our proposed RHIBE scheme consists of seven algorithms: *System setup, Encryption, Initial key extract, Initial key delegate, Time key update, Time key delegate* and *Decryption algorithms*.

- ***System setup*:** A trusted PKG takes a security parameter l and the total number $\mathcal{T}$ of periods as input. The PKG generates two groups G_1 and G_2 of order $N = p_1 p_2 p_3$, where p_1, p_2, p_3 are distinct primes, and a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$. Let G_{p_i} denote the subgroup of p_i in G_1. Let g_2 denote a generator of G_{p_2} and g_3 denote a generator of G_{p_3}. The PKG randomly chooses g, u, h, v, w, z, f from G_{p_1}, and α, β from Z_N. Finally, the PKG keeps the system secret key α and the time secret key β for itself, and publishes the public parameters $Parms = (N, G_1, G_2, \hat{e}, g, u, h, v, w, z, f, \hat{e}(g, g)^{\alpha}, \hat{e}(g, g)^{\beta})$. Then, the PKG transmits the time secret key β to the DRA via a secure channel.
- ***Encryption*:** For a period t, given a message M and a receiver $id = (id_1, \dots, id_j)$, a sender chooses random values $s, m_1, \dots, m_j \in Z_N$ and computes the ciphertext as

$$\begin{aligned} C &= (C_0, C_1, C_{i,2}, C_{i,3}, C_{i,4}, C_{i,5}), \quad \forall i \in \{1, \dots, j\} \\ &= \left(\hat{e}(g, g)^{(\alpha+\beta)s} \cdot M, g^s, w^s v^{m_i}, g^{m_i}, \left(u^{id_i} h\right)^{m_i}, \left(z^{t \cdot id_i} f\right)^{m_i}\right), \quad \forall i \in \{1, \dots, j\}. \end{aligned}$$

- ***Initial key extract*:** Given a user's identity $id = (id_1, \dots, id_j)$, the PKG chooses random values $r_1, \dots, r_j, y_1, \dots, y_j$ from Z_N and random values $\lambda_1, \dots, \lambda_j \in Z_N$ subject to the constraint that $\alpha = \lambda_1 + \lambda_2 + \cdots + \lambda_j$. Finally, the PKG computes the initial secret key $D_{id} = (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3}) = (g^{\lambda_i} w^{y_i}, g^{y_i}, v^{y_i} (u^{id_i} h)^{r_i}, g^{r_i})$, $\forall i \in \{1, \dots, j\}$ and transmits D_{id} to the user via a secure channel.
- ***Initial key delegate*:** The algorithm takes input an initial secret key $D_{id} = (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3})$, $\forall i \in \{1, \dots, j\}$ for a user's identity $id = (id_1, \dots, id_j)$ and a level $j + 1$ identity $id' = (id_1, \dots, id_j, id_{j+1})$. Then, it chooses random values $r'_1, \dots, r'_j, r'_{j+1}, y'_1, \dots, y'_j, y'_{j+1}$ from Z_N and random values $\lambda'_1, \dots, \lambda'_j, \lambda'_{j+1} \in Z_N$ subject to the constraint that $\lambda'_1 + \lambda'_2 + \cdots + \lambda'_j + \lambda'_{j+1} = 0$. Finally, this algorithm computes the initial secret key $D'_{id} = (d'_{i,0}, d'_{i,1}, d'_{i,2}, d'_{i,3}) = (d_{i,0} \cdot g^{\lambda'_i} w^{y'_i}, d_{i,1} \cdot g^{y'_i}, d_{i,2} \cdot v^{y'_i} (u^{id_i} h)^{r'_i}, d_{i,3} \cdot g^{r'_i})$, $\forall i \in \{1, \dots, j + 1\}$, where $d_{j+1,0}, d_{j+1,1}, d_{j+1,2}$ and $d_{j+1,3}$ are defined to be the identity element in G_1.
- ***Time key update*:** Given a user's identity $ID = (id_1, \dots, id_j)$ and a period $t \in [1, \mathcal{T}]$, the DRA chooses random values $k_1, \dots, k_j, x_1, \dots, x_j$ from Z_N and random values $\mu_1, \dots, \mu_j \in Z_N$ subject to the constraint $\beta = \mu_1 + \mu_2 + \cdots + \mu_j$. Finally, the PKG computes the time update key $T_{id,t} = (t_{i,0}, t_{i,1}, t_{i,2}, t_{i,3}) = (g^{\mu_i} w^{x_i}, g^{x_i}, v^{x_i} (z^{t \cdot id_i} f)^{k_i}, g^{k_i})$, $\forall i \in \{1, \dots, j\}$. Then the DRA transmits $T_{id,t}$ to the user via a public channel (e.g., E-mail system). Thus, the user can use D_{id} and $T_{id,t}$ to compute the decryption key for the period t as

$$\begin{aligned} D_{id,t} &= (D_{i,0}, D_{i,1}, D_{i,2}, D_{i,3}, D_{i,4}) \\ &= (d_{i,0} \cdot t_{i,0}, d_{i,1} \cdot t_{i,1}, d_{i,2} \cdot t_{i,2}, d_{i,3}, t_{i,3}) \\ &= \left(g^{\lambda_i+\mu_i} w^{y_i+x_i}, g^{y_i} \cdot g^{x_i}, v^{y_i}\left(u^{id_i}h\right)^{r_i} \cdot v^{x_i}\left(z^{t\cdot id_i}f\right)^{k_i}, g^{r_i}, g^{k_i}\right), \\ &\forall i \in \{1, \ldots, j\}. \end{aligned}$$

- ***Time key delegate*:** The algorithm takes input a time update key $T_{id,t} = (t_{i,0}, t_{i,1}, t_{i,2}, t_{i,3})$, $\forall i \in \{1, \ldots, j\}$ for user's identity $id = (id_1, \ldots, id_j)$ for a period t and a level $j + 1$ identity $id' = (id_1, \ldots, id_j, id_{j+1})$. Then, it chooses random values $k'_1, \ldots, k'_j, k'_{j+1}, x'_1, \ldots, x'_j, x'_{j+1}$ from Z_N and random values $\mu'_1, \ldots, \mu'_j, \mu'_{j+1} \in Z_N$ subject to the constraint that $\mu'_1 + \mu'_2 + \cdots + \mu'_j + \mu'_{j+1} = 0$. Finally, this algorithm computes the time update key $T'_{id,t} = (t'_{i,0}, t'_{i,1}, t'_{i,2}, t'_{i,3}) = (t_{i,0} \cdot g^{\mu'_i} w^{x'_i}, t_{i,1} \cdot g^{x'_i}, t_{i,2} \cdot v^{x'_i}(z^{t\cdot id_i} f)^{k'_i}, t_{i,3} \cdot g^{k'_i})$, $\forall i \in \{1, \ldots, j+1\}$, where $t_{j+1,0}, t_{j+1,1}, t_{j+1,2}$ and $t_{j+1,3}$ are defined to be the identity element in G_1.
- ***Decryption*:** Given a ciphertext $C = (C_0, C_1, C_{i,2}, C_{i,3}, C_{i,4}, C_{i,5})$, $\forall i \in \{1, \ldots, j\}$, the receiver can use the private key $D_{id,t} = (D_{i,0}, D_{i,1}, D_{i,2}, D_{i,3}, D_{i,4})$, $\forall i \in \{1, \ldots, j\}$ for $id = (id_1, \ldots, id_j)$ in a period t to decrypt C as follows:

$$\begin{aligned} B &= \prod_{i=1}^{j} \frac{\hat{e}(C_1, D_{i,0}) \cdot \hat{e}(C_{i,3}, D_{i,2})}{\hat{e}(C_{i,2}, D_{i,1}) \cdot \hat{e}(C_{i,4}, D_{i,3}) \cdot \hat{e}(C_{i,5}, D_{i,4})} \\ &= \prod_{i=1}^{j} \frac{\hat{e}(g^s, g^{\lambda_i+\mu_i} w^{y_i+x_i}) \cdot \hat{e}(g^{m_i}, v^{y_i}(u^{id_i}h)^{r_i} \cdot v^{x_i}(z^{t\cdot id_i}f)^{k_i})}{\hat{e}(w^s v^{m_i}, g^{y_i} \cdot g^{x_i}) \cdot \hat{e}((u^{id_i}h)^{m_i}, g^{r_i}) \cdot \hat{e}((z^{t\cdot id_i}f)^{m_i}, g^{k_i})} \\ &= \prod_{i=1}^{j} \frac{\hat{e}(g^s, g^{\lambda_i+\mu_i} w^{y_i+x_i}) \cdot \hat{e}(g^{m_i}, v^{y_i}(u^{id_i}h)^{r_i}) \cdot \hat{e}(g^{m_i}, v^{x_i}(z^{t\cdot id_i}f)^{k_i})}{\hat{e}(w^s v^{m_i}, g^{y_i}) \cdot \hat{e}(w^s v^{n_i}, g^{x_i}) \cdot \hat{e}((u^{id_i}h)^{r_i}, g^{m_i}) \cdot \hat{e}((z^{t\cdot id_i}f)^{k_i}, g^{m_i})} \\ &= \prod_{i=1}^{j} \frac{\hat{e}(g^s, g^{\lambda_i+\mu_i} w^{y_i+x_i}) \cdot \hat{e}(g^{m_i}, v^{y_i}) \cdot \hat{e}(g^{m_i}, (u^{id_i}h)^{r_i}) \cdot \hat{e}(g^{m_i}, v^{x_i}) \cdot \hat{e}(g^{m_i}, (z^{t\cdot id_i}f)^{k_i})}{\hat{e}(w^s v^{m_i}, g^{y_i}) \cdot \hat{e}(w^s v^{n_i}, g^{x_i}) \cdot \hat{e}((u^{id_i}h)^{r_i}, g^{m_i}) \cdot \hat{e}((z^{t\cdot id_i}f)^{k_i}, g^{m_i})} \\ &= \prod_{i=1}^{j} \frac{\hat{e}(g^s, g^{\lambda_i+\mu_i} w^{y_i+x_i}) \cdot \hat{e}(g^{m_i}, v^{y_i}) \cdot \hat{e}(g^{m_i}, v^{x_i})}{\hat{e}(w^s v^{m_i}, g^{y_i}) \cdot \hat{e}(w^s v^{n_i}, g^{x_i})} \\ &= \prod_{i=1}^{j} \frac{\hat{e}(g^s, g^{\lambda_i+\mu_i}) \cdot \hat{e}(g^s, w^{y_i+x_i}) \cdot \hat{e}(g^{m_i}, v^{y_i}) \cdot \hat{e}(g^{m_i}, v^{x_i})}{\hat{e}(w^s, g^{y_i}) \cdot \hat{e}(v^{m_i}, g^{y_i}) \cdot \hat{e}(w^s, g^{x_i}) \cdot \hat{e}(v^{m_i}, g^{x_i})} \\ &= \prod_{i=1}^{j} \frac{\hat{e}(g^s, g^{\lambda_i+\mu_i}) \cdot \hat{e}(g^s, w^{y_i+x_i})}{\hat{e}(w^s, g^{y_i+x_i})} \\ &= \prod_{i=1}^{j} \hat{e}(g^s, g^{\lambda_i+\mu_i}) \\ &= \hat{e}(g, g)^{(\alpha+\beta)s}, \end{aligned}$$

then

$$M = \frac{C_0}{B} = \frac{\hat{e}(g, g)^{(\alpha+\beta)s} \cdot M}{\hat{e}(g, g)^{(\alpha+\beta)s}}.$$

As defined in Section 3.1, we need three semi-functional algorithms (*Encryption SF, Initial key extract SF* and *Time key update SF*) for the proof of the full security in our dual system encryption RHIBE scheme. *Encryption SF, Initial key extract SF* and *Time key update SF* algorithms produce semi-functional ciphertexts, initial secret keys and time update keys, respectively. Note that three algorithms are used only for the security proof and not for the normal operation of the system.

- ***Encryption SF*:** For a period t, the semi-functional encryption algorithm first calls the Encryption algorithm to obtain a normal ciphertext $C = (C_0, C_1, C_{i,2}, C_{i,3}, C_{i,4}, C_{i,5})$, $\forall i \in \{1, \ldots, j\}$. This algorithm then chooses random values $\gamma, \delta \in Z_N$ and computes a semi-functional ciphertext as

$$\begin{aligned}\tilde{C} &= (\tilde{C}_0, \tilde{C}_1, \tilde{C}_{i,2}, \tilde{C}_{i,3}, \tilde{C}_{i,4}, \tilde{C}_{i,5}), \quad \forall i \in \{1, \ldots, j\}\\ &= (C_0, C_1 \cdot g_2^{\gamma}, C_{i,2} \cdot g_2^{\delta}, C_{i,3}, C_{i,4}, C_{i,5}), \quad \forall i \in \{1, \ldots, j\}.\end{aligned}$$

- ***Initial key extract SF*:** The semi-functional initial key extract algorithm first calls the initial key extract algorithm to obtain a normal initial secret key $D_{id} = (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3})$, $\forall i \in \{1, \ldots, j\}$. This algorithm chooses random values $\zeta, \eta, \hat{y}_j \in Z_N$ and computes the semi-functional initial secret key as

$$\begin{aligned}\tilde{D}_{id} &= (\tilde{d}_{i,0}, \tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,3}), \quad \text{for } i \in \{1, \ldots, j\}\\ &= \begin{cases}(d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3}), & \text{if } i \in \{1, \ldots, j-1\},\\ (d_{i,0} \cdot (g_2g_3)^{\eta\cdot\hat{y}_j}, d_{i,1} \cdot (g_2g_3)^{\hat{y}_j}, d_{i,2} \cdot (g_2g_3)^{\zeta\cdot\hat{y}_j}, d_{i,3}), & \text{if } i = j.\end{cases}\end{aligned}$$

- ***Time key update SF*:** The semi-functional time key update algorithm first calls the *time key update algorithm* to obtain a normal time update key $T_{id,t} = (t_{i,0}, t_{i,1}, t_{i,2}, t_{i,3})$, $\forall i \in \{1, \ldots, j\}$. This algorithm chooses random values $\sigma, \omega, \hat{x}_j \in Z_N$ and computes the semi-functional time update key as

$$\begin{aligned}\tilde{T}_{id,t} &= (\tilde{t}_{i,0}, \tilde{t}_{i,1}, \tilde{t}_{i,2}, \tilde{t}_{i,3}), \quad \text{for } i \in \{1, \ldots, j\}\\ &= \begin{cases}(t_{i,0}, t_{i,1}, t_{i,2}, t_{i,3}), & \text{if } i \in \{1, \ldots, j-1\},\\ (t_{i,0} \cdot (g_2g_3)^{\omega\cdot\hat{x}_j}, t_{i,1} \cdot (g_2g_3)^{\hat{x}_j}, t_{i,2} \cdot (g_2g_3)^{\sigma\cdot\hat{x}_j}, t_{i,3}), & \text{if } i = j.\end{cases}\end{aligned}$$

5. Security Analysis

Lewko and Waters (2011) used the nested dual system encryption argument (Waters, 2009) to prove full security of their HIBE scheme. We will use the same technique to prove our proposed RHIBE scheme. As similar to the proof of Lewko and Waters's dual system encryption HIBE scheme, we first define several variations of Game RHIBE.

Game RHIBE$_{WD}$: Game RHIBE$_{WD}$ is the same as Game RHIBE, except without delegation (initial key delegate and time key delegate). For initial key delegation, instead of making initial key create, initial key delegate and initial key reveal queries, the adversary simply makes initial key extract queries to the challenger. For time key delegation, instead of making time key create, time key delegate, and time key reveal queries, the adversary simply makes time key update queries. The only restriction is that no queried identity vectors can be prefixes of the challenge identity vector provided for the challenge ciphertext.

Game RHIBE$_C$: Game RHIBE$_C$ is the same as Game RHIBE$_{WD}$, except that the challenge ciphertext is generated by a call to *Encrypt SF* algorithm instead of *Encrypt* algorithms (i.e. a semi-functional ciphertext is given to the adversary).

Game RHIBE$_{SF}$: Game RHIBE$_{SF}$ is the same as Game RHIBE$_C$, except that the challenger replaces all *Initial key extract* and *Time key update* calls with calls to *Initial key extract SF* and *Time key update SF* algorithms, respectively. In other words, the challenge ciphertext, all the initial secret keys and all the time update keys given to the adversary will be semi-functional.

As similar to four security properties of Lewko and Waters's dual system encryption HIBE scheme, now we also define four security properties for our dual system encryption RHIBE as follows.

Initial Key and Time Key Delegation Invariance: We say that a dual system encryption RHIBE scheme Π_D has initial key and time key delegation invariance if for any PPT adversary $\mathcal{A}$, there exists another PPT adversary $\mathcal{A}'$ such that the advantage of $\mathcal{A}$ in Game RHIBE is negligibly close to advantage of $\mathcal{A}'$ in Game RHIBE$_{WD}$. We denote this as follows:

$$\left|Adv_{\mathcal{A}}^{\mathrm{RHIBE}}(l) - Adv_{\mathcal{A}'}^{\mathrm{RHIBE}_{WD}}(l)\right| = negl(l).$$

Semi-Functional Ciphertext Invariance: We say that a dual system encryption RHIBE scheme Π_D has semi-functional ciphertext invariance if for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in Game RHIBE$_{WD}$ is negligibly close to its advantage in Game RHIBE$_C$. We denote this as follows:

$$\left|Adv_{\mathcal{A}}^{\mathrm{RHIBE}_{WD}}(l) - Adv_{\mathcal{A}}^{\mathrm{RHIBE}_{C}}(l)\right| = negl(l).$$

Semi-Functional Initial Key and Time Key Invariance: We say that a dual system encryption RHIBE scheme Π_D has semi-functional initial key and time key invariance if for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in Game RHIBE$_C$ is negligibly close to its advantage in Game RHIBE$_{SF}$. We denote this as follows:

$$\left|Adv_{\mathcal{A}}^{\mathrm{RHIBE}_{C}}(l) - Adv_{\mathcal{A}}^{\mathrm{RHIBE}_{SF}}(l)\right| = negl(l).$$

Semi-Functional Security: We say that a dual system encryption RHIBE scheme Π_D has semi-functional security if for any PPT adversary $\mathcal{A}$, the advantages of $\mathcal{A}$ in Game RHIBE$_{SF}$ is negligible. We denote this as follows:

$$Adv_{\mathcal{A}}^{\mathrm{RHIBE}_{SF}}(l) = negl(l).$$

Theorem 1. *If a dual system encryption RHIBE scheme* $\Pi_D =$ (*System setup, Encryption, Encryption SF, Initial key extract, Initial key extract SF, Initial key delegate, Time key update, Time key update SF, Time key delegate and Decryption*) *has initial key and time key delegation invariance, semi-functional ciphertext invariance, semi-functional initial key and time key invariance and semi-functional security, then* $\Pi =$ (*System setup, Encryption, Initial key extract, Initial key delegate, Time key update, Time key delegate and Decryption*) *is a secure RHIBE scheme.*

Proof. Assume that an adversary $\mathcal{A}$ is a PPT adversary and there are no calls to the semi-functional algorithms *Encryption SF, Initial key extract SF, Time key update SF* of Π_D in the real RHIBE game. Hence, from the adversary $\mathcal{A}$'s perspective, the adversary $\mathcal{A}$ plays the RHIBE game with Π_D is the same as plays the RHIBE game with Π which is presented as follows. By initial key and time key delegation invariance, semi-functional ciphertext invariance, semi-functional initial key and time key invariance, we have

$$\left|Adv_{\mathcal{A}}^{\mathrm{RHIBE}}(l) - Adv_{\mathcal{A}'}^{\mathrm{RHIBE}_{WD}}(l)\right| = negl(l),$$
$$\left|Adv_{\mathcal{A}'}^{\mathrm{RHIBE}_{WD}}(l) - Adv_{\mathcal{A}'}^{\mathrm{RHIBE}_{C}}(l)\right| = negl(l),$$
$$\left|Adv_{\mathcal{A}'}^{\mathrm{RHIBE}_{C}}(l) - Adv_{\mathcal{A}'}^{\mathrm{RHIBE}_{SF}}(l)\right| = negl(l).$$

Thus, by the triangle inequality, we may conclude that

$$\left|Adv_{\mathcal{A}}^{\mathrm{RHIBE}}(l) - Adv_{\mathcal{A}'}^{\mathrm{RHIBE}_{SF}}(l)\right| = negl(l).$$

The quantity $Adv_{\mathcal{A}}^{\mathrm{RHIBE}}(l)$ must be negligible in the above triangle inequality, because semi-functional security is an existing property that implies the quantity $Adv_{\mathcal{A}'}^{\mathrm{RHIBE}_{SF}}(l)$ is negligible. Thus, we can say that the RHIBE scheme Π is secure. □

In the following, we will give four lemmas to prove that our dual system encryption RHIBE scheme $\Pi_D =$ (*System setup, Encryption, Encryption SF, Initial key extract, Initial key extract SF, Initial key delegate, Time key update, Time key update SF, Time key delegate and Decryption*) has four security properties that include initial key and time key delegation invariance, semi-functional ciphertext invariance, semi-functional initial key and time key invariance, as well as semi-functional security. We omit the complete proofs of some lemmas here, since they are very similar to the proofs of Lewko and Waters's dual system encryption RHIBE (2011).

Lemma 1. *Our dual system encryption RHIBE scheme has initial key and time key delegation invariance.*

Proof. For initial key delegation invariance, *Initial key* delegate algorithm additively uses random values $r'_1, \dots, r'_j, r'_{j+1}, y'_1, \dots, y'_j, y'_{j+1}, \lambda'_1, \dots, \lambda'_j, \lambda'_{j+1} \in Z_N$ subject to the constraint that $\lambda'_1 + \lambda'_2 + \cdots + \lambda'_j + \lambda'_{j+1} = 0$. It is obvious that the distribution of this initial secret key obtained through any sequence of delegations is the same as the distribution of the initial secret key for the same identity vector generated by a direct call to *Initial key* extract algorithm. For time key delegation invariance, *Time key delegate* algorithm additively uses random values $k'_1, \dots, k'_j, k'_{j+1}, x'_1, \dots, x'_j, x'_{j+1}, \mu'_1, \dots, \mu'_j, \mu'_{j+1} \in Z_N$ subject to the constraint that $\mu'_1 + \mu'_2 + \cdots + \mu'_j + \mu'_{j+1} = 0$. It is also obvious that the distribution of this time update key obtained through any sequence of delegations is the same as the distribution of the time update key for the same identity vector and period generated by a direct call to *Time key update* algorithm.

For any PPT adversary $\mathcal{A}$ in Game RHIBE, we can define a PPT adversary $\mathcal{A}'$ in Game RHIBE^{WD} that obtains the same advantage. When $\mathcal{A}$ makes initial key create, initial key delegate, time key create and time key delegate queries, $\mathcal{A}'$ makes no query. If $\mathcal{A}$ makes initial key reveal queries, then $\mathcal{A}'$ makes initial key extract queries for the same identity. When $\mathcal{A}$ makes time key reveal queries, then $\mathcal{A}'$ makes time key update queries for the same identity and period. Since the initial secret keys and the time update keys that $\mathcal{A}'$ obtains have the same distribution as those keys that $\mathcal{A}$ obtains, their advantages are identical. □

Lemma 2. *Under Assumption* 1*, our dual system encryption RHIBE scheme has semi-functional ciphertext invariance.*

Proof. Assume that there exists an adversary $\mathcal{A}$ who can obtain a non-negligible difference in advantage between Game RHIBE^{WD} and Game RHIBE$_C$. We may construct a PPT algorithm B to break Assumption 1 with non-negligible advantage.

We assume that the algorithm $\mathcal{B}$ is given $g \in G_{p_1}$ and T. $\mathcal{B}$ then chooses a, b, c, d, m, n, α, β from Z_N and gives the public parameters $Parms = (N, G_1, G_2, \hat{e}, g, u = g^a, h = g^b, v = g^c, w = g^d, z = g^m, f = g^n, \hat{e}(g,g)^\alpha, \hat{e}(g,g)^\beta)$ to the adversary $\mathcal{A}$. The algorithm $\mathcal{B}$ can use the system secret keys α and the time secret key β to respond to $\mathcal{A}$'s initial secret key and time update key requests by calling *Initial key extract* and *Time key update* algorithms to return $\mathcal{A}$ the resulting keys, respectively.

At some point, the adversary $\mathcal{A}$ provides a plaintext pair (M_0^*, M_1^*), a target identity vector id^* and a period t^* to the algorithm $\mathcal{B}$. $\mathcal{B}$ then chooses random values $m_1, \dots, m_j \in Z_N$ and $\mathfrak{b} \in \{0, 1\}$, and computes the ciphertext C^* as follows:

$$\begin{aligned} C^* &= \left(C_0^*, C_1^*, C_{i,2}^*, C_{i,3}^*, C_{i,4}^*, C_{i,5}^*\right), \quad \forall i \in \{1, \dots, j\} \\ &= \left(\hat{e}(g,T)^{(\alpha+\beta)} \cdot M_b^*, T, T^d v^{m_i}, g^{m_i}, (u^{id_i^*} h)^{m_i}, (z^{t^* \cdot id_i^*} f)^{m_i}\right), \quad \forall i \in \{1, \dots, j\}. \end{aligned}$$

Since $C_1^* = T$, it implicitly sets g^s equal to the G_{p_1} part of T. Meanwhile, we have the simulation of the ciphertext C^* as follows:

$$\begin{aligned} C^* &= \left(C_0^*, C_1^*, C_{i,2}^*, C_{i,3}^*, C_{i,4}^*, C_{i,5}^*\right) \\ &= \left(\hat{e}\left(g, g^s\right)^{(\alpha+\beta)} \cdot M_b^*, g^s, g^{s \cdot d} v^{m_i}, g^{m_i}, \left(u^{id_i^*} h\right)^{m_i}, \left(z^{t^* \cdot id_i^*} f\right)^{m_i}\right) \end{aligned}$$

$$= \left(\hat{e}(g,g)^{s\cdot(\alpha+\beta)} \cdot M_b^*, g^s, w^s v^{m_i}, g^{m_i}, \left(u^{id_i^*} h\right)^{m_i}, \left(z^{t^*\cdot id_i^*} f\right)^{m_i}\right),$$
$$\forall i \in \{1, \dots, j\}.$$

If $T \in G_{p_1}$, C^* is a well-distributed normal ciphertext, and B has properly simulated Game RHIBE^{WD}. If $T \in G_{p_1 p_2}$, C^* is a well-distributed semi-functional ciphertext (since the value of d modulo p_2 is uncorrelated from its value modulo p_1 by the Chinese Remainder Theorem). Hence, $\mathcal{B}$ has properly simulated Game RHIBE$_C$ in this case. By the above discussions, we say that $\mathcal{B}$ perfectly simulates the ciphertext C^*. Thus, $\mathcal{B}$ can use the output of $\mathcal{A}$ to achieve a non-negligible advantage against Assumption 1. It is obvious that the successful probability of $\mathcal{B}$ in breaking Assumption 1 is presented as $Adv1_{\mathcal{G},\mathcal{B}} = |\Pr[\mathcal{B}(G, g, T \in G_{p_1}) = 1] - \Pr[B(G, g, T \in G_{p_1 p_2}) = 1]|$. □

Lemma 3. *Under Assumptions* 3 *and* 4, *our dual system encryption RHIBE scheme has semi-functional initial key and semi-functional time key invariance.*

Proof. Under Assumptions 3 and 4, Lewko and Waters (2011) had proven that their dual system encryption HIBE has semi-functional key invariance. As presented in Section 2.3, the semi-functional key is

$$\begin{aligned} \tilde{D}_{id} &= (\tilde{d}_{i,0}, \tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,3}), \quad \text{for } i \in \{1, \dots, j\} \\ &= \begin{cases} (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3}), & \text{if } i \in \{1, \dots, j-1\}, \\ (d_{i,0} \cdot (g_2 g_3)^{\eta\cdot\hat{y}_j}, d_{i,1} \cdot (g_2 g_3)^{\hat{y}_j}, d_{i,2} \cdot (g_2 g_3)^{\zeta\cdot\hat{y}_j}, d_{i,3}), & \text{if } i = j. \end{cases} \end{aligned}$$

In our dual system encryption RHIBE, the semi-functional initial secret key and the semi-functional time update key are

$$\begin{aligned} \tilde{D}_{id} &= (\tilde{d}_{i,0}, \tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,3}), \quad \text{for } i \in \{1, \dots, j\} \\ &= \begin{cases} (d_{i,0}, d_{i,1}, d_{i,2}, d_{i,3}), & \text{if } i \in \{1, \dots, j-1\}. \\ (d_{i,0} \cdot (g_2 g_3)^{\eta\cdot\hat{y}_j}, d_{i,1} \cdot (g_2 g_3)^{\hat{y}_j}, d_{i,2} \cdot (g_2 g_3)^{\zeta\cdot\hat{y}_j}, d_{i,3}), & \text{if } i = j. \end{cases} \end{aligned}$$

and

$$\begin{aligned} \tilde{T}_{id,t} &= (\tilde{t}_{i,0}, \tilde{t}_{i,1}, \tilde{t}_{i,2}, \tilde{t}_{i,3}), \quad \text{for } i \in \{1, \dots, j\} \\ &= \begin{cases} (t_{i,0}, t_{i,1}, t_{i,2}, t_{i,3}), & \text{if } i \in \{1, \dots, j-1\}, \\ (t_{i,0} \cdot (g_2 g_3)^{\omega\cdot\hat{x}_j}, t_{i,1} \cdot (g_2 g_3)^{\hat{x}_j}, t_{i,2} \cdot (g_2 g_3)^{\sigma\cdot\hat{x}_j}, t_{i,3}), & \text{if } i = j. \end{cases} \end{aligned}$$

Obviously, the generations of our semi-functional initial key and the semi-functional time key are the same as the generation of Lewko and Waters's semi-functional key. Hence, we can say that our dual system encryption RHIBE scheme has semi-functional initial key and semi-functional time key invariance. Here, we omit the complete proof. □

Lemma 4. *Under Assumption* 2, *our dual system encryption RHIBE scheme has semi-functional security.*

Proof. Assume that there exists an adversary $\mathcal{A}$ who can obtain a non-negligible advantage in Game RHIBE$_{SF}$. We may construct a PPT algorithm $\mathcal{B}$ to break Assumption 2 with non-negligible advantage. We assume that $\mathcal{B}$ is given $g, g_2, g_3, g^{\alpha}X_2, g^{\beta}X_2, g^{s}Y_2$ and T. $\mathcal{B}$ chooses a, b, c, d, m, n from Z_N and gives the public parameters $Parms = (N, G_1, G_2, \hat{e}, g, u = g^a, h = g^b, v = g^c, w = g^d, z = g^m, f = g^n, \hat{e}(g, g^{\alpha}X_2), \hat{e}(g, g^{\beta}X_2))$ to $\mathcal{A}$. Assume that $\mathcal{B}$ does not know the system secret key α and the time secret key β.

Upon receiving an initial secret key query with identity $id = (id_1, \dots, id_j)$, $\mathcal{B}$ will generate a semi-functional initial secret key as follows. $\mathcal{B}$ chooses random values $r_1, \dots, r_j, y_1, \dots, y_{j-1}, y'_j, \lambda_1, \dots, \lambda_{j-1}, \lambda'_j \in Z_N$ subject to the constraint that $\lambda_1 + \lambda_2 + \cdots + \lambda_{j-1} + \lambda'_j = 0$. Since $\lambda_1 + \lambda_2 + \cdots + \lambda_j = \alpha$, it will implicitly set $\lambda_j = \alpha + \lambda j'$ mod p_1 and $y_j = \alpha + y'_j$ mod p_1. $\mathcal{B}$ also chooses a random value $\hat{k} \in Z_N$. The semi-functional initial secret key is presented as:

$$\begin{aligned} \tilde{D}_{id} &= (\tilde{d}_{i,0}, \tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,3}), \quad \text{for } i \in \{1, \dots, j\} \\ &= \begin{cases} (g^{\lambda_i} w^{y_i}, g^{y_i}, v^{y_i}(u^{id_i}h)^{r_i}, g^{r_i}), & \text{if } i \in \{1, \dots, j-1\} \\ ((g^{\alpha}X_2)^{d+1} \cdot g^{\lambda'_i} w^{y'_i} \cdot (g_2g_3)^{\hat{k}(d+1)}, (g^{\alpha}X_2) \cdot g^{y'_i} \cdot (g_2g_3)^{\hat{k}} \\ \quad \cdot (g^{\alpha}X_2)^{c} \cdot v^{y'_i} \cdot (u^{id_i}h)^{r_i} \cdot (g_2g_3)^{\hat{k}c}, g^{r_i}), & \text{if } i = j. \end{cases} \end{aligned}$$

Observe that this is a well-distributed semi-functional key with the following equation

$$\begin{cases} \eta = d + 1 \bmod p_2, \\ \zeta = c \bmod p_3, \end{cases} \quad \text{and} \quad \begin{cases} \hat{y}_j = \hat{k} + \log_{g_2} X_2 \bmod p_2, \\ \hat{y}_j = \hat{k} \bmod p_3 \end{cases}$$

we have that the resulting key $\tilde{D}_{id}$ is a well-distributed semi-functional initial secret key. Note that $\hat{y}_j$ is freshly random modulo p_2 and p_3 for each key, while ζ and η are the same for all keys, as specified by *Initial key extract SF* algorithm.

Upon receiving a time update key query with identity $id = (id_1, \dots, id_j)$ in a period t, $\mathcal{B}$ will generate a semi-functional time update key as follows. $\mathcal{B}$ chooses random values $k_1, \dots, k_j, x_1, \dots, x_{j-1}, x'_j, \mu_1, \dots, \mu_{j-1}, \mu'_j \in Z_N$ subject to the constraint that $\mu_1 + \mu_2 + \cdots + \mu_{j-1} + \mu'_j = 0$. Since $\mu_1 + \mu_2 + \cdots + \mu_j = \beta$, it will implicitly set $\mu_j = \beta + \mu'_j$ mod p_1 and $x_j = \beta + x'_j$ mod p_1. The algorithm $\mathcal{B}$ also chooses a random value $\hat{l} \in Z_N$. The semi-functional time update key is presented as:

$$\begin{aligned} \tilde{T}_{id,t} &= (\tilde{t}_{i,0}, \tilde{t}_{i,1}, \tilde{t}_{i,2}, \tilde{t}_{i,3}), \quad \text{for } i \in \{1, \dots, j\} \\ &= \begin{cases} (g^{\mu_i} w^{x_i}, g^{x_i}, v^{x_i}(z^{t \cdot id_i} f)^{k_i}, g^{k_i}), & \text{if } i \in \{1, \dots, j-1\} \\ ((g^{\beta}X_2)^{d+1} \cdot g^{\mu'_i} w^{x'_i} \cdot (g_2g_3)^{\hat{l}(d+1)}, (g^{\beta}X_2) \cdot g^{x'_i} \cdot (g_2g_3)^{\hat{l}} \\ \quad \cdot (g^{\beta}X_2)^{c} \cdot v^{x'_i} \cdot (z^{t \cdot id_i} f)^{k_i} \cdot (g_2g_3)^{\hat{l}c}, g^{k_i}), & \text{if } i = j. \end{cases} \end{aligned}$$

By observing the following equation

$$\begin{cases} \omega = d + 1 \bmod p_2, \\ \sigma = c \bmod p_3, \end{cases} \quad \text{and} \quad \begin{cases} \hat{x}_j = \hat{l} + \log_{g_2} X_2 \bmod p_2, \\ \hat{x}_j = \hat{l} \bmod p_3 \end{cases}$$

we have that $\tilde{T}_{id,t}$ is also a well-distributed semi-functional time update key. Notice that $\hat{x}_j$ is freshly random modulo p_2 and p_3 for each key, while σ and ω are the same for all keys, as specified by Time key update SF algorithm.

At some point, $\mathcal{A}$ provides a plaintext pair (M_0^*, M_1^*), a target identity vector id^* and a period t^*, and requests the challenge ciphertext C^*. The algorithm $\mathcal{B}$ chooses random values $m_1, \dots, m_j, \delta' \in Z_N$ and $\mathfrak{b} \in \{0, 1\}$, and computes the ciphertext as follows:

$$\begin{aligned} C^* &= \left(C_0^*, C_1^*, C_{i,2}^*, C_{i,3}^*, C_{i,4}^*, C_{i,5}^*\right), \quad \forall i \in \{1, \dots, j\} \\ &= (M_\gamma T, g^s Y_2, (g^s Y_2)^d \cdot v^{m_i} \cdot g_2^{\delta'}, g^{m_i}, (u^{id_i^*} h)^{m_i}, (z^{t^* \cdot id_i^*} f)^{m_i}), \quad \forall i \in \{1, \dots, j\}. \end{aligned}$$

As mentioned in Assumption 2, if $T = \hat{e}(g, g)^{(\alpha+\beta)\cdot s}$, this is a well-distributed semi-functional encryption of Mb with the following equation

$$\begin{cases} \tau = \log_{g_2} Y_2 \bmod p_2, \\ \delta = d \cdot \log_{g_2} Y_2 + \delta \bmod p_2. \end{cases}$$

Notice that δ' randomizes this so that there is no correlation with d mod p_2. Hence, this is uncorrelated from the exponents modulo p_2 of the semi-functional initial secret and time update keys. Thus B has properly simulated Game RHIBE$_{SF}$.

If T is a random element of G_2, then this is a semi-functional encryption of a random message. The advantage of $\mathcal{A}$ must be zero, because the ciphertext contains no information about b. Since we have assumed the advantage of $\mathcal{A}$ is non-negligible in Game RHIBE$_{SF}$, B can use the output of $\mathcal{A}$ to achieve a non-negligible advantage against Assumption 2. It is obvious that the successful probability of $\mathcal{B}$ in breaking Assumption 2 is presented as $Adv2_{\mathcal{G},\mathcal{B}} = |\Pr[\mathcal{B}(G, g, g_2, g_3, g^\alpha X_2, g^\beta X_2, g^s Y_2, T = \hat{e}(g, g)^{(\alpha+\beta)\cdot s}) = 1] - \Pr[B(G, g, g_2, g_3, g^\alpha X_2, g^\beta X_2, g^s Y_2, T \in G_2) = 1]|$. □

Theorem 2. *Under Assumptions* 1–4*, our proposed RHIBE scheme is fully secure.*

Proof. By Lemmas 1–4, we have proven that our proposed RHIBE scheme has initial key and time key delegation invariance, semi-functional ciphertext invariance, semi-functional initial key and time key invariance, and semi-functional security. By Theorem 1, we can say that our proposed RHIBE scheme is fully secure. □

6. Transformation from HIBE to RHIBE

Actually, in most of the existing HIBE systems, users hold only one private key or one set of private keys. As presented in Section 4, we have proposed a concrete RHIBE scheme with public revocation mechanism, which is extended from Lewko and Waters's HIBE scheme. In our proposed RHIBE scheme, it is easy to see that one private key is divided into a fixed initial secret key and a changed time update key along with periods. The fixed initial secret key is identical to the private key in Lewko and Waters's HIBE scheme. The changed time update key is generated by the DRA who can assist the PKG to revoke illegal or expired users from the system. The point is that any user in our proposed RHIBE scheme needs to get both the fixed initial secret key and the changed time update key to decrypt his/her ciphertext. If a receiver obtains only either the fixed initial secret key or the changed time update key, the receiver cannot get the complete private key.

In the same way, we show that the transformation technique is also suitable to construct a RHIBE scheme from a HIBE scheme in the random oracle model. In the following, we will present another RHIBE scheme from a famous HIBE scheme proposed by Gentry and Silverberg (2002).

Since we do not use bilinear groups of the composite order as in Section 2.1, we briefly redefine some parameters as follows. Let G_1 and G_2 be additive and multiplicative cyclic groups of large prime order q, respectively. An admissible bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ must satisfy the following properties:

(1) Bilinear: For all $P, Q \in G_1$, and $a, b \in Z_q^*$, we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
(2) Non-degenerate: There exist $P, Q \in G_1$ such that $\hat{e}(P, Q) \neq 1$.
(3) Computable: For $P, Q \in G_1$, there exists an efficient algorithm to compute $\hat{e}(P, Q)$.

We assumed that a user *ID*'s position in the following RHIBE scheme is defined by *ID*-tuple $(ID_1, \dots, ID_j)$. The user *ID*'s ancestors in the following RHIBE scheme are the PKG and users in lower level whose *ID*-tuple is $\{(ID_1, ID_2, \dots, ID_i)\colon 1 \leqslant i < j\}$. The concrete RHIBE scheme is presented as follows:

- ***Setup*:** Given a security parameter k and the total number $\mathcal{T}$ of periods, a trusted private key generator (PKG) generates two groups G_1, G_2 of prime order $q > 2^k$, an admissible bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ and a generator P_0 of G_1. The PKG performs the following tasks:

  (1) Randomly choose a system secret key $s_0 \in Z_q^*$ and set $Q_0 = s_0 \cdot P_0$.
  (2) Pick three hash functions H_1: $\{0,1\}^* \rightarrow G_1$, H_2: $\{0,1\}^* \rightarrow G_1$ and H_3: $G_2 \rightarrow \{0,1\}^n$ for some n.

  The message space is $\mathcal{M} = \{0,1\}^n$. The ciphertext space is $C = G_1^j \times \{0,1\}^n$, where j is the level of the recipient. Then the public parameters are presented as *Parms* = $\{G_1, G_2, \hat{e}, P_0, Q_0, H_1, H_2, H_3\}$.
- ***Lower-level setup*:** An entity $ID_j \in \text{Level}_j$ picks a random $s_j \in Z_q^*$ as its secret key.
- ***Initial key extract*:** Given an entity ID_j's *ID*-tuple $(ID_1, \dots, ID_j)$ at Level_j, where $(ID - 1, \dots, ID_i)$, for $1 \leqslant i < j$, is the *ID*-tuple of an entity ID_j's ancestor at Level_i. Then ID_j's parent performs the following tasks:

  (1) Compute $P_j = H_1(ID_1, \dots, ID_j) \in G_1$.

(2) Set D_0 to be the identity element of G_1 and compute ID_j's initial secret key D_j as follows:

$$D_j = D_{j-1} + s_{j-1} \cdot P_j = \sum_{i=1}^{j} s_{i-1} \cdot P_i.$$

(3) Give the values of $Q_i = s_i \cdot P_0$ for $1 \leqslant i \leqslant j-1$ to an entity ID_j.

- ***Time key update***: In a period t, given a non-revoked entity ID_j's ID-tuple $(ID_1, \ldots, ID_j)$ at Level$_j$, where $(ID_1, \ldots, ID_i)$, for $1 \leqslant i < j$, is the ID-tuple of an entity ID_j's ancestor at Level$_i$. Then ID_j's parent performs the following tasks:

  (1) Compute $T_j = H_2(ID_1, \ldots, ID_j, t) \in G_1$.
  (2) Set R_0 to be the identity element of G_1 and set ID_j's time secret key R_j as follows:

$$R_j = R_{j-1} + s_{j-1} \cdot T_j = \sum_{i=1}^{j} s_{i-1} \cdot T_i.$$

- ***Encryption***: In a period t, given a message m and the ID-tuple $(ID_1, \ldots, ID_j)$, a sender performs the following tasks:

  (1) Compute $W_j = P_j + T_j = H_1(ID_1, \ldots, ID_j) + H_2(ID_1, \ldots, ID_j, t) \in G_1$, for $1 \leqslant j \leqslant i$.
  (2) Choose a random $r \in Z_q^*$.
  (3) Set the ciphertext for the message m to be:

$$\begin{aligned} C &= [U_0, U_2, \ldots, U_j, V] \\ &= \big[r \cdot P_0, r \cdot W_2, \ldots, r \cdot W_j, m \oplus H_3\big(\hat{e}(Q_0, W+1)^r\big)\big]. \end{aligned}$$

- ***Decryption***: Given a ciphertext $C = [U_0, U_2, \ldots, U_j, V]$, the receiver ID_j can use his/her initial secret key D_j and time secret key R_j to compute $V \oplus H_3(\frac{\hat{e}(U_0, D_j + R_j)}{\prod_{i=2}^{j} \hat{e}(Q_{i-1}, U_i)}) = m$.

We present the correctness of the decryption equation as follows:

$$\begin{aligned} & V \oplus H_3\left(\frac{\hat{e}(U_0, D_j + R_j)}{\prod_{i=2}^{j} \hat{e}(Q_{i-1}, U_i)}\right) \\ &= V \oplus H_3\left(\frac{\hat{e}(r \cdot P_0, \sum_{i=1}^{j} s_{i-1} \cdot P_i + \sum_{i=1}^{j} s_{i-1} \cdot T_i)}{\hat{e}(Q_1, U_2) \cdot \hat{e}(Q_2, U_3) \cdots \hat{e}(Q_{j-1}, U_j)}\right) \\ &= V \oplus H_3\left(\frac{\hat{e}(r \cdot P_0, s_0 \cdot P_1 + s_0 \cdot T_1) \cdot \hat{e}(r \cdot P_0, s_1 \cdot P_2 + s_1 \cdot T_2) \cdots \hat{e}(r \cdot P_0, s_{j-1} \cdot P_j + s_{j-1} \cdot T_j)}{\hat{e}(s_1 \cdot P_0, r \cdot (P_2 + T_2)) \cdot \hat{e}(s_2 \cdot P_0, r \cdot (P_3 + T_3)) \cdots \hat{e}(s_{j-1} \cdot P_0, r \cdot (P_j + T_j))}\right) \\ &= V \oplus H_3\big(\hat{e}(r \cdot P_0, s_0 \cdot P_1 + s_0 \cdot T_1)\big) \end{aligned}$$

$$
\begin{aligned}
&= V \oplus H_3\big(\hat{e}\big(s_0 \cdot P_0, r \cdot (P_1 + T_1)\big)\big)\\
&= V \oplus H_3\big(\hat{e}(Q_0, W_1)^r\big)\\
&= m \oplus H_3\big(\hat{e}(Q_0, W_1)^r\big) \oplus H_3\big(\hat{e}(Q_0, W_1)^r\big)\\
&= m.
\end{aligned}
$$

By the proposed RHIBE scheme above, it is obvious that the transformation technique is also suitable to construct a RHIBE scheme from a HIBE scheme in the random oracle model. In the meantime, readers can easily find that there exists no DRA in the proposed RHIBE above. The work of the DRA is performed by the PKG. Indeed, the system can remain the efficiency for encryption and decryption procedures as compared to Gentry and Silverberg's HIBE scheme (2002). On the other hand, if we would like to increase the flexibility of the proposed RHIBE scheme for revoking illegal or expired users, a DRA can be involved into the proposed RHIBE scheme and it can assist the PKG to revoke any users in the whole system.

For the security of the proposed RHIBE scheme above, we can employ the work of Gentry and Silverberg (2002) to provide security proof in the random oracle model. In the security game of RHIBE scheme as mentioned in Section 3.2, a restriction in *Phase* 3 is that either id^* or (id^*, t^*) did not appear in *Phase* 2. It means that the adversary is allowed to query either the related information (including initial key create, delegate and reveal) of initial secret keys on id^* or the related information (including time key create, delegate and reveal) of time update keys on (id^*, t^*). Since the user's private key is divided into the initial secret key and the time update key, the adversary is unable to compute the user's private key. For simplicity of security proof, we consider two types of adversaries in the security game. One is an outside adversary who can be allowed to obtain the time update keys. The other is a revoked user who is unable to obtain its time update key in the present period, but the revoked user still owns the initial secret key. That is, the outside adversary is allowed to issue all queries in the security game except the related queries of initial secret key on id^*, while the revoked user can issue all queries except the related queries of the time update key on (id^*, t^*). For outside adversaries, we can employ the similar work of Gentry and Silverberg (2002) to prove that the proposed RHIBE in the random oracle model is secure. Certainly, we can also extend the work of Gentry and Silverberg (2002) to prove that the proposed RHIBE is secure for the attacks of revoked users. Here, we omit the security proofs.

7. Comparisons Between HIBE and RHIBE

Extended from Lewko and Waters's HIBE scheme (2011) (called LW-HIBE), we constructed the first RHIBE scheme (called LW-RHIBE) in the standard model (WROM). We also presented another RHIBE scheme (called GS-RHIBE) extended from Gentry and Silverberg's HIBE scheme (2002) (called GS-HIBE) in the random oracle model (ROM). Here, we make a comparison between the mentioned schemes in terms of security model,

Table 1
Comparisons between two famous HIBE schemes and our RHIBE schemes.

	GS-HIBE scheme (2002)	Our GS-RHIBE scheme	LW-HIBE scheme (2011)	Our LW-RHIBE scheme
Model	ROM	ROM	WROM	WROM
Encryption	T_p	T_p	0	0
Decryption	T_p	T_p	$4T_p$	$5T_p$
Revocation channel	Secure	Public	Secure	Secure
Periodical Encryption/Decryption for revocation	Required	Not required	Required	Not required
DRA	Not involved	Not involved	Not involved	Involved
Bit length of ciphertext	$j\|G_1\|+n$	$j\|G_1\|+n$	$(3j+1)\|G_1\|+\|G_2\|$	$(4j+1)\|G_1\|+\|G_2\|$

computational cost, revocable functionality and communication cost. Note that T_p denotes the computational cost of a pairing operation and $|C|$ denotes the bit length of C.

Table 1 lists comparisons between our proposed RHIBE schemes and two original HIBE schemes (Lewko and Waters, 2011; Gentry and Silverberg, 2002) in terms of security model, encryption cost, decryption cost for each level, revocable functionality, and communication cost. For the decryption of the LW-HIBE scheme and our proposed LW-RHIBE scheme, they require four and five pairing operations for each level, respectively. Although our LW-RHIBE scheme increases one pairing operation as compared to the LW-HIBE scheme for the decryption procedure, but the point is that our LW-RHIBE scheme provides a flexible revocation mechanism with a DRA using a public channel. Certainly, we can also involve a DRA into the GS-RHIBE scheme to assist the PKG for sharing the responsibility of revoking illegal or expired users. Both the LW-HIBE and the GS-HIBE schemes may employ the revocation mechanism suggested by Boneh and Franklin (2001), but they require secure channels to transmit the users' new private keys for each period. Thus, extra computational costs of encryption and decryption procedures for each period are required. Our proposed RHIBE schemes provide a public revocation mechanism to remove the requirement of secure channels and extra computational costs for periodic key updating required in the original HIBE schemes while remaining efficiency.

In the following, let us evaluate the communication cost. In the proposed GS-RHIBE scheme, the ciphertext is $C = [U_0, U_2, \dots, U_j, V]$, where j is the position of level. The bit length of the ciphertext C is $|U_0| + |U_2| + \cdots + |U_j| + |V|$, where V is bounded to the hash function $H_3()$. Since $U_0, U_2, \dots, U_j \in G_1$ and H_3 is mapped to $\{0, 1\}^n$, the bit length of the ciphertext is $j|G_1| + n$. Considering the bit length of ciphertext in the proposed LW-RHIBE scheme, the sender transmits a ciphertext $C = (C_0, C_1, C_{j,2}, C_{j,3}, C_{j,4}, C_{j,5})$ to the receiver, where j is the position of level. The bit length of the ciphertext C is $|C_0|+|C_1|+|C_{j,2}|+|C_{j,3}|+|C_{j,4}|+|C_{j,5}|$. Since $C_0 \in G_2$ and $g, u, h, v, w, z, f \in G_{p_1}$, where G_{p_1} is the subgroup of G_1, the bit length of the ciphertext in the proposed LW-RHIBE scheme is $(4j + 1)|G_1| + |G_2|$, where j is the position of level.

8. Conclusions

In this paper, we proposed a public revocation mechanism that is an exciting alternative to the existing revocation methods. We defined the framework of RHIBE scheme with public revocation mechanism. Meanwhile, its security notions were completely defined to formalize the possible threats. Based on Lewko and Waters's HIBE scheme, we employed the public revocation mechanism to propose a concrete RHIBE scheme. We have proven that the proposed RHIBE scheme is fully secure in the standard model while removing the requirement of secure channels for private key updating, and remaining the merits of Lewko and Waters's HIBE scheme. We also employed the public revocation mechanism to propose another concrete RHIBE scheme in the random oracle model. For the revocation flexibility, we also consider the situation whether a DRA should be put in a RHIBE scheme. We demonstrated that our RHIBE schemes not only provide a public revocation mechanism but also remain efficiency in encryption and decryption procedures as compared with the original RHIBE schemes. Our transformation technique is suitable to construct RHIBE schemes from most of the existing HIBE schemes. In the future, a general transformation technique from any existing HIBE scheme to RHIBE scheme is an interesting research issue.

Acknowledgements. The authors would like to thank the anonymous referees for their valuable comments and constructive suggestions. This research was partially supported by National Science Council, Taiwan, R.O.C., under contract No. NSC100-2221-E-018-027.

References

Bellare, M., Boldyreva, A., Palacio, A. (2004). An uninstantiable random oracle model scheme for a hybrid encryption problem. In: *Proceedings of Eurocrypt'04*, LNCS, Vol. 3027, pp. 171–188.

Bellare, M., Waters, B., Yilek, S. (2011). Identity-based encryption secure against selective opening attack. In: *Proceedings of TCC'11*, LNCS, Vol. 6597, pp. 235–252.

Boldyreva, A., Goyal, V., Kumar, V. (2008). Identity-based encryption with efficient revocation. In: *Proceedings of ACM CCS'08*, pp. 417–426.

Boneh, D., Boyen, X. (2004a). Efficient selective-ID secure identity-based encryption without random oracles. In: *Proceedings of Eurocrypt'04*, LNCS, Vol. 3027, pp. 223–238.

Boneh, D., Boyen, X. (2004b). Secure identity based encryption without random oracles. In: *Proceedings of Crypto'04*, LNCS, Vol. 3152, pp. 443–459.

Boneh, D., Franklin, M. (2001). Identity-based encryption from the Weil pairing. In: *Proceedings of Crypto'01*, LNCS, Vol. 2139, pp. 213–229.

Boneh, D., Hamburg, M. (2008). Generalized identity based and broadcast encryption schemes. In: *Proceedings of Asiacrypt'08*, LNCS, Vol. 5350, pp. 455–470.

Boyen, X., Waters, B. (2006). Anonymous hierarchical identity-based encryption (without random oracles). In: *Proceedings of Crypto'06*, LNCS, Vol. 4117, pp. 290–307.

Boneh, D., Boyen, X., Goh, E. (2005a). Hierarchical identity based encryption with constant size ciphertext. In: *Proceedings of Eurocrypt'05*, LNCS, Vol. 3494, pp. 440–456.

Boneh, D., Goh, E., Nissim, K. (2005b). Evaluating 2-DNF formulas on ciphertexts. In: *Proceedings of TCC'05*, LNCS, Vol. 3378, pp. 325–341.

Canetti, R., Halevi, S., Katz, J. (2003). A forward-secure public-key encryption scheme. In: *Proceedings of Eurocrypt'03*, LNCS, Vol. 2656, pp. 255–271.

Cha, J.C., Cheon, J.H. (2003). An identity-based signature from gap Diffie-Hellman groups. In: *Proceedings of PKC'03*, LNCS, Vol. 2567, pp. 18–30.

Chen, L., Cheng, Z., Smart, N.P. (2007). Identity-based key agreement protocols from pairings. *International Journal of Information Security*, 6(4), 213–241.

Chen, J., Chen, K., Wang, Y., Li, X., Long, Y., Wan, Z. (2012). Identity-based key-insulated signcryption. *Informatica*, 23(1), 27–45.

Gentry, C. (2006). Practical identity-based encryption without random oracles. In: *Proceedings of Eurocrypt'06*, LNCS, Vol. 4004, pp. 445–464.

Gentry, C., Halevi, S. (2009). Hierarchical identity based encryption with polynomially many levels. In: *Proceedings of TCC'09*, LNCS, Vol. 5444, pp. 437–456.

Gentry, C., Silverberg, A. (2002). Hierarchical ID-based cryptography. In: *Proceedings of Asiacrypt'02*, LNCS, Vol. 2501, pp. 548–566.

Horwitz, J., Lynn, B. (2002). Toward hierarchical identity-based encryption. In: *Proceedings of Eurocrypt'02*, LNCS, Vol. 2332, pp. 466–481.

Lewko, A., Waters, B. (2010). New techniques for dual system encryption and fully secure hibe with short ciphertexts. In: *Proceedings of TCC'10*, LNCS, Vol. 5978, pp. 455–479.

Lewko, A., Waters, B. (2011). Unbounded HIBE and attribute-based encryption. In: *Proceedings of Eurocrypt'11*, LNCS, Vol. 6632, pp. 547–567.

Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B. (2010). Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: *Proceedings of Eurocrypt'10*, LNCS, Vol. 6110, pp. 62–91.

Libert, B., Vergnaud, D. (2009). Adaptive-ID secure revocable identity-based encryption. In: *Proceedings of CT-RSA'09*, LNCS, Vol. 5473, pp. 1–15.

Okamoto, T., Takashima, K. (2009). Hierarchical predicate encryption for inner-products. In: *Proceedings of Asiacrypt'09*, LNCS, Vol. 5912, pp. 214–231.

Paterson, K.G. (2002). Identity-based signatures from pairings on elliptic curves. *Electronics Letters*, 38(18), 1025–1026.

Paterson, K.G., Schuldt, J.C.N. (2006). Efficient identity-based signatures secure in the standard model. In: *Proceedings of ACISP'06*, LNCS, Vol. 4058, pp. 207–222.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In: *Proceedings of Crypto'84*, LNCS, Vol. 196, pp. 47–53.

Shi, E., Waters, B. (2008). Delegating capabilities in predicate encryption systems. In: *Proceedings of ICALP'08, Part II*, LNCS, Vol. 5126, pp. 560–578.

Tsai, T.T., Tseng, Y.M., Wu, T.Y. (2012). A fully secure revocable ID-based encryption in the standard model. *Informatica*, 23(3), 481–499.

Tseng, Y.M., Tsai, T.T. (2012). Efficient revocable ID-based encryption with a public channel. *The Computer Journal*, 55(4), 475–486.

Tseng, Y.M., Wu, T.Y., Wu, J.D. (2008). A pairing-based user authentication scheme for wireless clients with smart cards. *Informatica*, 19(2), 285–302.

Waters, B. (2005). Efficient identity-based encryption without random oracles. In: *Proceedings of Eurocrypt'05*, LNCS, Vol. 3494, pp. 114–127.

Waters, B. (2009). Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: *Proceedings of Crypto'09*, LNCS, Vol. 5677, pp. 619–636.

Wu, T.Y., Tseng, Y.M. (2012). Towards ID-based authenticated group key exchange protocol with identifying malicious participants. *Informatica*, 23(2), 315–334.

Wu, T.Y., Tseng, Y.M., Yu, C.W. (2011). A secure ID-based authenticated group key exchange protocol resistant to insider attacks. *Journal of Information Science and Engineering*, 27(3), 915–932.

T.-T. Tsai received the BS degree in Department of Applied Mathematics, Chinese Culture University, Taiwan, in 2006. He received the MS degree at Department of Applied Mathematics, National Hsinchu University of Education, Taiwan, in 2009. He is currently a PhD candidate in Department of Mathematics, National Changhua University of Education, Taiwan. His research interests include applied cryptography and pairing-based cryptography.

Y.-M. Tseng is currently a Professor in Department of Mathematics, National Changhua University of Education, Taiwan. He is a member of IEEE Computer Society, IEEE Communications Society and the Chinese Cryptology and Information Security Association (CCISA). In 2006, his paper obtained the *Wilkes Award* from *The British Computer Society*. He has published over a hundred scientific journal and conference papers on various research areas of cryptography, security and computer network. His research interests include cryptography, network security, computer network and mobile communications. He serves as an editor of several international Journals: *Computer Standards & Interfaces*; *International Journal of Computer Mathematics*; *International Journal of Information and Network Security*; *International Journal of Security and Its Applications*; *Wireless Engineering and Technology*.

T.-Y. Wu received the BS and the MS degrees in Department of Applied Mathematics, Tatung University, Taiwan, in 2003 and 2005, respectively. He received the PhD degree in Department of Mathematics, National Changhua University of Education, Taiwan, in 2010. He is currently an assistant professor in School of Computer Science and Technology, Shenzhen Graduate School, Harbin Institute of Technology, China. His research interests include applied cryptography, pairing-based cryptography and information security, and computer network.

Atšaukiamos hierarchinės identifikatoriumi grįstos šifravimo sistemos (RHIBE) sukūrimas

Tung-Tso TSAI, Yuh-Min TSENG, Tsu-Yang WU

Bet kuri hierarchinė identifikatoriumi grįsta viešojo rakto šifravimo sistema (HIBE) privalo turėti galimybę, pašalinti neteisėtus vartotojus ir vartotojus, kurių raktų galiojimo laikas baigėsi. Bonch ir Franklin pasiūlė viešojo rakto identifikatoriumi grįstų sistemų raktų atšaukimo metodą, kuriame patikimas viešųjų raktų generatorius (PKG) generuoja naujus privačiuosius raktus visiems teisėtiems vartotojams ir juos persiunčia saugiu ryšio kanalu. Straipsnyje pasiūlyta nauja HIBE schema su raktų atšaukimo mechanizmu (RHIBE). Ši schema nereikalauja saugaus ryšio kanalo raktams persiųsti. Toks raktų atšaukimo mechanizmas yra gera alternatyva egzistuojantiems raktų atšaukimo metodams. Aprašytas HIBE schemos transformavimo į RHIBE schemą algoritmas.