Unbounded Hierarchical Identity-Based Encryption with Efficient Revocation

Geumsook Ryu, Kwangsu Lee, Seunghwan Park, and Dong Hoon Lee(✉)

CIST, Korea University, Seoul, Republic of Korea
{madeby_r,kwangsu.lee,sgusa,donghlee}@korea.ac.kr

Abstract. Hierarchical identity-based encryption (HIBE) is an extension of identity-based encryption (IBE) where an identity of a user is organized as a hierarchical structure and a user can delegate the private key generation to another user. Providing a revocation mechanism for HIBE is highly necessary to keep a system securely. Revocable HIBE (RHIBE) is an HIBE scheme that can revoke a user's private key if his credential is expired or revealed. In this paper, we first propose an unbounded HIBE scheme where the maximum hierarchy depth is not limited and prove its selective security under a q-type assumption. Next, we propose an efficient unbounded RHIBE scheme by combining our unbounded HIBE scheme and a binary tree structure, and then we prove its selective security. By presenting the unbounded RHIBE scheme, we solve the open problem of Seo and Emura in CT-RSA 2015.

Keywords: Identity-based encryption · Hierarchical identity-based encryption · Revocation · Unbounded hierarchy depth · Bilinear maps

1 Introduction

Identity-based encryption (IBE) is a kind of public key encryption (PKE) that uses any bit-string (e.g., e-mail address, phone number, or identity) as a public key of a user. Although the concept of IBE was introduced by Shamir [24], the first realization of IBE was achieved by Boneh and Franklin [4] by using bilinear maps. In IBE, a single key generation center (KGC) should issue private keys and establish secure channels to transmit private keys of users. To reduce the cost of private key generation of the KGC in IBE, the concept of hierarchical IBE (HIBE) was introduced such that the KGC delegates the key generation functionality to a lower level KGC [7,8]. After that, many IBE and HIBE schemes were suggested with additional functionalities [2,3,5,14,25,26].

To maintain a whole system securely, a revocation mechanism is absolutely necessary when a user's contract is expired or the user's private key is revealed. Boldyreva, Goyal and Kumar [1] introduced the concept of revocable IBE (RIBE) and proposed a scalable RIBE scheme by combining a fuzzy IBE scheme of Sahai Waters [20] and a tree based revocation system of Naor et al. [16]. In RIBE, each user initially obtains a private key from a KGC, and then the KGC periodically

H. Kim and D. Choi (Eds.): WISA 2015, LNCS 9503, pp. 122–133, 2016.
DOI: 10.1007/978-3-319-31875-2_11

publishes an update key for non-revoked users. If a user is not revoked in the update key, then he can derive a decryption key from his private key and the update key. After the work of Boldyreva et al., many different RIBE scheme were proposed [11,15,17,22].

It is a natural research direction to devise an efficient revocation mechanism for HIBE. By following the design strategy of Boldyreva et al. [1], Seo and Emura proposed efficient revocable HIBE (RHIBE) schemes [21,23]. In RHIBE, a KGC can delegate the key generation functionality and the revocation functionality to a lower level KGC or a user. Seo and Emura [21] first proposed a concrete RHIBE scheme by combining the HIBE scheme of Boneh and Boyen and a binary tree structure. After that, they also proposed new efficient RHIBE schemes by using the history-free update approach to reduce the size of private keys [23]. Although they proposed efficient RHIBE schemes, their RHIBE schemes have the inherent limitation that the size of public parameters linearly grows to the maximum hierarchy depth. Thus, they left it as an interesting problem to devise an unbounded RHIBE scheme [23].

1.1 Our Contributions

In this paper, we give an answer to the above problem of Seo and Emura by presenting an unbounded RHIBE scheme. Before presenting an unbounded RHIBE scheme, we first propose an HIBE scheme with no limitation in maximum hierarchy, denoted by unbounded HIBE. Our unbounded HIBE scheme is derived from the key-policy attribute-based encryption (KP-ABE) scheme of Rouselakis and Waters [18]. We use the observation that an HIBE scheme can be derived from a KP-ABE scheme if the KP-ABE scheme can be modified to support the delegation of private key generation. We prove the selective security of our unbounded HIBE scheme under the q-type assumption introduced by Rouselakis and Waters. Next, we propose an unbounded RHIBE scheme by combining our unbounded HIBE scheme and a tree-based revocation system. Mainly we follow the design strategy of the previous RHIBE scheme of Seo and Emura [23]. To prove the selective security of our RHIBE scheme, we show that our RHIBE scheme is selectively secure if our HIBE scheme is selectively secure.

1.2 Related Work

IBE and Its Extensions. As mentioned before, the concept of IBE was introduced by Shamir [24] where a public key can be the identity string of a user such as an e-mail address. The first IBE scheme that uses bilinear maps was constructed by Boneh and Franklin [4]. Since the pioneering work of Boneh and Franklin, many IBE schemes were proposed in bilinear maps [2,6,25]. The notion of IBE has been extended to several other encryption systems like HIBE [8], attribute-based encryption (ABE), predicate encryption (PE), and functional encryption (FE). The concept of HIBE was introduced by Horwitz and Lynn [8] and it additionally provides a key delegation mechanism by which the private key of a low level user is generated by a upper level user. After the introduction of HIBE, many HIBE schemes with different properties have been suggested in bilinear maps [2,3,5,7,12,13,26]. One inherent limitation of previous HIBE schemes is that the maximum hierarchy depth should be fixed in the setup phase. To remove this restriction, an unbounded HIBE scheme was proposed by Lewko and Waters [14].

Revocation in IBE. Boneh and Franklin [4] proposed the first IBE scheme that supports key revocation, but their scheme is not scalable since each user periodically connects to a KGC to receive a new private key. Boldyreva et al. [1] proposed a scalable RIBE scheme by combining the fuzzy IBE scheme of Sahai and Waters [20] and the tree based revocation system of Naor et al. [16]. Libert and Vergnaud [15] proposed first fully secure RIBE scheme by using a fully secure IBE scheme that is a variant of the Waters IBE [25]. Seo and Emura [22] refined the security model of RIBE by considering decryption key exposure attacks and proposed a fully secure RIBE scheme in their security model. To improve the efficiency of RIBE, Lee et al. [11] proposed a new RIBE scheme based on the subset difference method and Park et al. [17] proposed an RIBE scheme from multilinear maps. An efficient RHIBE scheme was first presented by Seo and Emura [21] and its improvement was also proposed by using the history-free update approach [23]. In RIBE, revoked user on the time T is still accessible to ciphertext that were encrypted before the time T in the cloud storage environment. To solve this problem, Sahai et al. [19] introduced revocable storage ABE (RS-ABE) for cloud storage by using the idea of RIBE. The improved RS-ABE schemes were presented in [9,10].

2 Preliminaries

In this section, we introduce the complexity assumption for our schemes and define the syntax of RHIBE and its security model.

2.1 Bilinear Groups

Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups of prime order p and g be a generator of $\mathbb{G}$. The bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ has the following properties: (1) Bilinearity: for all $u, v \in \mathbb{G}$ and for all $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$. (2) Non-degeneracy: for generator $g \in \mathbb{G}$, $e(g, g) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is an identity element in $\mathbb{G}_T$. Furthermore, we assume the existence of a group generator algorithm $\mathcal{G}$ which takes as input a security parameter λ and outputs a bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e)$ where p is a prime of $\Theta(\lambda)$ bits.

2.2 Complexity Assumption

For the proof of our schemes, we introduce the q-RW2 assumption of Rouselakis and Waters [18] that was used to prove the security of their attribute-based encryption schemes.

Assumption 1 (q-RW2, [18]). *Let $(p, \mathbb{G}, \mathbb{G}_T, e)$ be a description of the bilinear groups of prime order p. Let g be a random generator of $\mathbb{G}$. The q-RW2 assumption is that if the challenge tuple*

$$D = \Big((p, \mathbb{G}, \mathbb{G}_T, e), g, g^x, g^y, g^z, g^{(xz)^2}, \{g^{b_i}, g^{xzb_i}, g^{xz/b_i}, g^{x^2zb_i}, g^{y/b_i^2}, g^{y^2/b_i^2}\}_{\forall\ i\in[q]}, \{g^{xzb_i/b_j}, g^{yb_i/b_j^2}, g^{xyzb_i/b_j^2}, g^{(xz)^2 b_i/b_j}\}_{\forall\ i,j\in[q], i\neq j}\Big) \text{ and } Z$$

are given, no probabilistic polynomial time (PPT) algorithm $\mathcal{A}$ can distinguish $Z = Z_0 = e(g,g)^{xyz}$ from $Z = Z_1 = e(g,g)^f$ with more than a negligible advantage. The advantage of $\mathcal{A}$ is defined as $\boldsymbol{Adv}_{\mathcal{A}}^{q\text{-}RW2}(\lambda) = \big|\Pr[\mathcal{A}(D, Z_0) = 0] - \Pr[\mathcal{A}(D, Z_1) = 0]\big|$ where the probability is taken over random choices of $x, y, z, \{b_i\}_{i\in[q]}, f \in \mathbb{Z}_p$.

Lemma 1 ([18]). *The q-RW2 assumption holds in the generic group model.*

2.3 Hierarchical IBE

HIBE is an extension of IBE where an identity of a user is represented as a hierarchical structure such as $ID|_k = (I_1, \ldots, I_k)$ [7]. The syntax of HIBE is given as follows:

Definition 1 (HIBE). *An HIBE scheme consists of five algorithms* ***Setup***, ***GenKey***, ***Delegate***, ***Encrypt***, *and* ***Decrypt***, *which are defined as follows:*

Setup(1^λ). *The setup algorithm takes as input a security parameter 1^λ. It outputs a master key MK and public parameters PP.*

GenKey$(ID|_k, MK, PP)$. *The key generation algorithm takes as input an identity $ID|_k = (I_1, \ldots, I_k) \in \mathcal{I}^k$, the master key MK, and the public parameters PP. It outputs a private key $SK_{ID|_k}$ for $ID|_k$.*

Delegate$(ID|_k, SK_{ID|_{k-1}}, PP)$. *The delegation algorithm takes as input an identity $ID|_k$, a private key $SK_{ID|_{k-1}}$ for an identity $ID|_{k-1}$, and the public parameters PP. It outputs a delegated private key $SK_{ID|_k}$ for $ID|_k$.*

Encrypt$(ID|_k, M, PP)$. *The encryption algorithm takes as input an identity $ID|_k$, a message $M \in \mathcal{M}$, and the public parameters PP. It outputs a ciphertext $CT_{ID|_k}$ for $ID|_k$ and M.*

Decrypt$(CT_{ID|_k}, SK_{ID'|_\ell}, PP)$. *The decryption algorithm takes as input a ciphertext $CT_{ID|_k}$ for an identity $ID|_k$, a private key $SK_{ID'_\ell}$ for an identity ID'_ℓ, and the public parameters PP. It outputs an encrypted message M.*

The correctness of HIBE is defined as follows: For all MK, PP generated by ***Setup***, *all $ID|_k, ID'|_\ell$, any $SK_{ID|_k}$ generated by* ***GenKey***, *and any M, it is required that*

- *If $ID|_k$ is a prefix of $ID'|_\ell$, then* $\boldsymbol{Decrypt}(\boldsymbol{Encrypt}(ID'|_\ell, M, PP), SK_{ID|_k}, PP) = M$.
- *If $ID|_k$ is not a prefix of $ID'|_\ell$, then* $\boldsymbol{Decrypt}(\boldsymbol{Encrypt}(ID'|_\ell, M, PP), SK_{ID}, PP) = \perp$.

We follow the security model of HIBE given in [14]. The exact security of HIBE is given in the full version of this paper.

2.4 Revocable HIBE

RHIBE is an extension of HIBE that provides revocation functionality [21]. The syntax of RHIBE is given as follows:

Definition 2 (Revocable HIBE). *An RHIBE scheme for the identity space $\mathcal{I}$, the time space $\mathcal{T}$, and the message space $\mathcal{M}$, consists of seven algorithms* ***Setup****,* ***GenKey****,* ***UpdateKey****,* ***DeriveKey****,* ***Encrypt****,* ***Decrypt****, and* ***Revoke****, which are defined as follows:*

Setup(1^λ)*: This algorithm takes as input a security parameter 1^λ. It outputs a master key MK, an (empty) revocation list RL, a state information ST, and public parameters PP.*

GenKey$(ID|_k, ST_{ID|_{k-1}}, PP)$*: This algorithm takes as input an identity $ID|_k = (I_1, \dots, I_k) \in \mathcal{I}^k$, the state $ST_{ID|_{k-1}}$, and public parameters PP. It outputs a private key $SK_{ID|_k}$.*

UpdateKey$(T, RL_{ID|_{k-1}}, DK_{ID|_{k-1},T}, ST_{ID|_{k-1}}, PP)$*: This algorithm takes as input time $T \in \mathcal{T}$, the revocation list $RL_{ID|_{k-1}}$, the decryption key $DK_{ID|_{k-1},T}$, and public parameters PP. It outputs an update key $UK_{ID|_{k-1},T}$.*

DeriveKey$(SK_{ID|_k}, UK_{ID|_{k-1},T}, PP)$*: This algorithm takes as input a private key $SK_{ID|_k}$ for an identity $ID|_k$, an update key $UK_{ID|_{k-1},T}$ for time T, and the public parameters PP. It outputs a decryption key $DK_{ID|_k,T}$.*

Encrypt$(ID|_\ell, T, M, PP)$*: This algorithm takes as input an identity $ID|_\ell = (I_1, \dots, I_\ell) \in \mathcal{I}^\ell$, time T, a message M, and the public parameters PP. It outputs a ciphertext $CT_{ID|_\ell,T}$.*

Decrypt$(CT_{ID|_\ell,T}, DK_{ID'|_k,T'}, PP)$*: This algorithm takes as input a ciphertext $CT_{ID|_\ell,T}$, a decryption key $DK_{ID'|_k,T'}$ and the public parameters PP. It outputs an encrypted message M.*

Revoke$(ID|_k, T, RL_{ID|_{k-1}}, ST_{ID|_{k-1}})$*: This algorithm takes as input an identity $ID|_k$, revocation time T, the revocation list $RL_{ID|_{k-1}}$, and the state $ST_{ID|_{k-1}}$. It outputs the updated revocation list $RL_{ID|_{k-1}}$.*

The correctness of RHIBE is defined as follows: For all MK, RL, ST, and PP generated by ***Setup***(1^λ)*, SK_{ID} generated by* ***GenKey***(ID, MK, ST, PP) *for any ID, $UK_{T,R}$ generated by* ***UpdateKey***(T, RL, MK, ST, PP) *for any T and RL, $CT_{ID',T'}$ generated by* ***Encrypt***(ID', T', M, PP) *for any ID', T', and M, it is required that*

- *If $ID|_k$ is not revoked on time T, then* ***DeriveKey***$(SK_{ID|_k}, UK_{ID|_{k-1},T}, PP) = DK_{ID|_k,T}$.
- *If $ID|_k$ is revoked on time T, then* ***DeriveKey***$(SK_{ID|_k}, UK_{ID|_{k-1},T}, PP) = \perp$.
- *If $(ID' = ID) \wedge (T' = T)$, then* ***Decrypt***$(CT_{ID',T'}, DK_{ID,T}, PP) = M$.
- *If $(ID' \neq ID) \vee (T' \neq T)$, then* ***Decrypt***$(CT_{ID',T'}, DK_{ID,T}, PP) = \perp$.

The security model of RHIBE was introduced by Seo and Emura [21]. We follow the stronger security model of Seo and Emura [23] that considers decryption key exposure attackers and inside attackers. The detailed definition of the security model is given as follows:

Definition 3 (Selective IND-CPA Security). *The selective IND-CPA security of RHIBE is defined in terms of the following experiment between a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$:*

1. ***Init:*** *$\mathcal{A}$ initially submits a challenge identity $ID^*|_k = (I_1^*, \ldots, I_k^*)$ and challenge time T^*.*
2. ***Setup:*** *$\mathcal{C}$ runs **Setup**(1^λ) and obtains a master key MK, a revocation list RL, a state information ST, and public parameters PP. It keeps MK, RL, ST to itself and gives PP to $\mathcal{A}$.*
3. ***Phase 1:*** *$\mathcal{A}$ adaptively requests a polynomial number of queries. These queries are processed as follows:*
   - *If it is a private key query for an identity $ID|_k$, then $\mathcal{C}$ gives a private key $SK_{ID|_k}$ and a state information $ST_{ID|_k}$ by running **GenKey**$(ID|_k, ST_{ID|_{k-1}}, PP)$. There is a restriction: If $\mathcal{A}$ requested a private key query for $ID^*|_{k'}$ that is a prefix of $ID^*|_k$ where $k' \leq k$, then the identity $ID^*|_{k'}$ or one of its ancestors should be revoked at some time T where $T \leq T^*$.*
   - *If it is an update key query for an identity $ID|_{k-1}$ and time T, then $\mathcal{C}$ gives an update key $UK_{ID|_{k-1},T}$ by running **UpdateKey**$(T, RL_{ID|_{k-1}}, DK_{ID|_{k-1}}, ST_{ID|_{k-1}}, PP)$.*
   - *If it is a decryption key query for an identity $ID|_k$ and time T, then $\mathcal{C}$ gives a decryption key $DK_{ID|_k,T}$ by running **DeriveKey**$(SK_{ID|_k}, UK_{ID|_{k-1}}, PP)$. There is a restriction: $\mathcal{A}$ cannot request a private key query for the challenge identity $ID^*|_k$ or its ancestors on the challenge time T^*.*
   - *If it is a revocation query for an identity $ID|_k$ and time T, then $\mathcal{C}$ updates a revocation list $RL_{ID|_{k-1}}$ by running **Revoke**$(ID|_k, T, RL_{ID|_{k-1}}, ST_{ID|_{k-1}})$. There is a restriction: $\mathcal{A}$ cannot request a revocation query for $ID|_k$ on time T if he already requested an update key query for $ID|_k$ on time T.*

   Note that we assume that update key, decryption key, and revocation queries are requested in non-decreasing order of time.
4. ***Challenge:*** *$\mathcal{A}$ submits two challenge messages M_0^*, M_1^* with the same length. $\mathcal{C}$ flips a random coin $\mu \in \{0,1\}$ and gives the challenge ciphertext $CT_{ID^*|_k,T^*}$ to $\mathcal{A}$ by running **Encrypt**$(ID^*|_\ell, T^*, M_\mu^*, PP)$.*
5. ***Phase 2:*** *$\mathcal{A}$ may continue to request a polynomial number of queries subject to the same restrictions as before.*
6. ***Guess:*** *Finally, $\mathcal{A}$ outputs a guess $\mu' \in \{0,1\}$, and wins the game if $\mu = \mu'$.*

The advantage of $\mathcal{A}$ is defined as $\boldsymbol{Adv}_{\mathcal{A}}^{RHIBE}(\lambda) = \left|\Pr[\mu = \mu'] - \frac{1}{2}\right|$ where the probability is taken over all the randomness of the experiment. An RHIBE scheme is selectively secure under a chosen plaintext attack if for all PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the above experiment is negligible in the security parameter λ.

3 Hierarchical Identity-Based Encryption

In this section, we propose an unbounded HIBE scheme from the key-policy ABE scheme of Rouselakis and Waters [18] and prove its security.

3.1 Construction

Let $\mathcal{I} = \{0,1\}^\lambda$ be the identity space where λ is a security parameter. Our unbounded HIBE scheme is described as follows:

HIBE.Setup(1^λ): This algorithm takes as input a security parameter λ. It first runs the group generator $\mathcal{G}$ and obtains a bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e)$. Let g be a generator of $\mathbb{G}$. Next, it selects random elements $g, u, h \in \mathbb{G}$ and random exponents $x, y \in \mathbb{Z}_p$. It sets $w = g^x, v = g^y, \alpha = xy$. It outputs a master key $MK = \alpha$ and public parameters $PP = \big((p, \mathbb{G}, \mathbb{G}_T, e), g, u, h, w, v, \Omega = e(g,g)^\alpha\big)$.

HIBE.GenKey$(ID|_k, MK, PP)$: This algorithm takes as input an identity $ID|_k = (I_1, \ldots, I_k) \in \mathcal{I}^k$, the master key MK, and the public parameters PP. It chooses random exponents $r_1, \ldots, r_k \in \mathbb{Z}_p$ and outputs a private key $SK_{ID|_k} = \big(K_0 = g^\alpha \prod_{i=1}^k w^{r_i},\ \{K_{i,1} = (u^{I_i} h)^{-r_i},\ K_{i,2} = g^{r_i}\}_{i=1}^k\big)$.

HIBE.RandKey$(ID|_k, \gamma, SK_{ID|_k}, PP)$: This algorithm takes as input an identity $ID|_k = (I_1, \ldots, I_k) \in \mathcal{I}^k$, an exponent $\gamma \in \mathbb{Z}_p$, a private key $SK_{ID|_k} = (K'_0, \{K'_{i,1}, K'_{i,2}\}_{i=1}^k)$, and the public parameters PP. It chooses random exponents $r_1, \ldots, r_k \in \mathbb{Z}_p$ and outputs a re-randomized private key $SK_{ID|_k} = \big(K_0 = K'_0 \cdot g^\gamma \prod_{i=1}^k w^{r_i},\ \{K_{i,1} = K'_{i,1} \cdot (u^{I_i} h)^{-r_i},\ K_{i,2} = K'_{i,2} \cdot g^{r_i}\}_{i=1}^k\big)$.

HIBE.Delegate$(ID|_k, SK_{ID|_{k-1}}, PP)$: This algorithm takes as input an identity $ID|_k = (I_1, \ldots, I_k) \in \mathcal{I}^k$, a private key $SK_{ID|_{k-1}} = (K'_0, \{K'_{i,1}, K'_{i,2}\}_{i=1}^{k-1})$ for $ID|_{k-1}$, and the public parameters PP. It chooses a random exponent $r_k \in \mathbb{Z}_p$ and creates a temporal delegated private key $TSK_{ID|_k} = \big(K_0 = K'_0 \cdot w^{r_k},\ \{K_{i,1} = K'_{i,1},\ K_{i,2} = K'_{i,2}\}_{i=1}^{k-1},\ \{K_{k,1} = (u^{I_k} h)^{-r_k}, K_{k,2} = g^{r_k}\}\big)$. Next, it outputs a delegated private key $SK_{ID|_k}$ by running **HIBE.RandKey**$(ID|_k, 0, TSK_{ID|_k}, PP)$.

HIBE.Encrypt$(ID|_\ell, M, PP)$: This algorithm takes as input an identity $ID|_\ell = (I_1, \ldots, I_\ell) \in \mathcal{I}^\ell$, a message $M \in \mathcal{M}$, and the public parameters PP. It chooses random exponents $t, s_1, \ldots, s_k \in \mathbb{Z}_p$ and outputs a ciphertext $CT_{ID|_\ell} = \big(C = \Omega^t \cdot M,\ C_0 = g^t,\ \{C_{i,1} = g^{s_i},\ C_{i,2} = (u^{I_i} h)^{s_i} w^{-t}\}_{i=1}^\ell\big)$.

HIBE.Decrypt$(CT_{ID|_\ell}, SK_{ID'|_k}, PP)$: This algorithm takes as input a ciphertext $CT_{ID|_\ell} = (C, C_0, \{C_1, C_2\}_{i=1}^\ell)$ for $ID|_\ell$, a private key $SK_{ID'|_k} = (K_0, \{K_{i,1}, K_{i,2}\}_{i=1}^k)$ for $ID'|_k$, and the public parameters PP. If $ID'|_k$ is a prefix of $ID|_\ell$, it outputs an encrypted message by computing $M = C \cdot e(C_0, K_0)^{-1} \cdot \prod_{i=1}^k \big(e(C_{i,1}, K_{i,1}) \cdot e(C_{i,2}, K_{i,2})\big)^{-1}$. Otherwise, it outputs $\perp$.

3.2 Security Analysis

Theorem 2. *The above HIBE scheme is selectively IND-CPA secure if the q-RW2 assumption holds.*

Due to the lack of space, we briefly sketch the proof of our HIBE scheme. At first, HIBE can be considered as a kind of KP-ABE since an attribute is corresponding to an identity and an access policy in a private key just consists of the AND gate only. Our HIBE scheme is based on the KP-ABE scheme of Rouselakis and Waters [18] and the first component K_0 of a private key in our HIBE scheme is the multiplication of every components $\{K_{\tau,0}\}$ of their KP-ABE scheme. Thus, the security proof of our HIBE scheme can be easily simulated like that of their KP-ABE scheme.

4 Revocable Hierarchical Identity-Based Encryption

In this section, we propose an unbounded RHIBE scheme by using our unbounded HIBE scheme in the previous section. To provide the revocation functionality, we basically follow the design strategy of previous RIBE (or RHIBE) schemes that use a binary tree structure [1,21,23].

4.1 KUNode Algorithm

We use the KUNode algorithm of Boldyreva et al. [1] for our RHIBE scheme.

Definition 4 (KUNode Algorithm). *This algorithm takes as input a binary tree BT, a revocation list RL, and time T. It outputs a set of nodes. If η is a non-leaf node, then the left and right child node of η is denoted by η_{left} and η_{right}, respectively. Users are assigned to leaf nodes, and* ***Path***(η) *means the set of nodes on the path from η to the root node. If a user assigned to η is revoked on time T, then $(\eta, T) \in RL$. The algorithm is given below.*

```
KUNode(BT,RL,T):
    $X, Y \leftarrow \emptyset$
    $\forall(\eta_i, T_i) \in RL$
        If $T_i \leq T$ then add Path($\eta_i$) to $X$
    $\forall x \in X$
        If $x_{left} \notin X$ then add $x_{left}$ to $Y$
        If $x_{right} \notin X$ then add $x_{right}$ to $Y$
    If $Y \neq \emptyset$ then add root to $Y$
    Return $Y$
```

When a user requests a private key to a KGC, the KGC assigns a user to the leaf node η of a binary tree BT, and generates a private key. A private key is associated with the set of nodes $\mathbf{Path}(\eta)$. The KGC publishes the update key for a set $\mathbf{KUNode}(BT, RL, T)$ at time T, then only unrevoked users have at least one node in $\mathbf{Path}(\eta) \cap \mathbf{KUNode}(BT, RL, ST)$. Unrevoked users can derive the decryption key combining the secret key and the update key in that node.

4.2 Construction

Let $\mathcal{I} = \{0,1\}^\lambda$ be the identity space and $\mathcal{T} = \{0,1\}^\lambda$ be the time space where λ is a security parameter. Our RHIBE scheme from our HIBE scheme is described as follows:

RHIBE.Setup(1^λ): This algorithm takes as input a security parameter 1^λ. It first runs the group generator $\mathcal{G}$ and obtains a bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e)$. Let g be a generator of $\mathbb{G}$. Next, it selects random elements $u, h, u_0, h_0 \in \mathbb{G}$ and random exponents $x, y \in \mathbb{Z}_p$. It sets $w = g^x, v = g^y, \alpha = xy$. It outputs a master key $MK = \alpha$ and public parameters $PP = \big((p, \mathbb{G}, \mathbb{G}_T, e), g, u, h, w, v, \Omega = e(g,g)^\alpha,\ u_0, h_0\big)$.

RHIBE.GenKey$(ID|_k, ST_{ID|_{k-1}}, PP)$: This algorithm takes as input an identity $ID|_k = (I_1, \ldots, I_k) \in \mathcal{I}^k$, the state $ST_{ID|_{k-1}}$, and public parameters PP. Note that the state $ST_{ID|_{k-1}}$ contains $BT_{ID|_{k-1}}$.

1. It first assigns $ID|_k$ to a random leaf node in $BT_{ID|_{k-1}}$. Let $Path$ be a path node set defined by $\mathbf{Path}(ID|_k) \in BT_{ID|_{k-1}}$.
2. For each node $\theta \in Path$, it performs the following steps: It first retrieves $\gamma_\theta \in \mathbb{Z}_p$ from $BT_{ID|_{k-1}}$ where γ_θ is associated to the node θ. Note that if γ_θ is not defined, then it chooses a random exponent $\gamma_\theta \in \mathbb{Z}_p$ and stores it to the node θ. Next, it creates a partial private key $PSK_\theta = \big(K_0, \{K_{i,1}, K_{i,2}\}_{i=1}^k\big)$ by running **HIBE.GenKey**$(ID|_k, \gamma_\theta, PP)$.
3. Finally, it outputs a private key $SK_{ID|_k} = \big(\{\theta, PSK_\theta\}_{\theta \in Path}\big)$.

RHIBE.UpdateKey$(T, RL_{ID|_{k-1}}, DK_{ID|_{k-1},T}, ST_{ID|_{k-1}}, PP)$:
This algorithm takes as input time $T \in \mathcal{T}$, the revocation list $RL_{ID|_{k-1}}$, the decryption key $DK_{ID|_{k-1},T}$, the state $ST_{ID|_{k-1}}$ where it contains $BT_{ID|_{k-1}}$, and public parameters PP. Recall that $RL_{ID|_0} = RL_0$ and $ST_{ID|_0} = ST_0$. Note that $DK_{ID|_0,T} = \big(D_0 = g^\alpha (u_0^T h_0)^{-r_0}, D_1 = g^{r_0}\big)$ can be easily generated by using MK.

1. Let $KUNode$ be a covering set that is obtained by running **KUNode**$(BT_{ID|_{k-1}}, RL_{ID|_{k-1}}, T)$.
2. For each node $\theta \in KUNode$, it performs the following steps: It first retrieves $\gamma_\theta \in \mathbb{Z}_p$ from $BT_{ID|_{k-1}}$ where γ_θ is associated to the node θ.
   It obtains $DK'_{ID|_{k-1},T} = \big(D'_0, D'_1, \{D'_{i,1}, D'_{i,2}\}_{i=1}^{k-1}\big)$ by running **RHIBE.RandDK**$(DK_{ID|_{k-1},T}, PP)$. Next, it creates a time-constrained update key $TUK_\theta = \big(U_0 = g^{-\gamma_\theta} \cdot D'_0,\ U_1 = D'_1,\ \big\{U_{i,1} = D'_{i,1},\ U_{i,2} = D'_{i,2}\big\}_{i=1}^{k-1}\big)$.
3. Finally, it outputs an update key $UK_{ID|_{k-1},T} = \big(\{\theta, TUK_\theta\}_{\theta \in KUNode}\big)$.

RHIBE.DeriveKey$(SK_{ID|_k}, UK_{ID|_{k-1},T}, PP)$: This algorithm takes as input a private key $SK_{ID|_k}$ for an identity $ID|_k$, an update key $UK_{ID|_{k-1},T}$ for time T and the public parameters PP.

1. If $ID|_k \notin RL_{ID|_{k-1}}$, then it finds a unique node $\theta^* \in \mathbf{Path}(ID|_k) \cap$ **KUNode**$(BT_{ID|_{k-1}}, RL_{ID|_{k-1}}, T)$. Otherwise, it outputs $\perp$.
2. It derives $PSK_{\theta^*} = \big(K_0, \{K_{i,1}, K_{i,2}\}_{i=1}^k\big)$ from $SK_{ID|_k}$ and $TUK_{\theta^*} = \big(U_0, U_1, \{U_{i,1}, U_{i,2}\}_{i=1}^{k-1}\big)$ from $UK_{ID|_{k-1},T}$ for the node θ^*. Next, it creates a decryption key $DK_{ID|_k,T} = \big(D_0 = K_0 \cdot U_0,\ D_1 = U_1,\ \big\{D_{i,1} = K_{i,1} \cdot U_{i,1},\ D_{i,2} = K_{i,2} \cdot U_{i,2}\big\}_{i=1}^{k-1},\ \big\{D_{k,1} = K_{k,1},\ D_{k,2} = K_{k,2}\big\}\big)$ and rerandomizes it by running **RHIBE.RandDK**.

3. Finally, it outputs a (re-randomized) decryption key $DK_{ID|_k,T} = \big(D_0, D_1, \{D_{i,1}, D_{i,2}\}_{i=1}^{k}\big)$.

RHIBE.RandDK$(DK_{ID|_k,T}, PP)$: This algorithm takes as input a decryption key $DK_{ID|_k} = \big(D'_0, D'_1, \{D'_{i,1}, D'_{i,2}\}_{i=1}^{k}\big)$ for an identity $ID|_k = (I_1, I_2, \ldots, I_k) \in \mathcal{I}^k$ and time T, and the public parameters PP. It selects random exponents $r_0, r_1, \ldots, r_k \in \mathbb{Z}_p$ and outputs a re-randomized decryption key $DK_{ID|_k,T} = \big(D_0 = D'_0 \cdot (u_0^T h_0)^{-r_0} \prod_{i=1}^{k} w^{r_i},\ D_1 = D'_1 \cdot g^{r_0},\ \big\{D_{i,1} = D'_{i,1} \cdot (u^{I_i} h)^{-r_i},\ D_{i,2} = D'_{i,2} \cdot g^{r_i}\big\}_{i=1}^{k}\big)$.

RHIBE.Encrypt$(ID|_\ell, T, M, PP)$: This algorithm takes as input an identity $ID|_\ell = (I_1, \ldots, I_\ell) \in \mathcal{I}^k$, time T, a message M, and the public parameters PP. It first chooses random exponents $t, s_1, \ldots, s_\ell \in \mathbb{Z}_p$ and outputs a ciphertext $CT_{ID|_k,T} = \big(C = \Omega^t \cdot M,\ C_0 = g^t,\ C_1 = (u_0^T h_0)^t,\ \big\{C_{i,1} = g^{s_i},\ C_{i,2} = w^{-t}(u^{I_i} h)^{s_i}\big\}_{i=1}^{\ell}\big)$.

RHIBE.Decrypt$(CT_{ID|_\ell,T}, DK_{ID'|_k,T'}, PP)$: This algorithm takes as input a ciphertext $CT_{ID|_\ell,T} = (C, C_0, C_1, \{C_{i,1}, C_{i,2}\}_{i=1}^{\ell})$, a decryption key $DK_{ID'|_k,T'} = (D_0, D_1, \{D_{i,1}, D_{i,2}\}_{i=1}^{k})$ and the public parameters PP. If $ID'|_k$ is a prefix of $ID|_\ell$ and $T = T'$, then it outputs an encrypted message $M = C \cdot \big(e(C_0, D_0) \cdot e(C_1, D_1) \cdot \prod_{i=1}^{k} \big(e(C_{i,1}, D_{i,1}) \cdot e(C_{i,2}, D_{i,2})\big)\big)^{-1}$ Otherwise, it outputs $\perp$.

RHIBE.Revoke$(ID|_k, T, RL_{ID|_{k-1}}, ST_{ID|_{k-1}})$: This algorithm takes as input an identity $ID|_k$, revocation time T, the revocation list $RL_{ID|_{k-1}}$, and the state $ST_{ID|_{k-1}}$. If $(ID|_k, -) \notin ST_{ID|_{k-1}}$, then it outputs $\perp$ since the private key of $ID|_k$ was not generated. Otherwise, it adds $(ID|_k, T)$ to $RL_{ID|_{k-1}}$ and outputs the updated revocation list $RL_{ID|_{k-1}}$.

4.3 Security Analysis

Theorem 3. *The above RHIBE scheme is selectively IND-CPA secure if the underlying HIBE scheme is selectively IND-CPA secure.*

The proof of this theorem is given in the full version of this paper.

5 Conclusion

In this paper, we proposed the first unbounded RHIBE scheme using proposed HIBE and the history-free approach of Seo and Emura [23]. To achieve our scheme, we first proposed an unbounded HIBE scheme from the KP-ABE scheme of Rouselakis and Waters [18]. Our proposed RHIBE scheme makes it efficient to generate private keys in IBE for a large number of users since it allows the delegation of the key generation using a hierarchical structure among users and provides the revocation functionality. Furthermore it solves the open problem of removing the limitation on maximum hierarchy.

The security of our RHIBE scheme was proved in the selective model. It will be interesting to construct a fully secure RHIBE scheme with no limitations on maximum hierarchy. Our RHIBE scheme essentially uses the complete subtree (CS) method for revocation. We expect that the subset difference (SD) method also can be applied to our RHIBE scheme since Seo and Emura [23] also proposed an RHIBE scheme that uses the SD method by following the methodology of Lee et al. [11].

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2010-0029121). The first two authors (Geumsook Ryu and Kwangsu Lee) equally contributed to this work.

References

1. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. In: Ning, P., Syverson, P.F., Jha, S., (eds.) ACM Conference on Computer and Communications Security, pp. 417–426. ACM (2008)
2. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
3. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
4. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, p. 213. Springer, Heidelberg (2001)
5. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
6. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
7. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
8. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
9. Lee, K.: Self-updatable encryption with short public parameters and its extensions. Des. Codes Crypt. 1–41 (2015). http://dx.doi.org/10.1007/s10623-015-0039-9
10. Lee, K., Choi, S.G., Lee, D.H., Park, J.H., Yung, M.: Self-updatable encryption: time constrained access control with hidden attributes and better efficiency. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 235–254. Springer, Heidelberg (2013)
11. Lee, K., Lee, D.H., Park, J.H.: Efficient revocable identity-based encryption via subset difference methods. Cryptology ePrint Archive, Report 2014/132.(2014). http://eprint.iacr.org/2014/132
12. Lee, K., Park, J.H., Lee, D.H.: Anonymous HIBE with short ciphertexts: full security in prime order groups. Des. Codes Crypt. **74**(2), 395–425 (2015)

13. Lewko, A., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
14. Lewko, A., Waters, B.: Unbounded HIBE and attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 547–567. Springer, Heidelberg (2011)
15. Libert, B., Vergnaud, D.: Adaptive-ID secure revocable identity-based encryption. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 1–15. Springer, Heidelberg (2009)
16. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
17. Park, S., Lee, K., Lee, D.H.: New constructions of revocable identity-based encryption from multilinear maps. IEEE Trans. Inf. Forensic Secur. **10**(8), 1564–1577 (2015)
18. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: Sadeghi, A.R., Gligor, V.D., Yung, M., (eds.) ACM Conference on Computer and Communications Security, pp. 463–474. ACM (2013)
19. Sahai, A., Seyalioglu, H., Waters, B.: Dynamic credentials and ciphertext delegation for attribute-based encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 199–217. Springer, Heidelberg (2012)
20. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
21. Seo, J.H., Emura, K.: Efficient delegation of key generation and revocation functionalities in identity-based encryption. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 343–358. Springer, Heidelberg (2013)
22. Seo, J.H., Emura, K.: Revocable identity-based encryption revisited: security model and construction. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 216–234. Springer, Heidelberg (2013)
23. Seo, J.H., Emura, K.: Revocable hierarchical identity-based encryption: history-free update, security against insiders, and short ciphertexts. In: Nyberg, K. (ed.) CT-RSA 2015. LNCS, vol. 9048, pp. 106–123. Springer, Heidelberg (2015)
24. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
25. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
26. Waters, B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)