Generic Constructions of Revocable Hierarchical Identity-based Encryption

Keita Emura[1], Atsushi Takayasu[1], and Yohei Watanabe[2,3]

[1] National Institute of Information and Communications Technology (NICT), Japan
{k-emura, takayasu}@nict.go.jp
[2] The University of Electro-Communications, Japan
watanabe@uec.ac.jp
[3] National Institute of Advanced Industrial Science and Technology, Japan

Abstract. Revocable hierarchical identity-based encryption (RHIBE) is an extension of hierarchical identity-based encryption (HIBE) supporting the key revocation mechanism. In this paper, we propose a generic construction of RHIBE from HIBE with the complete subtree method. Then, we obtain the first RHIBE schemes under the quadratic residuosity assumption, CDH assumption without pairing, factoring Blum integers, LPN assumption, and code-based assumption, and the first almost tightly secure RHIBE schemes under the k-linear assumption. Furthermore, by using pairing-based (dual) identity-based broadcast encryption, we obtain the variants of the scheme with shorter ciphertexts or shorter key updates.

1 Introduction

(Hierarchical) dentity-based encryption ((H)IBE) is an extension of the traditional public key encryption. (H)IBE can use any string as each user's public key and HIBE has delegatable secret keys. HIBE schemes have been constructed based on pairing-based assumptions (e.g., [10,12,20,23,24,25,41,42]) and learning with errors (LWE) assumption [1,2,5,7]. Moreover, since IBE implies HIBE [14], IBE schemes under the quadratic residuosity (QR) assumption [13], CDH assumption without pairing and factoring Blum integers [15], LPN assumption [6], and code-based assumption [19] imply HIBE schemes based on the same assumptions. Among them, only pairing-based HIBE schemes [10,12,20,23,24,25,41,42] satisfy adaptive security in the standard model. Furthermore, [23,24,25] achieve almost tight security under the k-linear assumption.

Due to the absence of the public key infrastructure, the key revocation functionality is indispensable property to use (H)IBE in practice. In particular, the functionality enables the system to revoke malicious users dynamically and efficiently. Starting with the seminal work of Boldyreva et al. [4], several (non-hierarchical) revocable IBE (RIBE) schemes have been proposed under various assumptions such as pairing-based assumptions (e.g., [4,16,28,30,35,39]), LWE assumption [11,38,22], CDH assumption without pairing and factoring Blum integers [21], and code-based assumption [8]. All the constructions utilize Naor

et al.'s subset cover frameworks [33] such as the complete subtree method (CS) and subset difference method (SD). Specifically, all the schemes use the CS except that pairing-based scheme [28] used the SD. Ma and Lin [31] proposed a generic construction of RIBE from IBE with the CS. Ma and Lin's RIBE schemes have shorter secret keys and larger ciphertexts than the other direct constructions. By following Ma and Lin's framework, Ma and Lin [32] and Lee [26] proposed a generic construction of RIBE from *variants* of IBE with the SD. Unfortunately, they could construct the variants of IBE based only on pairing-based assumptions. Revocable HIBE (RHIBE) was introduced by Seo and Emura [36]. As opposed to RIBE, there are only a few constructions of RHIBE schemes based only on pairing-based assumptions [17,18,29,34,37] and LWE assumption [22,40] since there are no generic constructions of RHIBE[4] such as [31,32,26]. All the constructions except [29] used the CS. Since only [29] used the SD, the scheme has shorter key updates than the other RHIBE schemes; however, [29] satisfies only selective revocation list security that is weaker than selective security under a q-type assumption.

Our Contributions. In this paper, we propose a generic construction of RHIBE from HIBE with the CS by extending Ma and Lin's generic construction of RIBE [31]. As a result, we obtain a result that HIBE with the CS implies RHIBE. Therefore, we obtain the first RHIBE scheme under various assumptions such as the QR assumption [13], CDH assumption without pairing and factoring Blum integers [15], LPN assumption [6], and code-based assumption [19] by combining with Döttling and Garg's technique [14]. The resulting RHIBE scheme of our generic framework is a hierarchical extension of Ma and Lin's generic RIBE scheme with the CS [31]. Thus, we obtain the first adaptively secure RHIBE scheme with short secret keys. Furthermore, since the reductions of our framework are almost tight, we obtain the first almost tightly secure RHIBE schemes with adaptive security under the k-linear assumption. Furthermore, we use pairing-based hierarchical identity-based (dual) identity-based broadcast encryption and propose adaptively secure RHIBE schemes with shorter cipehrtexts or shorter key updates.

Independent and Concurrent Work. Recently, Lee and Kim [27] proposed a generic construction of RHIBE. Their first result is a generic construction of RHIBE from HIBE with the CS. Therefore, their construction is almost the same as ours. Nevertheless, our scheme is slightly more efficient than Lee and Kim's scheme since we use one HIBE scheme for constructing RHIBE whereas Lee and Kim used two HIBE schemes. Lee and Kim also proposed a shorter ciphertext variant, while they did not propose a shorter key update variant. As opposed to our shorter ciphertext variant from pairing-based hierarchical identity-based identity-based broadcast encryption, Lee and Kim used HIBE with compact ciphertexts. Lee and Kim also proposed a generic construction of RHIBE with the SD; however, the construction requires not familiar hierarchical identity-based

[4] Emura et al.'s construction [18] is a semi-generic construction from HIBE with a few additional properties that several pairing-based HIBE schemes satisfy. The reduction loss depends on the number of secret key queries made by an adversary.

single revocation encryption. Indeed, they showed only one instantiation of a hierarchical identity-based single revocation encryption scheme with selective security from the DBDH assumption.

2 Preliminaries

Notations. Let λ denote the security parameter. For non-negative integers a, b with $a \leq b$, we define $[a, b] \coloneqq \{a, a+1, \ldots, b\}$ and $[a] \coloneqq [1, a]$. For a finite set S, let $x \leftarrow_R S$ denote sampling x from S uniformly at random. For a κ_1-bit binary string $\eta_1 \in \{0,1\}^{\kappa_1}$ and a κ_2-bit binary string $\eta_2 \in \{0,1\}^{\kappa_2}$, let $\eta_1 \| \eta_2 \in \{0,1\}^{\kappa_1+\kappa_2}$ denote a concatenation of η_1 and η_2. Similarly, let $\{0,1\}^{\kappa_1} \| \{0,1\}^{\kappa_2} = \{0,1\}^{\kappa_1+\kappa_2}$. Furthermore, let $\{0,1\}^{\kappa} \| \{0\}$ and $\{0,1\}^{\kappa} \| \{1\}$ denote sets of $(\kappa+1)$-bit binary strings whose last bit is 0 and 1, respectively.

Hierarchical Identity and Time Period. Let $\mathcal{I} \coloneqq \{0,1\}^{\kappa_{\mathcal{I}}}$ denote an element identity space and $\mathcal{T} \coloneqq \{0,1\}^{\kappa_{\mathcal{T}}}$ denote a time period space. In this case, let $\mathcal{I}^{\leq L}$ denote an identity space of RHIBE with hierarchical depth L. Let an ℓ-dimensional vector $\mathtt{ID} = (\mathtt{id}_1, \ldots, \mathtt{id}_\ell) \in \mathcal{I}^\ell$ denote an identity at level ℓ. Let $|\mathtt{ID}| \coloneqq \ell$ denote a hierarchical level of $\mathtt{ID}$. For notational convenience, we regard kgc as a "root" user, and let $\mathcal{I}^0 \coloneqq \{\mathsf{kgc}\}$ unless otherwise stated. We define several notations for prefix of an identity $\mathtt{ID} = (\mathtt{id}_1, \cdots, \mathtt{id}_{|\mathtt{ID}|})$. Let $\mathsf{pa}(\mathtt{ID}) \coloneqq (\mathtt{id}_1, \cdots, \mathtt{id}_{|\mathtt{ID}|-1})$ denote a direct ancestor of $\mathtt{ID}$. In general, let $\mathtt{ID}_{[\ell]} \coloneqq (\mathtt{id}_1, \ldots, \mathtt{id}_\ell)$ denote an ℓ-dimensional prefix of $\mathtt{ID}$ for a non-negative integer $\ell \leq |\mathtt{ID}|$. By definition, $\mathtt{ID}_{[0]} = \mathsf{kgc}$ for any $\mathtt{ID} \in \mathcal{I}^{\leq L}$ unless otherwise stated. As the case of binary strings, let $\mathsf{prefix}(\mathtt{ID}) \coloneqq \{\mathtt{ID}_{[1]}, \mathtt{ID}_{[2]}, \ldots, \mathtt{ID}_{|\mathtt{ID}|-1} = \mathsf{pa}(\mathtt{ID})\}$ a set of all prefixes of $\mathtt{ID}$ and let $\mathsf{prefix}^+(\mathtt{ID}) \coloneqq \mathsf{prefix}(\mathtt{ID}) \cup \{\mathtt{ID}\}$.

2.1 HIBE

Let $\mathsf{HIBE}(L)$ denote the HIBE with the maximum depth L.

Syntax. $\mathsf{HIBE}(L)$ consists of the following four algorithms ($\mathsf{HIBE.Setup}$, $\mathsf{HIBE.Enc}$, $\mathsf{HIBE.KeyGen}$, $\mathsf{HIBE.Dec}$):

$\mathsf{HIBE.Setup}(1^\lambda, L) \to (\mathsf{HIBE.pp}, \mathsf{HIBE.msk})$: The *setup* algorithm takes the security parameter 1^λ and the maximum hierarchical depth L as input, and outputs a public parameter $\mathsf{HIBE.pp}$ and master secret key $\mathsf{HIBE.msk}$.

$\mathsf{HIBE.Enc}(\mathsf{HIBE.pp}, \mathtt{ID}, \mathsf{M}) \to \mathsf{HIBE.ct}_{\mathtt{ID}}$: The *encryption* algorithm takes a $\mathsf{HIBE.pp}$, $\mathtt{ID} \in \mathcal{I}^{\leq L}$, and plaintext $\mathsf{M} \in \mathcal{M}$ as input, and outputs a ciphertext $\mathsf{HIBE.ct}_{\mathtt{ID}}$.

$\mathsf{HIBE.KeyGen}(\mathsf{HIBE.pp}, \mathsf{HIBE.sk}_{\mathtt{ID}'}, \mathtt{ID}) \to \mathsf{HIBE.sk}_y$: The *secret key generation* algorithm takes a $\mathsf{HIBE.pp}$, $\mathtt{ID}$'s secret key $\mathsf{HIBE.sk}_{\mathtt{ID}'}$, and $\mathtt{ID} \in \mathcal{I}^{\leq L}$ as input, and outputs a secret key $\mathsf{HIBE.sk}_{\mathtt{ID}}$. The algorithm can take $\mathsf{HIBE.msk}$ as input in place of $\mathsf{HIBE.sk}_{\mathtt{ID}'}$.

$\mathsf{HIBE.Dec}(\mathsf{HIBE.pp}, \mathsf{HIBE.sk}_{\mathtt{ID}}, \mathsf{HIBE.ct}_{\mathtt{ID}}) \to \mathsf{M}$ or $\perp$: The *decryption* algorithm takes $\mathsf{HIBE.pp}$, $\mathsf{HIBE.sk}_{\mathtt{ID}}$, and $\mathsf{HIBE.ct}_{\mathtt{ID}}$ as input, and outputs M or $\perp$.

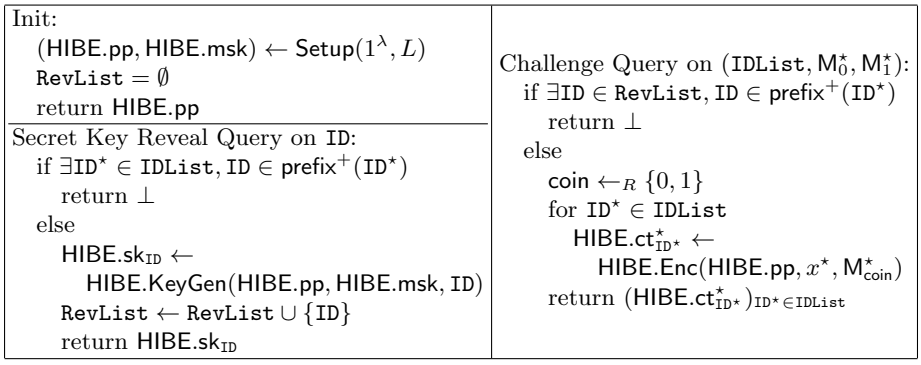

Fig. 1: Security game of $\mathsf{HIBE}(L)$

Correctness. Roughly speaking, we define correctness for HIBE.KeyGen and HIBE.Dec. The correctness for HIBE.KeyGen requires that for all $\lambda \in \mathbb{N}$, $L \in \mathbb{N}$, $(\mathsf{HIBE.pp}, \mathsf{HIBE.msk}) \leftarrow \mathsf{HIBE.Setup}(1^\lambda, L)$, $\mathtt{ID}, \mathtt{ID}' \in \mathcal{I}^{\leq L}$ such that $\mathtt{ID}' \in \mathsf{prefix}(\mathtt{ID})$, it holds that two distributions $\mathsf{HIBE.KeyGen}(\mathsf{HIBE.pp}, \mathsf{HIBE.msk}, \mathtt{ID})$ and $\mathsf{HIBE.KeyGen}(\mathsf{HIBE.pp}, \mathsf{HIBE.sk}_{\mathtt{ID}'}, \mathtt{ID})$ are statistically close. The correctness for HIBE.Dec requires that for all $\lambda \in \mathbb{N}$, $L \in \mathbb{N}$, $(\mathsf{HIBE.pp}, \mathsf{HIBE.msk}) \leftarrow \mathsf{HIBE.Setup}(1^\lambda, L)$, $\mathsf{M} \in \mathcal{M}$, and $\mathtt{ID} \in \mathcal{I}^{\leq L}$, it holds that $\mathsf{M} = \mathsf{M}'$ with overwhelming probability after executing $\mathsf{HIBE.ct}_{\mathtt{ID}} \leftarrow \mathsf{HIBE.Enc}(\mathsf{HIBE.pp}, \mathtt{ID}, \mathsf{M})$, $\mathsf{HIBE.sk}_{\mathtt{ID}} \leftarrow \mathsf{HIBE.KeyGen}(\mathsf{HIBE.pp}, \mathsf{HIBE.msk}, \mathtt{ID})$, and $\mathsf{M}' \leftarrow \mathsf{HIBE.Dec}(\mathsf{HIBE.pp}, \mathsf{HIBE.sk}_{\mathtt{ID}}, \mathsf{HIBE.ct}_{\mathtt{ID}})$.

Security. We define adaptive security of an HIBE scheme Π as the security game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$. $\mathcal{A}$ is allowed to make secret key queries polynomially many times and challenge query only once. As opposed to the standard security definition, $\mathcal{A}$ makes a challenge query on multiple identities with the same plaintexts. For this purpose, let $\mathtt{IDList} \subset \mathcal{I}^{\leq L}$ denote a set of identities. In Figure 1, we describe a behavior of $\mathcal{C}$ upon $\mathcal{A}$'s queries. We note that $\mathcal{C}$ runs Init upon $\mathcal{A}$'s query on $\mathtt{ID}^\star$ in a selective security game. At the end of the game, $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$. In this game, $\mathcal{A}$'s advantage is defined by $\mathsf{Adv}^{\mathsf{HIBE}(L)}_{\Pi,\mathcal{A}}(\lambda) := |\Pr[\widehat{\mathsf{coin}} = \mathsf{coin}] - 1/2|$. We say that Π satisfies adaptive security if $\mathsf{Adv}^{\mathsf{HIBE}(L)}_{\Pi,\mathcal{A}}(\lambda)$ is negligible for all PPT adversaries $\mathcal{A}$. We note that any HIBE scheme with the standard security definition such as $|\mathtt{IDList}| = 1$ achieves our security definition with $|\mathtt{IDList}|$ reduction loss.

2.2 RHIBE

We review the most strict Katsumata et al.'s definition [22].

Syntax. Let $\mathsf{RHIBE}(L)$ denote RHIBE with the depth L. An $\mathsf{RHIBE}(L)$ scheme Π consists of the six algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{GenSK}, \mathsf{KeyUp}, \mathsf{GenDK}, \mathsf{Dec})$ and a revoke algorithm. All parent users $\mathsf{pa}(\mathtt{ID}) \in \mathcal{I}^{\leq L-1}$ (including $\mathsf{kgc} \in \mathcal{I}^0$) keep a

revocation list $\mathsf{RL}_{\mathsf{pa(ID)},\mathtt{T}}$ at time period $\mathtt{T}$. When $\mathsf{pa}(\mathtt{ID})$ revokes their child user $\mathtt{ID}$, they update $\mathsf{RL}_{\mathsf{pa(ID)},\mathtt{T}} \leftarrow \mathsf{RL}_{\mathsf{pa(ID)},\mathtt{T}} \cup \{\mathtt{ID}\}$.

$\mathsf{Setup}(1^\lambda, L) \to (\mathsf{pp}, \mathsf{sk}_{\mathsf{kgc}})$: The *setup* algorithm takes the security parameter 1^λ and the depth L as input, and outputs a public parameter pp and kgc's secret key $\mathsf{sk}_{\mathsf{kgc}}$. pp implicitly contains the description of $\mathcal{I}$, $\mathcal{T}$, and plaintext space $\mathcal{M}$ that are determined only by λ.

$\mathsf{Enc}(\mathsf{pp}, \mathtt{ID}, \mathtt{T}, \mathsf{M}) \to \mathsf{ct}_{\mathtt{ID},\mathtt{T}}$: The *encryption* algorithm takes a pp, identity $\mathtt{ID} \in \mathcal{I}^{\leq L}$, time period $\mathtt{T} \in \mathcal{T}$, and plaintext $\mathsf{M} \in \mathcal{M}$ as input, and outputs a ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$.

$\mathsf{GenSK}(\mathsf{pp}, \mathsf{sk}_{\mathsf{pa(ID)}}, \mathtt{ID}) \to \mathsf{sk}_{\mathtt{ID}}$: The *secret key generation* algorithm takes a pp, $\mathsf{pa}(\mathtt{ID})$'s secret key $\mathsf{sk}_{\mathsf{pa(ID)}}$, and $\mathtt{ID}$ as input, and outputs a secret key $\mathsf{sk}_{\mathtt{ID}}$.

$\mathsf{KeyUp}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa(ID)},\mathtt{T}}, \mathsf{RL}_{\mathtt{ID},\mathtt{T}}, \mathtt{T}) \to \mathsf{ku}_{\mathtt{ID},\mathtt{T}}$: The *key update generation* algorithm takes a pp, $\mathtt{ID}$'s secret key $\mathsf{sk}_{\mathtt{ID}}$, $\mathsf{pa}(\mathtt{ID})$'s key update $\mathsf{ku}_{\mathsf{pa(ID)},\mathtt{T}}$ and revocation list $\mathsf{RL}_{\mathsf{pa(ID)},\mathtt{T}}$ at a time period $\mathtt{T} \in \mathcal{T}$, and $\mathtt{T}$ as input, and outputs an $\mathtt{ID}$'s key update $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$ at $\mathtt{T}$. In the special case, we define $\mathsf{ku}_{\mathsf{pa(kgc)},\mathtt{T}} := \bot$ for all $\mathtt{T} \in \mathcal{T}$.

$\mathsf{GenDK}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa(ID)},\mathtt{T}}) \to \mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ or $\bot$: The *decryption key generation* algorithm takes a pp, $\mathtt{ID}$'s secret key $\mathsf{sk}_{\mathtt{ID}}$, and $\mathsf{pa}(\mathtt{ID})$'s key update $\mathsf{ku}_{\mathsf{pa(ID)},\mathtt{T}}$ as input, and outputs a decryption key $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ if $\mathtt{ID} \notin \mathsf{RL}_{\mathsf{pa(ID)},\mathtt{T}}$ and $\bot$ otherwise.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T}}, \mathsf{ct}_{\mathtt{ID},\mathtt{T}}) \to \mathsf{M}$: The *decryption* algorithm takes pp, $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$, and $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$ as input, and outputs M or $\bot$.

Correctness. For all $\lambda \in \mathbb{N}$, $L \in \mathbb{N}$, $(\mathsf{pp}, \mathsf{sk}_{\mathsf{kgc}}) \leftarrow \mathsf{Setup}(1^\lambda, L)$, $\mathsf{M} \in \mathcal{M}$, $\mathtt{ID} \in \mathcal{I}^{\leq L}$, $\mathtt{T} \in \mathcal{T}$, and $\mathsf{RL}_{\mathsf{pa(ID}_{[\ell]}),\mathtt{T}}$ such that $\mathtt{ID}_{[\ell]} \notin \mathsf{RL}_{\mathsf{pa(ID}_{[\ell]}),\mathtt{T}}$ for $\ell \in [|\mathtt{ID}|]$, it holds that $\mathsf{M} = \mathsf{M}'$ with overwhelming probability after executing $\mathsf{ct}_{\mathtt{ID},\mathtt{T}} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathtt{ID}, \mathtt{T}, \mathsf{M})$, $\mathsf{sk}_{\mathtt{ID}_{[\ell]}} \leftarrow \mathsf{GenSK}(\mathsf{pp}, \mathsf{sk}_{\mathsf{pa(ID}_{[\ell]})}, \mathtt{ID}_{[\ell]})$ and $\mathsf{ku}_{\mathsf{pa(ID}_{[\ell]}),\mathtt{T}} \leftarrow \mathsf{KeyUp}(\mathsf{pp}, \mathsf{sk}_{\mathsf{pa(ID}_{[\ell]})}, \mathsf{ku}_{\mathsf{pa(pa(ID}_{[\ell]}))},\mathtt{T}, \mathsf{RL}_{\mathsf{pa(ID}_{[\ell]}),\mathtt{T}}, \mathtt{T})$ for $\ell = 1, 2, \ldots, |\mathtt{ID}|$, $\mathsf{dk}_{\mathtt{ID},\mathtt{T}} \leftarrow \mathsf{GenDK}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa(ID)},\mathtt{T}})$, and $\mathsf{M}' \leftarrow \mathsf{Dec}(\mathsf{pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T}}, \mathsf{ct}_{\mathtt{ID},\mathtt{T}})$.

Security. We define adaptive security of an $\mathsf{RHIBE}(L)$ scheme Π as the security game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$. The game has the global counter $\mathtt{T}_{\mathsf{cu}}$ initialized with 1 to denote the "current time period". $\mathcal{A}$ is allowed to make five types of queries. $\mathcal{A}$ can make secret key generation queries, secret key reveal queries, and decryption key reveal queries polynomially many times, revoke & key update queries $|\mathcal{T}| - 1$ times, and challenge query only once. In Figure 2, we describe a behavior of $\mathcal{C}$ which is controlled by $\mathtt{T}_{\mathsf{cu}}$ upon $\mathcal{A}$'s queries. We note that $\mathcal{C}$ runs Init upon $\mathcal{A}$'s query on $(\mathtt{ID}^\star, \mathtt{T}^\star)$ in a selective security game. At the end of the game, $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$. In this game, $\mathcal{A}$'s advantage is defined by $\mathsf{Adv}^{\mathsf{RHIBE}(L)}_{\Pi,\mathcal{A}}(\lambda) := |\Pr[\widehat{\mathsf{coin}} = \mathsf{coin}] - 1/2|$. We say that Π satisfies adaptive security if $\mathsf{Adv}^{\mathsf{RHIBE}(L)}_{\Pi,\mathcal{A}}(\lambda)$ is negligible for all PPT adversaries $\mathcal{A}$.

2.3 Complete Subtree Method

We review Naor et al.'s subset cover framework [33] called the complete subtree method (CS).

Init:
$(\mathsf{pp}, \mathsf{sk}_{\mathsf{kgc}}) \leftarrow \mathsf{Setup}(1^\lambda, L)$
$\mathtt{T}_{\mathtt{cu}} = 1$, $\mathtt{SKList} = \{(\mathsf{kgc}, \mathsf{sk}_{\mathsf{kgc}})\}$, $\mathtt{RevList} = \emptyset$
$\mathsf{ku}_{\mathsf{kgc},1} \leftarrow \mathsf{KeyUp}(\mathsf{MPK}, \mathsf{sk}_{\mathsf{kgc}}, \perp, \mathsf{RL}_{\mathsf{kgc},1} = \emptyset, \mathtt{T}_{\mathtt{cu}} = 1)$
return pp, $\mathsf{ku}_{\mathsf{kgc},1}$

Secret Key Generation Query on $\mathtt{ID}$:
if $(\mathtt{ID}, \mathsf{sk}_{\mathtt{ID}}) \in \mathtt{SKList} \vee (\mathsf{pa}(\mathtt{ID}), \mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}) \notin \mathtt{SKList}$
return $\perp$
else
$\mathsf{sk}_{\mathtt{ID}} \leftarrow \mathsf{GenSK}(\mathsf{pp}, \mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}, \mathtt{ID})$
$\mathtt{SKList} \leftarrow \mathtt{SKList} \cup \{(\mathtt{ID}, \mathsf{sk}_{\mathtt{ID}})\}$
for $\mathtt{T} \in [\mathtt{T}_{\mathtt{cu}}]$
$\mathsf{ku}_{\mathtt{ID},\mathtt{T}} \leftarrow \mathsf{KeyUp}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}, \mathsf{RL}_{\mathtt{ID},\mathtt{T}} = \emptyset, \mathtt{T})$
return $(\mathsf{ku}_{\mathtt{ID},\mathtt{T}})_{\mathtt{T} \in [\mathtt{T}_{\mathtt{cu}}]}$

Secret Key Reveal Query on $\mathtt{ID}$:
if $\mathtt{T}_{\mathtt{cu}} \geq \mathtt{T}^\star \wedge \mathtt{ID} \in \mathsf{prefix}(\mathtt{ID}^\star) \wedge \mathtt{ID} \notin \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}^\star}$
return $\perp$
else
$\mathtt{RevList} \leftarrow \mathtt{RevList} \cup \{\mathtt{ID}\}$
return $\mathsf{sk}_{\mathtt{ID}} \in \mathtt{SKList}$

Revoke & Key Update Query on $\mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}}$:
if $\mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}-1} \nsubseteq \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}} \vee (\exists \mathtt{ID} \in \mathtt{SKList}, \mathsf{pa}(\mathtt{ID}) \in \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}} \wedge \mathtt{ID} \notin \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}}) \vee$
$(\exists \mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star), \mathtt{T}_{\mathtt{cu}} = \mathtt{T}^\star - 1 \wedge \mathtt{ID} \in \mathtt{RevList} \setminus \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}})$
return $\perp$
else
$\mathtt{T}_{\mathtt{cu}} \leftarrow \mathtt{T}_{\mathtt{cu}} + 1$
for $\mathtt{ID} \in (\mathtt{SKList} \cap \mathcal{I}^{\leq L-1}) \setminus \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}-1}$ in the breadth-first order on $|\mathtt{ID}|$
$\mathsf{RL}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}} \leftarrow \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}-1} \cap \mathcal{I}_{\mathtt{ID}}$
$\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}} \leftarrow \mathsf{KeyUp}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}_{\mathtt{cu}}}, \mathsf{RL}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}}, \mathtt{T}_{\mathtt{cu}})$
return $(\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}})_{\mathtt{ID} \in (\mathtt{SKList} \cap \mathcal{I}^{\leq L-1}) \setminus \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}-1}}$

Decryption Key Reveal Query on $(\mathtt{ID}, \mathtt{T})$:
if $\mathtt{T} > \mathtt{T}_{\mathtt{cu}} \vee \mathtt{ID} \in \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} \vee (\mathtt{ID}, \mathtt{T}) = (\mathtt{ID}^\star, \mathtt{T}^\star)$
return $\perp$
else
$\mathsf{dk}_{\mathtt{ID},\mathtt{T}} \leftarrow \mathsf{GenDK}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}, (\mathtt{ID}, \mathtt{T}))$
$\mathtt{RevList} \leftarrow \mathtt{RevList} \cup \{(\mathtt{ID}, \mathtt{T})\}$
return $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$

Challenge Query on $(\mathtt{ID}^\star, \mathtt{T}^\star, \mathsf{M}_0^\star, \mathsf{M}_1^\star)$:
if $(\exists \mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star), \mathtt{ID} \in \mathtt{RevList} \setminus \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}^\star}) \vee (\mathtt{ID}^\star, \mathtt{T}^\star) \in \mathtt{RevList}$
return $\perp$
else
$\mathsf{coin} \leftarrow_R \{0, 1\}$
$\mathsf{ct}^\star \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathtt{ID}^\star, \mathtt{T}^\star, \mathsf{M}_{\mathsf{coin}}^\star)$
return $\mathsf{ct}^\star$

Fig. 2: Security game of $\mathsf{RHIBE}(L)$

Binary Tree. Let $\mathtt{BT}$ denote a binary tree with 2^D leaves. Each leaf node η has a label as an integer in $[0, 2^D-1]$ as illustrated in Figure 3, e.g., the leftmost and rightmost leaf nodes have labels 0 and 2^D-1, respectively. Since $\mathtt{BT}$ has $2^{D+1}-1$ nodes, we assume that each node θ (including the leaf nodes) is expressed as a $(D+1)$-bit binary string and let $\mathcal{N} := \{0,1\}^{D+1}$ denote a node space. For all nodes $\theta \in \mathcal{N}$, let 0 and 1 denote a path from θ to its left and right child node, respectively. In this section, we simply use a bit string which denotes a path from the root to specify each node θ as illustrated in Figure 3. As a special case, let ε denote a root node. For every node θ, $\mathsf{pa}(\theta)$ denote a direct ancestor. Let $\mathsf{Path}(\eta)$ denote all nodes θ in a path from the root to the leaf η. Please keep in mind that all the descriptions are public information.

CS. Let $H : \mathcal{I} \to \mathcal{N}$ be a collision resistant hash function. CS consists of three *deterministic* algorithms $(\mathsf{Assign}, \mathsf{Cover}, \mathsf{Match})$:

$\mathsf{Assign}(\mathtt{ID}) \to \mathsf{PS}(\mathtt{ID})$: The *assign* algorithm takes a hierarchical identity $\mathtt{ID} = (\mathtt{id}_1, \ldots, \mathtt{id}_{|\mathtt{ID}|})$ as input, and outputs a private set $\mathsf{PS}(\mathtt{ID}) := \mathsf{Path}(H(\mathtt{ID}))$.[5]

$\mathsf{Cover}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}) \to \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$: The *cover* algorithm takes a set of hierarchical identities $\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} \subseteq \mathcal{I}_{\mathsf{pa}(\mathtt{ID})}$ as input, and outputs a covering set $\mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}) := \{\theta \mid \theta \notin \bigcup_{\mathtt{ID}\in\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}} \mathsf{Path}(H(\mathtt{ID})) \wedge \mathsf{pa}(\theta) \in \bigcup_{\mathtt{ID}\in\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}} \mathsf{Path}(H(\mathtt{ID}))\}$.

$\mathsf{Match}(\mathsf{PS}(\mathtt{ID}), \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})) \to \tilde{\theta}$ or $\perp$: The *match* algorithm takes an $\mathtt{ID}$'s private set $\mathsf{PS}(\mathtt{ID})$ and covering set $\mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$ as input, and outputs $\tilde{\theta} \in \mathsf{PS}(\mathtt{ID}) \cap \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$ if such a node exists and $\perp$ otherwise.

By definition, $\mathsf{PS}(\mathtt{ID})$ consists of $D+1$ nodes and let $\mathsf{PS}(\mathtt{ID}) = \{\theta_0, \theta_1, \ldots, \theta_D\}$. Let $R_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ denote the number of nodes in $\mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$ and let $\mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}) = \{\bar{\theta}_1, \ldots, \bar{\theta}_{R_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}}\}$. The CS satisfies the following three properties:

Correctness: The Match algorithm does not output $\perp$ if $\mathtt{ID} \notin \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ holds.

Security: The Match algorithm outputs $\perp$ if $\mathtt{ID} \in \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ holds.

Scalability: It holds that $R_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} = O(|\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}| \log(2^D/|\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}|))$.

Example. We use Figures 3 and 4 to illustrate the examples of the CS. As illustrated in Figure 3, if $H(\mathtt{ID}) = 6$, $\mathsf{PS}(\mathtt{ID}) = \{\varepsilon, 1, 11, 110\}$. As illustrated in Figure 4, if $H(\mathtt{ID}) = 2$ and $H(\mathtt{ID}) = 3$ for $\mathtt{ID} \in \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$, $\mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}) = \{00, 1\}$. Here, we use Figure 4 to show that CS satisfies the correctness and security. See [33] for more information. If $H(\mathtt{ID}) = 6$ for $\mathtt{ID} \notin \mathsf{RL}$, $\mathsf{PS}(\mathtt{ID})$ and $\mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$ share the common node 1. We note that $\mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}) = \{\varepsilon\}$ if $\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} = \emptyset$. If $H(\mathtt{ID}) = 2$ and $H(\mathtt{ID}) = 3$ for $\mathtt{ID} \in \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$, $\mathsf{PS}(\mathtt{ID}) = \{\varepsilon, 0, 01, 010\}$ and $\mathsf{PS}(\mathtt{ID}) = \{\varepsilon, 0, 01, 011\}$ do not share a common node with $\mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$.

[5] Even when $\mathtt{ID} \neq \mathtt{ID}'$, it holds that $\mathsf{PS}(\mathtt{ID}) = \mathsf{PS}(\mathtt{ID}')$ if $H(\mathtt{id}_{|\mathtt{ID}|}) = H(\mathtt{id}'_{|\mathtt{ID}'|})$ holds. Furthermore, since $H(\cdot)$ is a collision resistant hash function, we assume that $\mathsf{PS}(\mathtt{ID}) = \mathsf{PS}(\mathtt{ID}')$ holds only when $\mathtt{id}_{|\mathtt{ID}|} = \mathtt{id}'_{|\mathtt{ID}'|}$ holds.

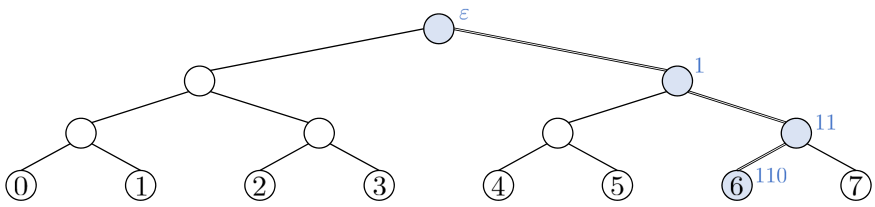

Fig. 3: Example of $\mathsf{PS}(\mathtt{ID})$ output by Assign if $H(\mathtt{ID}) = 6$

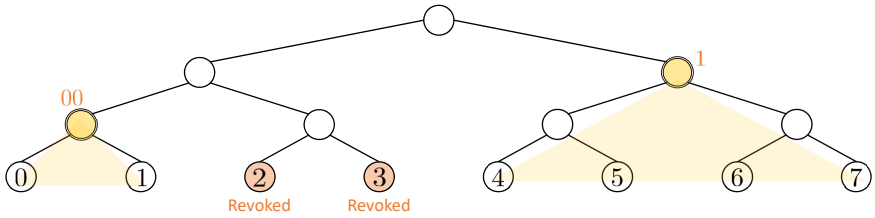

Fig. 4: Example of $\mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$ output by Cover if $H(\mathtt{ID}) = 2$ and $H(\mathtt{ID}) = 3$ for $\mathtt{ID} \in \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$

3 Basic Scheme: Generic Construction from HIBE

In this section, we propose our generic construction.

Parameters and Construction. Let the element identity space and time period space of RHIBE be $\mathcal{I} = \{0,1\}^{\kappa_{\mathcal{I}}} \| \{0\}$ and $\mathcal{T} = \{0,1\}^{\kappa_{\mathcal{T}}} \| \{1\}$ so that $\mathcal{I} \cap \mathcal{T} = \emptyset$ and $\mathcal{I} \cap \{0,1\}^{\lceil \log(L+1) \rceil} \| \mathcal{N} \| \mathcal{T} = \emptyset$. Let the element identity space $\mathsf{HIBE}.\mathcal{I}$ of HIBE be $\{0,1\}^{\kappa_{\mathsf{HIBE}}}$, where $1 + \max\{\kappa_{\mathcal{I}}, \lceil \log(L+1) \rceil + (D+1) + \kappa_{\mathcal{T}}\} \leq \kappa_{\mathsf{HIBE}}$ so that $\mathcal{I} \cup \mathcal{T} \cup \{0,1\}^{\lceil \log(L+1) \rceil} \| \mathcal{N} \| \mathcal{T} \subseteq \mathsf{HIBE}.\mathcal{I}$. We use $\mathsf{HIBE}(L+1)$ for constructing $\mathsf{RHIBE}(L)$ as described in Figure 5.

Overview. The spirit of our generic construction is similar to lattice-based RHIBE schemes [22,40]. The RHIBE ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$ consists of level-ℓ ciphertexts $\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell}$ that are encryptions of M_ℓ for $\ell \in [|\mathtt{ID}|] \cup \{L+1\}$, where $\mathsf{M} = \bigoplus_{\ell \in [\mathtt{ID}] \cup \{L+1\}} \mathsf{M}_\ell$. In other words, the decryption succeeds only when non-revoked user $\mathtt{ID}$ can decrypt all level-ℓ ciphertexts and recover M_ℓ for $\ell \in [|\mathtt{ID}|] \cup \{L+1\}$. The level-$\ell$ ciphertexts $\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}|]$ consist of $D+1$ HIBE ciphertexts $\mathsf{HIBE}.\mathsf{ct}_{(\mathtt{ID}_{[\ell-1]},\ell \| \theta_{\ell,d} \| \mathtt{T})}$ for $\theta_{\ell,d} \in \mathsf{PS}(\mathtt{ID}_{[\ell]})$ whereas the level-$(L+1)$ ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1}$ is a single HIBE ciphertext $\mathsf{HIBE}.\mathsf{ct}_{(\mathtt{ID},\mathtt{T})}$. To satisfy the correctness of Dec, $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ for non-revoked $\mathtt{ID}$ consists of level-ℓ decryption keys $\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}|] \cup \{L+1\}$ so that $\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}$ can decrypt one of the HIBE ciphertexts in $\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}|]$ and $\mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1}$ can decrypt the HIBE ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1}$. For this purpose, the level-ℓ decryption keys $\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}|]$

$\mathsf{Setup}(1^\lambda, L)$:
- $(\mathsf{HIBE.pp}, \mathsf{HIBE.msk}) \leftarrow \mathsf{HIBE.Setup}(1^\lambda, L+1)$
- return $\mathsf{pp} = \mathsf{HIBE.pp}$, $\mathsf{sk_{kgc}} = \mathsf{HIBE.msk}$

$\mathsf{Enc}(\mathsf{pp}, \mathtt{ID}, \mathtt{T}, \mathsf{M})$:
- $(\mathsf{M}_1, \ldots, \mathsf{M}_{|\mathtt{ID}|}) \leftarrow_R \mathcal{M}^{|\mathtt{ID}|}$, $\mathsf{M}_{L+1} = \mathsf{M} \bigoplus_{\ell \in [|\mathtt{ID}|]} \mathsf{M}_\ell$
- for $\ell \in [|\mathtt{ID}|]$
  - $\mathsf{PS}(\mathtt{ID}_{[\ell]}) = (\theta_{\ell,0}, \theta_{\ell,1}, \ldots, \theta_{\ell,D}) \leftarrow \mathsf{Assign}(\mathtt{ID}_{[\ell]})$
  - for $d \in [0, D]$
    - $\mathsf{HIBE.ct}_{(\mathtt{ID}_{[\ell-1]}, \ell \| \theta_{\ell,d} \| \mathtt{T})} \leftarrow \mathsf{HIBE.Enc}(\mathsf{HIBE.pp}, (\mathtt{ID}_{[\ell-1]}, \ell \| \theta_{\ell,d} \| \mathtt{T}), \mathsf{M}_\ell)$
  - $\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell} \coloneqq (\mathsf{HIBE.ct}_{(\mathtt{ID}_{[\ell-1]}, \ell \| \theta_{\ell,d} \| \mathtt{T})})_{d \in [0,D]}$
- $\mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1} \coloneqq \mathsf{HIBE.ct}_{(\mathtt{ID},\mathtt{T})} \leftarrow \mathsf{HIBE.Enc}(\mathsf{HIBE.pp}, (\mathtt{ID}, \mathtt{T}), \mathsf{M}_{L+1})$
- return $\mathsf{ct}_{\mathtt{ID},\mathtt{T}} \coloneqq (\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|] \cup \{L+1\}}$

$\mathsf{GenSK}(\mathsf{pp}, \mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}, \mathtt{ID})$:
- $\mathsf{HIBE.sk}_{\mathtt{ID}} \leftarrow \mathsf{HIBE.KeyGen}(\mathsf{HIBE.pp}, \mathsf{HIBE.sk}_{\mathsf{pa}(\mathtt{ID})}, \mathtt{ID})$
- return $\mathsf{sk}_{\mathtt{ID}} \coloneqq \mathsf{HIBE.sk}_{\mathtt{ID}}$

$\mathsf{KeyUp}(\mathsf{pp}, \mathtt{T}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{RL}_{\mathtt{ID},\mathtt{T}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$:
- $(\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|] \cup \{L+1\}} \leftarrow \mathsf{GenDK}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$
- $\mathsf{CoS}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}}) = (\bar{\theta}_{|\mathtt{ID}|+1,1}, \bar{\theta}_{|\mathtt{ID}|+1,2}, \ldots, \bar{\theta}_{|\mathtt{ID}|+1,R_{\mathtt{ID},\mathtt{T}}}) \leftarrow \mathsf{Cover}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}})$
  - for $\bar{d} \in [R_{\mathtt{ID},\mathtt{T}}]$
    - $\mathsf{HIBE.sk}_{(\mathtt{ID},(|\mathtt{ID}|+1) \| \bar{\theta}_{|\mathtt{ID}|+1,\bar{d}} \| \mathtt{T})} \leftarrow$
      - $\mathsf{HIBE.KeyGen}(\mathsf{HIBE.pp}, \mathsf{HIBE.sk}_{\mathtt{ID}}, (\mathtt{ID}, (|\mathtt{ID}|+1) \| \bar{\theta}_{|\mathtt{ID}|+1,\bar{d}} \| \mathtt{T}))$
- return $\mathsf{ku}_{\mathtt{ID},\mathtt{T}} \coloneqq ((\tilde{\theta}_\ell, \mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|]}, (\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}, \mathsf{HIBE.sk}_{(\mathtt{ID},(|\mathtt{ID}|+1) \| \bar{\theta}_{|\mathtt{ID}|+1,\bar{d}} \| \mathtt{T})})_{\bar{d} \in [R_{\mathtt{ID},\mathtt{T}}]})$

$\mathsf{GenDK}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$:
- if $\perp \leftarrow \mathsf{Match}(\mathsf{PS}(\mathtt{ID}_{[|\mathtt{ID}|]}), \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}))$
  - return $\perp$
- else
  - $\tilde{\theta}_{|\mathtt{ID}|} \leftarrow \mathsf{Match}(\mathsf{PS}(\mathtt{ID}_{[|\mathtt{ID}|]}), \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}))$
  - $\mathsf{dk}_{\mathtt{ID},\mathtt{T},|\mathtt{ID}|} \coloneqq \mathsf{HIBE.sk}_{(\mathsf{pa}(\mathtt{ID}),|\mathtt{ID}| \| \tilde{\theta}_{|\mathtt{ID}|} \| \mathtt{T})}$
  - $\mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1} \coloneqq \mathsf{HIBE.sk}_{(\mathtt{ID},\mathtt{T})} \leftarrow \mathsf{HIBE.KeyGen}(\mathsf{HIBE.pp}, \mathsf{HIBE.sk}_{\mathtt{ID}}, (\mathtt{ID}, \mathtt{T}))$
  - return $\mathsf{dk}_{\mathtt{ID},\mathtt{T}} \coloneqq ((\tilde{\theta}_\ell, \mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|]}, \mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1})$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T}}, \mathsf{ct}_{\mathtt{ID},\mathtt{T}})$:
- for $\ell \in [|\mathtt{ID}|]$
  - $\mathsf{M}_\ell \leftarrow \mathsf{HIBE.Dec}(\mathsf{HIBE.pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}, \mathsf{HIBE.ct}_{(\mathtt{ID}_{[\ell-1]}, \ell \| \tilde{\theta}_\ell \| \mathtt{T})})$
- $\mathsf{M}_{L+1} \leftarrow \mathsf{HIBE.Dec}(\mathsf{HIBE.pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1}, \mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1})$
- return $\mathsf{M} = \bigoplus_{\ell \in [\mathtt{ID}] \cup \{L+1\}} \mathsf{M}_\ell$

Fig. 5: $\mathsf{RHIBE}(L)$ scheme from $\mathsf{HIBE}(L+1)$

are HIBE secret keys $\mathsf{HIBE.sk}_{(\mathtt{ID}_{[\ell-1]}, \ell \| \tilde{\theta}_\ell \| \mathtt{T})}$ for some $\tilde{\theta}_\ell \in \mathsf{PS}(\mathtt{ID}_{[\ell]})$ whereas the level-$(L+1)$ decryption key $\mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1}$ is an HIBE secret key $\mathsf{HIBE.sk}_{(\mathtt{ID},\mathtt{T})}$.

We set the RHIBE secret key $\mathsf{sk}_{\mathtt{ID}}$ as an HIBE secret key $\mathsf{HIBE.sk}_{\mathtt{ID}}$; thus, $\mathtt{ID}$ can create the level-$(L+1)$ decryption key $\mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1}$. Since $\mathtt{ID}$'s level-ℓ ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}|-1]$ depends only on $(\mathtt{ID}_{[\ell]}, \mathtt{T})$, we can set $\mathsf{dk}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\ell} = \mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}| - 1]$. The parent user $\mathsf{pa}(\mathtt{ID})$'s key update $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ consists of $\mathtt{ID}$'s level-ℓ decryption keys $\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}| - 1]$ and $R_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ HIBE secret keys

$\mathsf{HIBE.sk}_{(\mathsf{pa(ID)},|\mathtt{ID}||\bar{\theta}_{|\mathtt{ID}|,\bar{d}}\|\mathtt{T})}$ for $\bar{\theta}_{|\mathtt{ID}|,\bar{d}} \in \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa(ID)},\mathtt{T}})$ so that the HIBE secret keys are level-$|\mathtt{ID}|$ decryption keys for all non-revoked users $\mathtt{ID}$. Indeed, if $\mathtt{ID}$ is not revoked, the correctness of the CS ensures that there is a node $\tilde{\theta}_{|\mathtt{ID}|} \in \mathsf{PS}(\mathtt{ID}_{[|\mathtt{ID}|]}) \cap \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa(ID)},\mathtt{T}})$; thus, we set $\mathtt{ID}$'s level-$|\mathtt{ID}|$ decryption key $\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}$ as an HIBE secret key $\mathsf{HIBE.sk}_{(\mathsf{pa(ID)},|\mathtt{ID}||\tilde{\theta}_{|\mathtt{ID}|}\|\mathtt{T})} \in \mathsf{ku}_{\mathsf{pa(ID)},\mathtt{T}}$.

Correctness. Since $\mathsf{pa}(\mathtt{ID}) \in \mathsf{prefix}(\mathtt{ID})$ and $\mathtt{ID} \in \mathsf{prefix}((\mathtt{ID}, (|\mathtt{ID}|+1)\|\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}\|\mathtt{T}))$, the correctness of HIBE ensures that the GenSK and KeyUp algorithms correctly output $\mathsf{sk}_{\mathtt{ID}}$ and $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$. If $\mathtt{ID} \notin \mathsf{RL}_{\mathsf{pa(ID)},\mathtt{T}}$, the correctness of the CS ensures that the GenDK algorithm does not output $\perp$. Moreover, since $\mathtt{ID} \in \mathsf{prefix}((\mathtt{ID},\mathtt{T}))$, the correctness of HIBE ensures that the GenDK algorithm correctly outputs $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ if $\mathtt{ID} \notin \mathsf{RL}_{\mathsf{pa(ID)},\mathtt{T}}$. Since $\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell} = \mathsf{HIBE.sk}_{(\mathtt{ID}_{[\ell-1]},\ell\|\tilde{\theta}_\ell\|\mathtt{T})}$ for $\ell \in [|\mathtt{ID}|]$, $\mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1} = \mathsf{HIBE.sk}_{(\mathtt{ID},\mathtt{T})}$, and $\mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1} = \mathsf{HIBE.ct}_{(\mathtt{ID},\mathtt{T})}$, the correctness of HIBE ensures that the Dec algorithm correctly computes M_ℓ for $\ell \in [|\mathtt{ID}|] \cup \{L+1\}$. Thus, the Dec algorithm correctly outputs M.

Theorem 1. *If the underlying* HIBE *scheme satisfy adaptive (resp. selective) security, then the* RHIBE *scheme in Figure 5 satisfies adaptive (resp. selective) security.*

To prove Theorem 1, we divide an RHIBE adversary $\mathcal{A}$'s attack strategy into the following two types:

Type-I : $\mathcal{A}$ is called Type-I iff $\mathcal{A}$ makes a secret key *reveal* query on some $\mathtt{ID} \in \mathsf{prefix}(\mathtt{ID}^\star)$.

Type-II : $\mathcal{A}$ is called Type-II iff $\mathcal{A}$ does not make secret key *reveal* queries on any $\mathtt{ID} \in \mathsf{prefix}(\mathtt{ID}^\star)$.

Let $\ell^\star \in [|\mathtt{ID}^\star|]$ denote an integer such that $\mathcal{A}$ of the Type-I strategy makes a secret key *reveal* query on $\mathtt{ID}^\star_{[\ell^\star]}$, while $\mathcal{A}$ does not make secret key *reveal* queries on any $\mathtt{ID} \in \mathsf{prefix}(\mathtt{ID}^\star_{[\ell^\star]})$. In this case, $\mathcal{A}$ of the Type-I strategy that has $\mathsf{sk}_{\mathtt{ID}^\star_{[\ell^\star]}} = \mathsf{HIBE.sk}_{\mathtt{ID}^\star_{[\ell^\star]}}$ can create all level-ℓ decryption keys $\mathsf{dk}_{\mathtt{ID}^\star,\mathtt{T}^\star,\ell}$ for $\ell \in [\ell^\star+1, |\mathtt{ID}^\star|] \cup \{L+1\}$. Although the security definition of RHIBE ensures that $\mathtt{ID}^\star_{[\ell^\star]}$ is revoked by the challenge time period $\mathtt{T}^\star$, $\mathsf{pa}(\mathtt{ID}^\star_{[\ell^\star]}) = \mathtt{ID}^\star_{[\ell^\star-1]}$ may not be revoked. Then, $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell^\star]}),\mathtt{T}^\star}$ contains $\mathtt{ID}^\star$'s level-ℓ decryption keys $\mathsf{dk}_{\mathtt{ID}^\star,\mathtt{T}^\star,\ell}$ for $\ell \in [\ell^\star-1]$. Therefore, $\mathcal{A}$ can decrypt all level-ℓ challenge ciphertexts $\mathsf{ct}_{\mathtt{ID}^\star,\mathtt{T}^\star,\ell}$ for $\ell \in ([|\mathtt{ID}^\star|] \cup \{L+1\}) \setminus \{\ell^\star\}$. Nevertheless, we can prove Theorem 1 against $\mathcal{A}$ of the Type-I strategy since it does not have a way for decrypting the level-$\ell^\star$ challenge ciphertext $\mathsf{ct}_{\mathtt{ID}^\star,\mathtt{T}^\star,\ell^\star}$ without breaking the security of HIBE.

Since $\mathcal{A}$ of the Type-II strategy does not make secret key *reveal* queries on any $\mathtt{ID} \in \mathsf{prefix}(\mathtt{ID}^\star)$, $\mathtt{ID}^\star$ may not be revoked by the challenge time period $\mathtt{T}^\star$. Then, $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}^\star),\mathtt{T}^\star}$ contains $\mathtt{ID}^\star$'s level-ℓ decryption keys $\mathsf{dk}_{\mathtt{ID}^\star,\mathtt{T}^\star,\ell}$ for $\ell \in [|\mathtt{ID}^\star|]$. Therefore, $\mathcal{A}$ can decrypt all level-ℓ challenge ciphertexts $\mathsf{ct}_{\mathtt{ID}^\star,\mathtt{T}^\star,\ell}$ for $\ell \in [|\mathtt{ID}^\star|]$. Nevertheless, we can prove Theorem 1 against $\mathcal{A}$ of the Type-II strategy since it does not have a way for decrypting the level-$(L+1)$ challenge ciphertext $\mathsf{ct}_{\mathtt{ID}^\star,\mathtt{T}^\star,L+1}$ without breaking the security of HIBE.

Proof of Theorem 1. At first, we show a proof against $\mathcal{A}$ of the Type-I strategy. For this purpose, we show that there is a reduction algorithm $\mathcal{B}$ for breaking the

security of HIBE if there exists any PPT adversary $\mathcal{A}$ of RHIBE. $\mathcal{B}$ answers all $\mathcal{A}$'s key queries by making secret key queries on the corresponding hierarchical identities to the HIBE challenger $\mathcal{C}$. Let $\ell^\star \in [|\mathtt{ID}^\star|]$ denote an integer introduced above. To answer $\mathcal{A}$'s key queries, $\mathcal{B}$ makes HIBE secret key queries on hierarchical identities $\mathsf{HIBE}.\mathtt{ID} \in \mathsf{HIBE}.\mathcal{I}^{\leq L+1}$ such that $\mathsf{HIBE}.\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star_{[\ell^\star-1]}, \ell^\star \| \theta_{\ell^\star,d} \| \mathtt{T}^\star)$ for all $\theta_{\ell^\star,d} \in \mathsf{PS}(\mathtt{ID}^\star_{[\ell^\star]})$. Upon $\mathcal{A}$'s challenge query on $(\mathtt{ID}^\star, \mathtt{T}^\star, \mathsf{M}^\star_0, \mathsf{M}^\star_1)$, $\mathcal{B}$ guesses the value $\ell^\star \leftarrow_R [|\mathtt{ID}^\star|]$ with the success probability at least $1/L$. Then, $\mathcal{B}$ samples $(\mathsf{M}_\ell)_{\ell \in ([|\mathtt{ID}^\star|] \cup \{L+1\}) \setminus \{\ell^\star\}} \leftarrow_R \mathcal{M}^{|\mathtt{ID}^\star|}$, makes multiple HIBE challenge queries on

$$\left((\mathtt{ID}^\star_{[\ell^\star-1]}, \ell^\star \| \theta_{\ell^\star,d} \| \mathtt{T}^\star)_{\theta_{\ell^\star,d} \in \mathsf{PS}(\mathtt{id}^\star_{\ell^\star})}, \begin{matrix} \mathsf{M}^\star_0 \bigoplus_{\ell \in ([|\mathtt{ID}^\star|] \cup \{L+1\}) \setminus \{\ell^\star\}} \mathsf{M}_\ell, \\ \mathsf{M}^\star_1 \bigoplus_{\ell \in ([|\mathtt{ID}^\star|] \cup \{L+1\}) \setminus \{\ell^\star\}} \mathsf{M}_\ell \end{matrix} \right)$$

to $\mathcal{C}$, receives $(\mathsf{HIBE}.\mathsf{ct}^\star_{\theta_{\ell^\star,d}})_{\theta_{\ell^\star,d} \in \mathsf{PS}(\mathtt{id}^\star_{\ell^\star})}$, and sets $\mathsf{ct}^\star_{\mathtt{ID}^\star,\mathtt{T}^\star,\ell^\star} = (\mathsf{HIBE}.\mathsf{ct}^\star_{\theta_{\ell^\star,d}})_{\theta_{\ell^\star,d} \in \mathsf{PS}(\mathtt{id}^\star_{\ell^\star})}$. $\mathcal{B}$ computes $\mathsf{HIBE}.\mathsf{ct}^\star_\ell$ for $\ell \in [|\mathtt{ID}^\star|] \setminus \{\ell^\star\}$ and $\mathsf{HIBE}.\mathsf{ct}^\star$ by itself as the encryptions of M_ℓ and M_{L+1}, respectively. Then, $\mathcal{B}$ returns the RHIBE challenge ciphertext $\mathsf{ct}^\star = (\mathsf{ct}^\star_{\mathtt{ID}^\star,\mathtt{T}^\star,\ell})_{\ell \in [|\mathtt{ID}^\star|] \cup \{L+1\}}$ to $\mathcal{A}$. Let $\mathsf{M}_{\ell^\star} = \mathsf{M}^\star_{\mathsf{coin}} \bigoplus_{\ell \in ([|\mathtt{ID}^\star|] \cup \{L+1\}) \setminus \{\ell^\star\}} \mathsf{M}_\ell$, where the level-$\ell^\star$ challenge ciphertext $\mathsf{ct}^\star_{\mathtt{ID}^\star,\mathtt{T}^\star,\ell^\star}$ is the encryption of $\mathsf{M}_{\ell^\star}$. Given $(\mathsf{M}_\ell)_{\ell \in [|\mathtt{ID}^\star|]}$, they are distributed in $\mathcal{M}^{|\mathtt{ID}^\star|}$ uniformly at random. Moreover, it holds that $\mathsf{M}^\star_{\mathsf{coin}} \bigoplus_{\ell \in [|\mathtt{ID}^\star|]} \mathsf{M}_\ell = \mathsf{M}^\star_{\mathsf{coin}} \oplus \mathsf{M}_{\ell^\star} \bigoplus_{\ell \in [|\mathtt{ID}^\star|] \setminus \{\ell^\star\}} \mathsf{M}_\ell = \mathsf{M}^\star_{\mathsf{coin}} \oplus \mathsf{M}^\star_{\mathsf{coin}} \bigoplus_{\ell \in ([|\mathtt{ID}^\star| \cup \{L+1\}]) \setminus \{\ell^\star\}} \mathsf{M}_\ell \bigoplus_{\ell \in [|\mathtt{ID}^\star|] \setminus \{\ell^\star\}} \mathsf{M}_\ell = \mathsf{M}_{L+1}$; therefore, the RHIBE challenge ciphertext $\mathsf{ct}^\star$ is properly distributed.

Hereafter, we check that $\mathcal{B}$ made HIBE secret key queries on $\mathsf{HIBE}.\mathtt{ID}$ only when $\mathsf{HIBE}.\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star_{[\ell^\star-1]}, \ell^\star \| \theta_{\ell^\star,d} \| \mathtt{T}^\star)$ for all $\theta_{\ell^\star,d} \in \mathsf{PS}(\mathtt{id}^\star_{\ell^\star})$ if the guess of $\ell^\star$ is correct by using the conditions $\mathcal{I} \cap \mathcal{T} = \emptyset$ and $\mathcal{I} \cap \{0,1\}^{\lceil \log(L+1) \rceil} \| \mathcal{N} \| \mathcal{T} = \emptyset$.

Secret Key Generation Query. Upon $\mathcal{A}$'s queries on $\mathtt{ID}$, $\mathcal{B}$ has to answer $(\mathsf{ku}_{\mathtt{ID},\mathtt{T}})_{\mathtt{T} \in [\mathtt{T}_{\mathtt{cu}}]}$ such that $\mathsf{RL}_{\mathtt{ID},\mathtt{T}} = \emptyset$. For this purpose, $\mathcal{B}$ made HIBE secret key queries on $(\mathtt{ID}, |\mathtt{ID}|+1 \| \varepsilon \| \mathtt{T})$ for $\mathtt{T} \in [\mathtt{T}_{\mathtt{cu}}]$, where ε denotes the root node. Observe that $(\mathtt{ID}, |\mathtt{ID}|+1 \| \varepsilon \| \mathtt{T}) \in \mathsf{prefix}^+(\mathtt{ID}^\star_{[\ell^\star-1]}, \ell^\star \| \theta_{\ell^\star,d} \| \mathtt{T}^\star)$ holds only when $\mathtt{ID} = \mathtt{ID}^\star_{[\ell^\star-1]} \wedge \varepsilon = \theta_{\ell^\star,d} \wedge \mathtt{T} = \mathtt{T}^\star$. Here, we use the fact that $\mathtt{id} \neq \ell \| \theta \| \mathtt{T}$ holds for all $\mathtt{id} \in \mathcal{I}, \ell \in [L], \theta \in \mathcal{N}$, and $\mathtt{T} \in \mathcal{T}$ since $\mathcal{I} \cap [L] \| \mathcal{N} \| \mathcal{T} = \emptyset$ holds. Since $\mathcal{A}$ made secret key *reveal* query on $\mathtt{ID}^\star_{[\ell^\star]}$, $\mathcal{A}$ revoked $\mathtt{ID}^\star_{[\ell^\star]}$ before $\mathtt{T}^\star$. Moreover, $\mathcal{A}$ made secret key *generation* query on $\mathtt{ID}^\star_{[\ell^\star-1]}$ before $\mathtt{T}^\star$. Thus, it holds that $\mathtt{ID} = \mathtt{ID}^\star_{[\ell^\star-1]} \Rightarrow \varepsilon \neq \theta_{\ell^\star,d} \wedge \mathtt{T} \neq \mathtt{T}^\star$. Therefore, $(\mathtt{ID}, |\mathtt{ID}|+1 \| \varepsilon \| \mathtt{T}) \notin \mathsf{prefix}^+(\mathtt{ID}^\star_{[\ell^\star-1]}, \ell^\star \| \theta_{\ell^\star,d} \| \mathtt{T}^\star)$ holds for all $\mathtt{T} \in [\mathtt{T}_{\mathtt{cu}}]$.

Secret Key Reveal Query. Upon $\mathcal{A}$'s queries on $\mathtt{ID}$, $\mathcal{B}$ has to answer $\mathsf{sk}_{\mathtt{ID}}$. For this purpose, $\mathcal{B}$ made HIBE secret key queries on $\mathtt{ID}$. Observe that $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star_{[\ell^\star-1]}, \ell^\star \| \theta_{\ell^\star,d} \| \mathtt{T}^\star)$ holds only when $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star_{[\ell^\star-1]})$. Due to the definition of $\ell^\star$, it holds that $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star_{[\ell^\star-1]})$.

Revoke & Key Update Query. Upon $\mathcal{A}$'s queries on $\mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}}$, $\mathcal{B}$ increments the time period $\mathtt{T}_{\mathtt{cu}} \leftarrow \mathtt{T}_{\mathtt{cu}} + 1$ and has to answer $(\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}})_{\mathtt{ID} \in (\mathtt{SKList} \cap \mathcal{I}^{\leq L-1}) \setminus \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}-1}}$ such that $\mathsf{RL}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}} = \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}-1} \cap \mathcal{I}_{\mathtt{ID}}$. For this purpose, $\mathcal{B}$ made HIBE secret key

queries on $(\mathtt{ID}, (|\mathtt{ID}|+1)\|\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}, \mathtt{T}_{\mathsf{cu}})$ for all $\mathtt{ID} \in (\mathtt{SKList} \cap \mathcal{I}^{\leq L-1}) \setminus \mathsf{RL}_{\mathtt{T}_{\mathsf{cu}}-1}$ and $\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}} \in \mathsf{CoS}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}_{\mathsf{cu}}})$. Observe that $(\mathtt{ID}, (|\mathtt{ID}|+1)\|\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}, \mathtt{T}_{\mathsf{cu}}) \in \mathsf{prefix}^{+}(\mathtt{ID}^{\star}_{[\ell^{\star}-1]}, \ell^{\star}\|\theta_{\ell^{\star},d}\|\mathtt{T}^{\star})$ only when $\mathtt{ID} = \mathtt{ID}^{\star}_{[\ell^{\star}-1]} \wedge \bar{\theta}_{|\mathtt{ID}|+1,\bar{d}} = \theta_{\ell^{\star},d} \wedge \mathtt{T} = \mathtt{T}^{\star}$. Since $\mathcal{A}$ made secret key *reveal* query on $\mathtt{ID}^{\star}_{[\ell^{\star}]}$, $\mathcal{A}$ revoked $\mathtt{ID}^{\star}_{[\ell^{\star}]}$ before $\mathtt{T}^{\star}$. Thus, the security of CS ensures that $\mathtt{ID} = \mathtt{ID}^{\star}_{[\ell^{\star}-1]} \wedge \mathtt{T} = \mathtt{T}^{\star} \Rightarrow \bar{\theta}_{|\mathtt{ID}|+1,\bar{d}} \neq \theta_{\ell^{\star},d}$ holds.

Decryption Key Reveal Query. Upon $\mathcal{A}$'s queries on $(\mathtt{ID}, \mathtt{T})$, $\mathcal{B}$ has to answer $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$. For this purpose, $\mathcal{B}$ made HIBE secret key queries on $((\mathtt{ID}_{[\ell-1]}, \ell\|\tilde{\theta}_{\ell}\|\mathtt{T}))_{\ell\in[|\mathtt{ID}|]}$ for some $\tilde{\theta}_{\ell} \in \mathsf{PS}(\mathtt{ID}_{[\ell]}) \cap \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}_{[\ell]}),\mathtt{T}})$ and $(\mathtt{ID}, \mathtt{T})$. At first, observe that $(\mathtt{ID}, \mathtt{T}) \in \mathsf{prefix}^{+}(\mathtt{ID}^{\star}_{[\ell^{\star}-1]}, \ell^{\star}\|\theta_{\ell^{\star},d}\|\mathtt{T}^{\star})$ never holds for all $\ell \in [|\mathtt{ID}|]$ even when $(\mathtt{ID}, \mathtt{T}) = (\mathtt{ID}^{\star}, \mathtt{T}^{\star})$. Next, observe that $(\mathtt{ID}_{[\ell-1]}, \ell\|\tilde{\theta}_{\ell}\|\mathtt{T}) \in \mathsf{prefix}^{+}(\mathtt{ID}^{\star}_{[\ell^{\star}-1]}, \ell^{\star}\|\theta_{\ell^{\star},d}\|\mathtt{T}^{\star})$ holds only when $\mathtt{ID}_{[\ell-1]} = \mathtt{ID}^{\star}_{[\ell^{\star}-1]} \wedge \tilde{\theta}_{\ell} = \theta_{\ell^{\star},d} \wedge \mathtt{T} = \mathtt{T}^{\star}$. As we observed so far, $\mathcal{A}$ revoked $\mathtt{ID}^{\star}_{[\ell^{\star}]}$ before $\mathtt{T}^{\star}$. Therefore, the security of CS ensures that $\mathtt{ID}_{[\ell^{\star}-1]} = \mathtt{ID}^{\star}_{[\ell^{\star}-1]} \wedge \mathtt{T} = \mathtt{T}^{\star} \Rightarrow \tilde{\theta}_{\ell} \neq \theta_{\ell^{\star},d}$.

Summarizing the analysis so far, $\mathcal{B}$ made HIBE secret key queries on $\mathsf{HIBE}.\mathtt{ID}$ only when $\mathsf{HIBE}.\mathtt{ID} \notin \mathsf{prefix}^{+}(\mathtt{ID}^{\star}_{[\ell^{\star}-1]}, \ell^{\star}\|\theta_{\ell^{\star},d}\|\mathtt{T}^{\star})$ for all $\theta_{\ell^{\star},d} \in \mathsf{PS}(\mathtt{id}^{\star}_{\ell^{\star}})$ if the guess of $\ell^{\star}$ is correct. Thus, we complete the proof against $\mathcal{A}$ of the Type-I strategy.

Next, we show a proof against $\mathcal{A}$ of the Type-II strategy in the same way as a proof against $\mathcal{A}$ of the Type-I strategy. The only exception of the reduction algorithm $\mathcal{B}$'s behavior is the way for creating the challenge ciphertext $\mathsf{ct}^{\star}$. Upon $\mathcal{A}$'s challenge query on $(\mathtt{ID}^{\star}, \mathtt{T}^{\star}, \mathsf{M}^{\star}_0, \mathsf{M}^{\star}_1)$, $\mathcal{B}$ samples $(\mathsf{M}_{\ell})_{\ell\in[|\mathtt{ID}^{\star}|]} \leftarrow_R \mathcal{M}^{|\mathtt{ID}^{\star}|}$, makes an HIBE challenge query on

$$\left((\mathtt{ID}^{\star}, \mathtt{T}^{\star}), \mathsf{M}^{\star}_0 \bigoplus_{\ell\in[|\mathtt{ID}^{\star}|]} \mathsf{M}_{\ell}, \mathsf{M}^{\star}_1 \bigoplus_{\ell\in[|\mathtt{ID}^{\star}|]} \mathsf{M}_{\ell}\right)$$

to $\mathcal{C}$, receives $\mathsf{HIBE}.\mathsf{ct}^{\star}_{(\mathtt{ID}^{\star},\mathtt{T}^{\star})}$, and sets $\mathsf{ct}^{\star}_{\mathtt{ID}^{\star},\mathtt{T}^{\star},L+1} = \mathsf{HIBE}.\mathsf{ct}^{\star}_{(\mathtt{ID}^{\star},\mathtt{T}^{\star})}$. $\mathcal{B}$ computes $\mathsf{ct}^{\star}_{\mathtt{ID}^{\star},\mathtt{T}^{\star},\ell}$ for $\ell \in [|\mathtt{ID}^{\star}|]$ by itself as the encryptions of M_{ℓ}, respectively. Then, $\mathcal{B}$ returns the RHIBE challenge ciphertext $\mathsf{ct}^{\star} = (\mathsf{ct}^{\star}_{\mathtt{ID}^{\star},\mathtt{T}^{\star},\ell})_{\ell\in[|\mathtt{ID}^{\star}|]\cup\{L+1\}}$ to $\mathcal{A}$. Since $\mathsf{ct}^{\star}_{\mathtt{ID}^{\star},\mathtt{T}^{\star},L+1}$ is the encryption of $\mathsf{M}^{\star}_{\mathsf{coin}} \bigoplus_{\ell\in[|\mathtt{ID}^{\star}|]} \mathsf{M}_{\ell}$, the challenge ciphertext $\mathsf{ct}^{\star}$ is properly distributed.

Hereafter, we check that $\mathcal{B}$ made HIBE secret key queries on $\mathsf{HIBE}.\mathtt{ID}$ only when $\mathsf{HIBE}.\mathtt{ID} \notin \mathsf{prefix}^{+}((\mathtt{ID}^{\star}, \mathtt{T}^{\star}))$ by using the conditions $\mathcal{I} \cap \mathcal{T} = \emptyset$ and $\mathcal{I} \cap \{0,1\}^{\lceil \log(L+1)\rceil}\|\mathcal{N}\|\mathcal{T} = \emptyset$.

Secret Key Generation Query. Upon $\mathcal{A}$'s queries on $\mathtt{ID}$, $\mathcal{B}$ made HIBE secret key queries on $(\mathtt{ID}, (|\mathtt{ID}|+1)\|\varepsilon\|\mathtt{T})$ for $\mathtt{T} \in [\mathtt{T}_{\mathsf{cu}}]$ as we explained in the proof against $\mathcal{A}$ of the Type-I strategy. Since it holds that $\mathcal{I} \cap \{0,1\}^{\lceil \log(L+1)\rceil}\|\mathcal{N}\|\mathcal{T} = \emptyset$ and $\mathcal{T} \cap \{0,1\}^{\lceil \log(L+1)\rceil}\|\mathcal{N}\|\mathcal{T} = \emptyset$, $(\mathtt{ID}, (|\mathtt{ID}|+1)\|\varepsilon\|\mathtt{T}) \notin \mathsf{prefix}^{+}((\mathtt{ID}^{\star}, \mathtt{T}^{\star}))$ holds for all $\mathtt{T} \in [\mathtt{T}_{\mathsf{cu}}]$ even when $\mathtt{ID} = \mathtt{ID}^{\star}$ and $\mathtt{T}^{\star} \in [\mathtt{T}_{\mathsf{cu}}]$.

Secret Key Reveal Query. Upon $\mathcal{A}$'s queries on $\mathtt{ID}$, $\mathcal{B}$ made HIBE secret key queries on $\mathtt{ID}$ as we explained in the proof against $\mathcal{A}$ of the Type-I strategy. Observe that $\mathtt{ID} \in \mathsf{prefix}^{+}((\mathtt{ID}^{\star}, \mathtt{T}^{\star}))$ holds only when $\mathtt{ID} \in \mathsf{prefix}^{+}(\mathtt{ID}^{\star})$ since

$\mathcal{I} \cap \mathcal{T} = \emptyset$ holds. Due to the definition of the Type-II strategy, it holds that $\mathtt{ID} \notin \mathsf{prefix}^{+}(\mathtt{ID}^{\star})$. Therefore, $\mathtt{ID} \notin \mathsf{prefix}^{+}((\mathtt{ID}^{\star}, \mathtt{T}^{\star}))$ holds.

Revoke & Key Update Query. Upon $\mathcal{A}$'s queries on $\mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}}$, $\mathcal{B}$ made HIBE secret key queries on $(\mathtt{ID}, (|\mathtt{ID}|+1)\|\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}, \mathtt{T}_{\mathtt{cu}})$ for all $\mathtt{ID} \in (\mathtt{SKList} \cap \mathcal{I}^{\leq L-1}) \setminus \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}-1}$ and $\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}} \in \mathsf{CoS}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}})$ as we explained in the proof against $\mathcal{A}$ of the Type-I strategy. Since it holds that $\mathcal{I} \cap \{0,1\}^{\lceil \log(L+1) \rceil}\|\mathcal{N}\|\mathcal{T} = \emptyset$ and $\mathcal{T} \cap \{0,1\}^{\lceil \log(L+1) \rceil}\|\mathcal{N}\|\mathcal{T} = \emptyset$, $(\mathtt{ID}, (|\mathtt{ID}|+1)\|\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}, \mathtt{T}_{\mathtt{cu}}) \notin \mathsf{prefix}^{+}((\mathtt{ID}^{\star}, \mathtt{T}^{\star}))$ holds for all $\mathtt{ID} \in (\mathtt{SKList} \cap \mathcal{I}^{\leq L-1}) \setminus \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}-1}$ and $\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}} \in \mathsf{CoS}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}})$ even when $\mathtt{ID}^{\star} \in (\mathtt{SKList} \cap \mathcal{I}^{\leq L-1}) \setminus \mathsf{RL}_{\mathtt{T}_{\mathtt{cu}}-1}$ and $\mathtt{T}^{\star} = \mathtt{T}_{\mathtt{cu}}$.

Decryption Key Reveal Query. Upon $\mathcal{A}$'s queries on $(\mathtt{ID}, \mathtt{T})$, $\mathcal{B}$ made HIBE secret key queries on $((\mathtt{ID}_{[\ell-1]}, \ell\|\tilde{\theta}_{\ell}\|\mathtt{T}))_{\ell \in [|\mathtt{ID}|]}$ for some $\tilde{\theta}_{\ell} \in \mathsf{PS}(\mathtt{ID}_{[\ell]}) \cap \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}_{[\ell]}),\mathtt{T}})$ and $(\mathtt{ID}, \mathtt{T})$ as we explained in the proof against $\mathcal{A}$ of the Type-I strategy. Since it holds that $\mathcal{I} \cap \{0,1\}^{\lceil \log(L+1) \rceil}\|\mathcal{N}\|\mathcal{T} = \emptyset$ and $\mathcal{T} \cap \{0,1\}^{\lceil \log(L+1) \rceil}\|\mathcal{N}\|\mathcal{T} = \emptyset$, $(\mathtt{ID}_{[\ell-1]}, \ell\|\tilde{\theta}_{\ell}\|\mathtt{T}) \notin \mathsf{prefix}^{+}((\mathtt{ID}^{\star}, \mathtt{T}^{\star}))$ holds for all $\ell \in [|\mathtt{ID}|]$ even when $\mathtt{ID} = \mathtt{ID}^{\star}$ and $\mathtt{T} = \mathtt{T}^{\star}$. Moreover, $(\mathtt{ID}, \mathtt{T}) \in \mathsf{prefix}^{+}((\mathtt{ID}^{\star}, \mathtt{T}^{\star}))$ holds only when $(\mathtt{ID}, \mathtt{T}) = (\mathtt{ID}^{\star}, \mathtt{T}^{\star})$ since it holds that $\mathcal{I} \cap \mathcal{T} = \emptyset$. Due to the security definition of RHIBE, $\mathcal{A}$ never made the query on $(\mathtt{ID}^{\star}, \mathtt{T}^{\star})$. Therefore, $(\mathtt{ID}, \mathtt{T}) \notin \mathsf{prefix}^{+}((\mathtt{ID}^{\star}, \mathtt{T}^{\star}))$ holds.

Summarizing the analysis so far, $\mathcal{B}$ made HIBE secret key queries on $\mathsf{HIBE}.\mathtt{ID}$ only when $\mathsf{HIBE}.\mathtt{ID} \notin \mathsf{prefix}^{+}((\mathtt{ID}^{\star}, \mathtt{T}^{\star}))$. Thus, we complete the proof against $\mathcal{A}$ of the Type-II strategy. □

4 Achieving Shorter Ciphertexts/Key Updates

Ma and Lin's basic RIBE [31] is a generic construction from IBE and level-2 HIBE. They suggested that by replacing IBE with identity-based broadcast encryption (IBBE), there is a generic construction of RIBE with shorter ciphertexts. By following the idea, we show that by replacing HIBE of our basic RHIBE scheme in Section 3 with hierarchical identity-based IBBE (HIBBE), there is a generic construction of RHIBE with shorter ciphertexts. Furthermore, we obtain a dual construction of the variant. Specifically, we show that by replacing HIBE with hierarchical identity-based dual IBBE (HDIBBE), there is a generic construction of RHIBE with shorter key updates.

4.1 Hierarchical Identity-based (Dual) IBBE

An IBBE ciphertext $\mathsf{IBBE}.\mathsf{ct}_{\mathsf{PU}}$ depends on a set of privileged users $\mathsf{PU} \in 2^{\mathcal{I}}$ whereas an IBBE secret key $\mathsf{IBBE}.\mathsf{sk}_{\mathtt{id}}$ depends on an element identity $\mathtt{id} \in \mathcal{I}$. The secret key $\mathsf{IBBE}.\mathsf{sk}_{\mathtt{id}}$ can decrypt $\mathsf{IBBE}.\mathsf{ct}_{\mathsf{PU}}$ iff $\mathtt{id} \in \mathsf{PU}$. Let $\mathsf{IBBE}(M)$ denote IBBE with the maximum number M of privileged users in PU. By definition, $\mathsf{IBBE}(1) = \mathsf{IBE}$. The dual IBBE (DIBBE) is the same as IBBE except that the roles of PU and id are swapped such as $\mathsf{DIBBE}.\mathsf{ct}_{\mathtt{id}}$ and $\mathsf{DIBBE}.\mathsf{sk}_{\mathsf{PU}}$.

$\mathsf{HIBBE}(L, M)$ is a conjunction of $\mathsf{HIBE}(L)$ and $\mathsf{IBBE}(M)$, where an HIBBE ciphertext $\mathsf{HIBBE}.\mathsf{ct}_{\mathtt{ID},\mathsf{PU}}$ depends on a hierarchical identity $\mathtt{ID} \in \mathsf{HIBE}.\mathcal{I}^{\leq L}$ and

set of privileged users $\mathsf{PU} \in 2^{\mathsf{IBBE}.\mathcal{I}}$ such that $|\mathsf{PU}| \leq M$ whereas an HIBBE secret key $\mathsf{HIBBE.sk}_{\mathtt{ID},\mathtt{id}}$ depends on a hierarchical identity $\mathtt{ID} \in \mathsf{HIBE}.\mathcal{I}^{\leq L}$ and element identity $\mathtt{id} \in \mathsf{IBBE}.\mathcal{I}$. The secret key $\mathsf{HIBBE.sk}_{\mathtt{ID}',\mathtt{id}}$ can decrypt $\mathsf{HIBBE.ct}_{\mathtt{ID},\mathsf{PU}}$ if $\mathtt{ID}' \in \mathsf{prefix}^{+}(\mathtt{ID}) \wedge \mathtt{id} \in \mathsf{PU}$. There is a special case, where $\mathsf{HIBBE.ct}_{\mathtt{ID},\mathsf{PU}}$ and $\mathsf{HIBBE.sk}_{\mathtt{ID},\mathtt{id}}$ may not depend on any hierarchical identity $\mathtt{ID}$. In this case, we use the notations $\mathsf{HIBBE.ct}_{\mathtt{kgc},\mathsf{PU}}$ and $\mathsf{HIBBE.sk}_{\mathtt{kgc},\mathtt{id}}$. There is another special case, where $\mathsf{HIBBE.ct}_{\mathtt{ID},\mathsf{PU}}$ and $\mathsf{HIBBE.sk}_{\mathtt{ID},\mathtt{id}}$ may not depend on any set of privileged users PU and element identity $\mathtt{id}$, respectively. In this case, we use the notation $\mathsf{HIBBE.sk}_{\mathtt{ID},*}$ that can be used for creating $\mathsf{HIBBE.sk}_{\mathtt{ID}',\mathtt{id}}$ for any $\mathtt{id} \in \mathsf{IBBE}.\mathcal{I}$ if $\mathtt{ID} \in \mathsf{prefix}(\mathtt{ID}')$. $\mathsf{HDIBBE}(L, M)$ that is a conjunction of $\mathsf{HIBE}(L)$ and $\mathsf{DIBBE}(M)$ is defined in the same way.

We can obtain adaptively secure pairing-based $\mathsf{HIBBE}(L, M)$ schemes with compact ciphertexts and $\mathsf{HDIBBE}(L, M)$ schemes with compact secret keys under the k-linear assumption through the predicate encoding framework [3,9,10,43]. It is well known that $(M+1)$-dimensional (pairing-based) inner product encryption ($\mathsf{IPE}(M+1)$) is sufficient for constructing $\mathsf{IBBE}(M)$ and $\mathsf{DIBBE}(M)$. The predicate encoding schemes of $\mathsf{HIBE}(L)$ and $\mathsf{IPE}(M+1)$ with compact ciphertexts and compact secret keys are summarized in [9]. Based on the conjunction of predicate encoding schemes in [3], we obtain predicate encoding schemes of $\mathsf{HIBBE}(L, M)$ with compact ciphertexts and $\mathsf{HDIBBE}(L, M)$ with compact secret keys. Finally, we obtain adaptively secure pairing-based schemes under the k-linear assumption through the generic compilers for predicate encoding schemes [9,10].

4.2 Variants with Shorter Ciphertexts

We replace $\mathsf{HIBE}(L+1)$ by $\mathsf{HIBBE}(L+1, M)$ with compact ciphertexts for constructing $\mathsf{RHIBE}(L)$ with shorter ciphertexts than the scheme in Section 3, where the spirit is almost the same.[6] The level-ℓ ciphertexts $\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell}$ of the scheme in Section 3 for $\ell \in [|\mathtt{ID}|]$ consist of $D+1$ HIBE ciphertexts $\mathsf{HIBE.ct}_{(\mathtt{ID}_{[\ell-1]},\ell\|\theta_{\ell,d}\|\mathtt{T})}$ for $d \in [0, D]$. In other words, the information for each $\theta_{\ell,d}$ is used to create a single HIBE ciphertext. In turn, we use the information for each M-tuple $(\theta_{\ell,(m-1)M}, \theta_{\ell,(m-1)M+1}, \ldots, \theta_{\ell,mM-1})$ to create a single HIBBE ciphertext. Thus, we replace the $D+1$ HIBE ciphertexts of the level-ℓ RHIBE ciphertext by $\lceil (D+1)/M \rceil$ HIBBE ciphertexts $\mathsf{HIBBE.ct}_{\mathtt{ID}_{[\ell-1]},\mathsf{PU}_{\ell,\mathtt{T},m}}$, where

$$\mathsf{PU}_{\ell,\mathtt{T},m} = (\ell\|\theta_{\ell,(m-1)M}\|\mathtt{T}, \ell\|\theta_{\ell,(m-1)M+1}\|\mathtt{T}, \ldots, \ell\|\theta_{\ell,mM-1}\|\mathtt{T})$$

for $m \in [\lceil (D+1)/M \rceil - 1]$ and

$$\mathsf{PU}_{\ell,\mathtt{T},\lceil (D+1)/M \rceil} = (\ell\|\theta_{\ell,(\lceil (D+1)/M \rceil - 1)M}\|\mathtt{T}, \ldots, \ell\|\theta_{\ell,D}\|\mathtt{T}).$$

The level-ℓ decryption keys $\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}|]$ are $\mathsf{HIBBE.sk}_{\mathtt{ID}_{[\ell-1]},\ell\|\tilde{\theta}_\ell\|\mathtt{T}}$ such that $\tilde{\theta}_\ell \in \mathsf{PS}(\mathtt{ID}_{[\ell]}) \cap \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}_{[\ell]}),\mathtt{T}})$. The level-$(L+1)$ ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1}$ and

[6] Ma and Lin [31] only suggested that by replacing IBE with $\mathsf{IBBE}(D+1)$, there is a generic construction of RIBE with shorter ciphertexts. Since we introduce a flexible parameter $M \in [D+1]$, we achieve an efficiency trade-off between the sizes of pp, $\mathsf{sk}_{\mathtt{ID}}$, and $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$ and the size of $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$.

$\mathsf{Setup}(1^\lambda, L)$:
- $(\mathsf{HIBBE.pp}, \mathsf{HIBBE.msk}) \leftarrow \mathsf{HIBBE.Setup}(1^\lambda, L+1, M)$
- return $\mathsf{pp} = \mathsf{HIBBE.pp}$, $\mathsf{sk_{kgc}} = \mathsf{HIBBE.msk}$

$\mathsf{Enc}(\mathsf{pp}, \mathtt{ID}, \mathtt{T}, \mathsf{M})$:
- $(\mathsf{M}_1, \ldots, \mathsf{M}_{|\mathtt{ID}|}) \leftarrow_R \mathcal{M}^{|\mathtt{ID}|}$, $\mathsf{M}_{L+1} = \mathsf{M} \bigoplus_{\ell \in [|\mathtt{ID}|]} \mathsf{M}_\ell$
- for $\ell \in [|\mathtt{ID}|]$
  - $\mathsf{PS}(\mathtt{ID}_{[\ell]}) = (\theta_{\ell,0}, \theta_{\ell,1}, \ldots, \theta_{\ell,D}) \leftarrow \mathsf{Assign}(\mathtt{id}_\ell)$
  - for $m \in [\lceil (D+1)/M \rceil - 1]$
    - $\mathsf{PU}_{\ell,\mathtt{T},m} = (\ell \| \theta_{\ell,(m-1)M} \| \mathtt{T}, \ell \| \theta_{\ell,(m-1)M+1} \| \mathtt{T}, \ldots, \ell \| \theta_{\ell,mM-1} \| \mathtt{T})$
    - $\mathsf{HIBBE.ct}_{\mathtt{ID}_{[\ell-1]}, \mathsf{PU}_{\ell,\mathtt{T},m}} \leftarrow \mathsf{HIBBE.Enc}(\mathsf{HIBBE.pp}, (\mathtt{ID}_{[\ell-1]}, \mathsf{PU}_{\ell,\mathtt{T},m}), \mathsf{M}_\ell)$
  - $\mathsf{PU}_{\ell,\mathtt{T},\lceil (D+1)/M \rceil} = (\ell \| \theta_{\ell,(\lceil (D+1)/M \rceil - 1)M} \| \mathtt{T}, \ldots, \ell \| \theta_{\ell,D} \| \mathtt{T})$
  - $\mathsf{HIBBE.ct}_{\mathtt{ID}_{[\ell-1]}, \mathsf{PU}_{\ell,\mathtt{T},\lceil (D+1)/M \rceil}} \leftarrow$
    - $\mathsf{HIBBE.Enc}(\mathsf{HIBBE.pp}, (\mathtt{ID}_{[\ell-1]}, \mathsf{PU}_{\ell,\mathtt{T},\lceil (D+1)/M \rceil}), \mathsf{M}_\ell)$
  - $\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell} \coloneqq (\mathsf{HIBBE.ct}_{\mathtt{ID}_{[\ell-1]}, \mathsf{PU}_{\ell,\mathtt{T},m}})_{m \in [\lceil (D+1)/M \rceil]}$
- $\mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1} \coloneqq \mathsf{HIBBE.ct}_{(\mathtt{ID},\mathtt{T}),*} \leftarrow \mathsf{HIBBE.Enc}(\mathsf{HIBBE.pp}, ((\mathtt{ID}, \mathtt{T}), *), \mathsf{M}_{L+1})$
- return $\mathsf{ct}_{\mathtt{ID},\mathtt{T}} \coloneqq (\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|] \cup \{L+1\}}$

$\mathsf{GenSK}(\mathsf{pp}, \mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}, \mathtt{ID})$:
- $\mathsf{HIBBE.sk}_{\mathtt{ID},*} \leftarrow \mathsf{HIBBE.KeyGen}(\mathsf{HIBBE.pp}, \mathsf{HIBBE.sk}_{\mathsf{pa}(\mathtt{ID}),*}, (\mathtt{ID}, *))$
- return $\mathsf{sk}_{\mathtt{ID}} \coloneqq \mathsf{HIBBE.sk}_{\mathtt{ID},*}$

$\mathsf{KeyUp}(\mathsf{pp}, \mathtt{T}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{RL}_{\mathtt{ID},\mathtt{T}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$:
- $(\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|] \cup \{L+1\}} \leftarrow \mathsf{GenDK}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$
- $\mathsf{CoS}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}}) = (\bar{\theta}_{|\mathtt{ID}|+1,1}, \bar{\theta}_{|\mathtt{ID}|+1,2}, \ldots, \bar{\theta}_{|\mathtt{ID}|+1,R_{\mathtt{ID},\mathtt{T}}}) \leftarrow \mathsf{Cover}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}})$
  - for $\bar{d} \in [R_{\mathtt{ID},\mathtt{T}}]$
    - $\mathsf{HIBBE.sk}_{\mathtt{ID},(|\mathtt{ID}|+1)\|\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}\|\mathtt{T}} \leftarrow$
      - $\mathsf{HIBBE.KeyGen}(\mathsf{HIBBE.pp}, \mathsf{HIBBE.sk}_{\mathtt{ID},*}, (\mathtt{ID}, (|\mathtt{ID}|+1)\|\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}\|\mathtt{T}))$
- return $\mathsf{ku}_{\mathtt{ID},\mathtt{T}} \coloneqq ((\tilde{\theta}_\ell, \mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|]}, (\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}, \mathsf{HIBBE.sk}_{\mathtt{ID},(|\mathtt{ID}|+1)\|\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}\|\mathtt{T}})_{\bar{d} \in [R_{\mathtt{ID},\mathtt{T}}]})$

$\mathsf{GenDK}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$:
- if $\perp \leftarrow \mathsf{Match}(\mathsf{PS}(\mathtt{ID}_{[|\mathtt{ID}|]}), \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}))$
  - return $\perp$
- else
  - $\mathsf{PU}_{\ell,\mathtt{T},\tilde{m}_\ell} \ni \tilde{\theta}_{|\mathtt{ID}|} \leftarrow \mathsf{Match}(\mathsf{PS}(\mathtt{ID}_{[|\mathtt{ID}|]}), \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}))$
  - $\mathsf{dk}_{\mathtt{ID},\mathtt{T},|\mathtt{ID}|} \coloneqq \mathsf{HIBBE.sk}_{\mathsf{pa}(\mathtt{ID}),|\mathtt{ID}|\|\tilde{\theta}_{|\mathtt{ID}|}\|\mathtt{T}}$
  - $\mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1} \coloneqq \mathsf{HIBBE.sk}_{(\mathtt{ID},\mathtt{T}),*} \leftarrow$
    - $\mathsf{HIBBE.KeyGen}(\mathsf{HIBBE.pp}, \mathsf{HIBBE.sk}_{\mathtt{ID},*}, ((\mathtt{ID}, \mathtt{T}), *))$
  - return $\mathsf{dk}_{\mathtt{ID},\mathtt{T}} \coloneqq ((\mathsf{PU}_{\ell,\mathtt{T},\tilde{m}_\ell}, \mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|]}, \mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1})$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T}}, \mathsf{ct}_{\mathtt{ID},\mathtt{T}})$:
- for $\ell \in [|\mathtt{ID}|]$
  - $\mathsf{M}_\ell \leftarrow \mathsf{HIBBE.Dec}(\mathsf{HIBBE.pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}, \mathsf{HIBBE.ct}_{\mathtt{ID}_{[\ell-1]}, \mathsf{PU}_{\ell,\mathtt{T},\tilde{m}_\ell}})$
- $\mathsf{M}_{L+1} \leftarrow \mathsf{HIBBE.Dec}(\mathsf{HIBBE.pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1}, \mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1})$
- return $\mathsf{M} = \bigoplus_{\ell \in [\mathtt{ID}] \cup \{L+1\}} \mathsf{M}_\ell$

Fig. 6: $\mathsf{RHIBE}(L)$ scheme from $\mathsf{HIBBE}(L+1, M)$

level-$(L+1)$ decryption key $\mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1}$ are $\mathsf{HIBBE.ct}_{(\mathtt{ID},\mathtt{T}),*}$ and $\mathsf{HIBBE.sk}_{(\mathtt{ID},\mathtt{T}),*}$, respectively. By definition of HIBBE, all the level-ℓ decryption keys $\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}$ can be used to decrypt the level-ℓ ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}|] \cup \{L+1\}$. A secret

key $\mathsf{sk}_{\mathtt{ID}}$ is an HIBBE secret key $\mathsf{HIBBE.sk}_{\mathtt{ID},*}$; thus, $\mathsf{sk}_{\mathtt{ID}}$ can be used for creating level-$(L+1)$ decryption key $\mathsf{HIBBE.sk}_{(\mathtt{ID},\mathtt{T}),*}$ and the elements of $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$, i.e., $\mathsf{HIBBE.sk}_{\mathtt{ID},(|\mathtt{ID}|+1)\|\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}\|\mathtt{T}}$ for $\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}} \in \mathsf{CoS}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}})$. Therefore, the $\mathsf{RHIBE}(L)$ scheme from $\mathsf{HIBBE}(L+1,M)$ is correct. We can prove security of the scheme in the same way as the proof of Theorem 1. We describe the scheme as Figure 6.

4.3 Variants with Shorter Key Updates

We replace $\mathsf{HIBE}(L+1)$ by $\mathsf{HDIBBE}(L+1,M)$ with compact secret keys for constructing $\mathsf{RHIBE}(L)$ with shorter key updates than the scheme in Section 3, where the spirit is almost the same. The key update $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$ other than level-ℓ decryption keys $\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}|]$ of the scheme in Section 3 consists of $R_{\mathtt{ID},\mathtt{T}}$ HIBE secret keys $\mathsf{HIBE.sk}_{(\mathtt{ID},|\mathtt{ID}|+1\|\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}}\|\mathtt{T})}$ for $d \in [R_{\mathtt{ID},\mathtt{T}}]$. In other words, the information for each $\theta_{|\mathtt{ID}|+1,\bar{d}}$ is used to create a single HIBE secret key. In turn, we use the information for each M tuple $(\theta_{|\mathtt{ID}|+1,(m-1)M+1}, \theta_{|\mathtt{ID}|+1,(m-1)M+1}, \ldots, \theta_{|\mathtt{ID}|+1,mM})$ to create a single HDIBBE secret key. Thus, we replace the $R_{\mathtt{ID},\mathtt{T}}$ HIBE secret keys by $\lceil R_{\mathtt{ID},\mathtt{T}}/M \rceil$ HDIBBE secret keys $\mathsf{HDIBBE.sk}_{\mathtt{ID},\mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},m}}$, where

$$\mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},m} = (|\mathtt{ID}|+1\|\bar{\theta}_{|\mathtt{ID}|+1,(m-1)M+1}\|\mathtt{T}, \ldots, |\mathtt{ID}|+1\|\bar{\theta}_{|\mathtt{ID}|+1,mM}\|\mathtt{T})$$

for $m \in [\lceil R_{\mathtt{ID},\mathtt{T}}/M \rceil - 1]$ and

$$\begin{aligned}&\mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},\lceil R_{\mathtt{ID},\mathtt{T}}/M \rceil}\\ &= (|\mathtt{ID}|+1\|\theta_{|\mathtt{ID}|+1,(\lceil R_{\mathtt{ID},\mathtt{T}}/M \rceil-1)M+1}\|\mathtt{T}, \ldots, |\mathtt{ID}|+1\|\theta_{|\mathtt{ID}|+1,R_{\mathtt{ID},\mathtt{T}}}\|\mathtt{T}).\end{aligned}$$

The level-ℓ ciphertexts $\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}|]$ are $\mathsf{HDIBBE.ct}_{\mathtt{ID}_{[\ell-1]},\ell\|\theta_{\ell,d}\|\mathtt{T}}$ for $d \in [0,D]$. The level-$(L+1)$ ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1}$ and level-$(L+1)$ decryption key $\mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1}$ are $\mathsf{HDIBBE.ct}_{(\mathtt{ID},\mathtt{T}),*}$ and $\mathsf{HDIBBE.sk}_{(\mathtt{ID},\mathtt{T}),*}$, respectively. By definition of HDIBBE, all the level-ℓ decryption keys $\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}$ can be used to decrypt the level-ℓ ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell}$ for $\ell \in [|\mathtt{ID}|] \cup \{L+1\}$. A secret key $\mathsf{sk}_{\mathtt{ID}}$ is an HDIBBE secret key $\mathsf{HDIBBE.sk}_{\mathtt{ID},*}$; thus, $\mathsf{sk}_{\mathtt{ID}}$ can be used for level-$(L+1)$ decryption key $\mathsf{HDIBBE.sk}_{(\mathtt{ID},\mathtt{T}),*}$ and the elements of $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$, i.e., $\mathsf{HDIBBE.sk}_{\mathtt{ID},(|\mathtt{ID}|+1)\|\mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},m}\|\mathtt{T}}$ for $\bar{\theta}_{|\mathtt{ID}|+1,\bar{d}} \in \mathsf{CoS}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}})$. Therefore, the $\mathsf{RHIBE}(L)$ scheme from $\mathsf{HDIBBE}(L+1,M)$ is correct. We can prove security of the scheme in the same way as the proof of Theorem 1. We describe the scheme as Figure 7.

5 Comparison

In this section, we compare our proposed schemes and previous RHIBE schemes. Table 1 compares our proposed adaptively secure RHIBE schemes with the other adaptively secure schemes proposed by Emura et al. [18] and Lee and Kim's scheme [27]. Specifically, we compare the space efficiency of $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathtt{ID},\mathtt{T}}$, and $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ in terms of the number of group elements. We use Chen and Gong's HIBE scheme [10] and Gong et al.'s HIBE scheme [20] to instantiate Emura et

$\mathsf{Setup}(1^\lambda, L)$:
$(\mathsf{HDIBBE.pp}, \mathsf{HDIBBE.msk}) \leftarrow \mathsf{HDIBBE.Setup}(1^\lambda, L+1)$
return $\mathsf{pp} = \mathsf{HDIBBE.pp}$, $\mathsf{sk_{kgc}} = \mathsf{HDIBBE.msk}$

$\mathsf{Enc}(\mathsf{pp}, \mathtt{ID}, \mathtt{T}, \mathsf{M})$:
$(\mathsf{M}_1, \ldots, \mathsf{M}_{|\mathtt{ID}|}) \leftarrow_R \mathcal{M}^{|\mathtt{ID}|}$, $\mathsf{M}_{L+1} = \mathsf{M} \bigoplus_{\ell \in [|\mathtt{ID}|]} \mathsf{M}_\ell$
for $\ell \in [|\mathtt{ID}|]$
$\mathsf{PS}(\mathtt{ID}_{[\ell]}) = (\theta_{\ell,0}, \theta_{\ell,1}, \ldots, \theta_{\ell,D}) \leftarrow \mathsf{Assign}(\mathtt{id}_\ell)$
for $d \in [0, D]$
$\mathsf{HDIBBE.ct}_{\mathtt{ID}_{[\ell-1]}, \ell \| \theta_{\ell,d} \| \mathtt{T}} \leftarrow \mathsf{HDIBBE.Enc}(\mathsf{HDIBBE.pp}, (\mathtt{ID}_{[\ell-1]}, \ell \| \theta_{\ell,d} \| \mathtt{T}), \mathsf{M}_\ell)$
$\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell} \coloneqq (\mathsf{HDIBBE.ct}_{(\mathtt{ID}_{[\ell-1]}, \ell \| \theta_{\ell,d} \| \mathtt{T})})_{d \in [0,D]}$
$\mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1} \coloneqq \mathsf{HDIBBE.ct}_{(\mathtt{ID},\mathtt{T}),*} \leftarrow \mathsf{HDIBBE.Enc}(\mathsf{HDIBBE.pp}, ((\mathtt{ID}, \mathtt{T}), *), \mathsf{M}_{L+1})$
return $\mathsf{ct}_{\mathtt{ID},\mathtt{T}} \coloneqq (\mathsf{ct}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|] \cup \{L+1\}}$

$\mathsf{GenSK}(\mathsf{pp}, \mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}, \mathtt{ID})$:
$\mathsf{HDIBBE.sk}_{\mathtt{ID},*} \leftarrow \mathsf{HDIBBE.KeyGen}(\mathsf{HDIBBE.pp}, \mathsf{HDIBBE.sk}_{\mathsf{pa}(\mathtt{ID}),*}, (\mathtt{ID}, *))$
return $\mathsf{sk}_{\mathtt{ID}} \coloneqq \mathsf{HDIBBE.sk}_{\mathtt{ID},*}$

$\mathsf{KeyUp}(\mathsf{pp}, \mathtt{T}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{RL}_{\mathtt{ID},\mathtt{T}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$:
$(\mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|] \cup \{L+1\}} \leftarrow \mathsf{GenDK}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$
$\mathsf{CoS}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}}) = (\bar{\theta}_{|\mathtt{ID}|+1,1}, \bar{\theta}_{|\mathtt{ID}|+1,2}, \ldots, \bar{\theta}_{|\mathtt{ID}|+1,R_{\mathtt{ID},\mathtt{T}}}) \leftarrow \mathsf{Cover}(\mathsf{RL}_{\mathtt{ID},\mathtt{T}})$
for $m \in [\lceil R_{\mathtt{ID},\mathtt{T}}/M \rceil - 1]$
$\mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},m} = (|\mathtt{ID}|+1 \| \bar{\theta}_{|\mathtt{ID}|+1,(m-1)M+1} \| \mathtt{T}, \ldots, |\mathtt{ID}|+1 \| \bar{\theta}_{|\mathtt{ID}|+1,mM} \| \mathtt{T})$
$\mathsf{HDIBBE.sk}_{\mathtt{ID},\mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},m}} \leftarrow$
$\mathsf{HDIBBE.KeyGen}(\mathsf{HDIBBE.pp}, \mathsf{HDIBBE.sk}_{\mathtt{ID},*}, (\mathtt{ID}, \mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},m}))$
$\mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},\lceil R_{\mathtt{ID},\mathtt{T}}/M \rceil} =$
$(|\mathtt{ID}|+1 \| \theta_{|\mathtt{ID}|+1,(\lceil R_{\mathtt{ID},\mathtt{T}}/M \rceil - 1)M+1} \| \mathtt{T}, \ldots, |\mathtt{ID}|+1 \| \theta_{|\mathtt{ID}|+1,R_{\mathtt{ID},\mathtt{T}}} \| \mathtt{T})$
$\mathsf{HDIBBE.sk}_{\mathtt{ID},\mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},\lceil R_{\mathtt{ID},\mathtt{T}}/M \rceil}} \leftarrow$
$\mathsf{HDIBBE.KeyGen}(\mathsf{HDIBBE.pp}, \mathsf{HDIBBE.sk}_{\mathtt{ID},*}, (\mathtt{ID}, \mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},\lceil R_{\mathtt{ID},\mathtt{T}}/M \rceil}))$
return $\mathsf{ku}_{\mathtt{ID},\mathtt{T}} \coloneqq \begin{pmatrix} (\tilde{\theta}_\ell, \mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|]}, \\ (\mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},m}, \mathsf{HDIBBE.sk}_{\mathtt{ID},\mathsf{PU}_{|\mathtt{ID}|+1,\mathtt{T},m}})_{m \in [\lceil R_{\mathtt{ID},\mathtt{T}}/M \rceil]} \end{pmatrix}$

$\mathsf{GenDK}(\mathsf{pp}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$:
if $\perp \leftarrow \mathsf{Match}(\mathsf{PS}(\mathtt{ID}_{[|\mathtt{ID}|]}), \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}))$
return $\perp$
else
$\mathsf{PU}_{|\mathtt{ID}|,\mathtt{T},\tilde{m}_{|\mathtt{ID}|}} \ni \tilde{\theta}_{|\mathtt{ID}|} \leftarrow \mathsf{Match}(\mathsf{PS}(\mathtt{ID}_{[|\mathtt{ID}|]}), \mathsf{CoS}(\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}))$
$\mathsf{dk}_{\mathtt{ID},\mathtt{T},|\mathtt{ID}|} \coloneqq \mathsf{HDIBBE.sk}_{\mathsf{pa}(\mathtt{ID}),\mathsf{PU}_{|\mathtt{ID}|,\mathtt{T},\tilde{m}_{|\mathtt{ID}|}}}$
$\mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1} \coloneqq \mathsf{HDIBBE.sk}_{(\mathtt{ID},\mathtt{T}),*} \leftarrow$
$\mathsf{HDIBBE.KeyGen}(\mathsf{HDIBBE.pp}, \mathsf{HDIBBE.sk}_{\mathtt{ID},*}, ((\mathtt{ID}, \mathtt{T}), *))$
return $\mathsf{dk}_{\mathtt{ID},\mathtt{T}} \coloneqq ((\tilde{\theta}_\ell, \mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell})_{\ell \in [|\mathtt{ID}|]}, \mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1})$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T}}, \mathsf{ct}_{\mathtt{ID},\mathtt{T}})$:
for $\ell \in [|\mathtt{ID}|]$
$\mathsf{M}_\ell \leftarrow \mathsf{HDIBBE.Dec}(\mathsf{HDIBBE.pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T},\ell}, \mathsf{HDIBBE.ct}_{\mathtt{ID}_{[\ell-1]}, \ell \| \tilde{\theta}_\ell \| \mathtt{T}})$
$\mathsf{M}_{L+1} \leftarrow \mathsf{HDIBBE.Dec}(\mathsf{HDIBBE.pp}, \mathsf{dk}_{\mathtt{ID},\mathtt{T},L+1}, \mathsf{ct}_{\mathtt{ID},\mathtt{T},L+1})$
return $\mathsf{M} = \bigoplus_{\ell \in [\mathtt{ID}] \cup \{L+1\}} \mathsf{M}_\ell$

Fig. 7: $\mathsf{RHIBE}(L)$ scheme from $\mathsf{HDIBBE}(L+1, M)$

al.'s scheme, Lee and Kim's scheme, and our proposed scheme in Section 3. Therefore, all the schemes are based on the same SXDH assumption. Here, let $|\mathtt{ID}| = \ell$ and let $R_{\mathtt{ID},\mathtt{T}}$ denote the number of identities in the revocation list $\mathsf{RL}_{\mathtt{ID},\mathtt{T}}$. As described in Section 2.3, $|\mathsf{CoS}| = O(R_{\mathtt{ID},\mathtt{T}} \log(2^D/R_{\mathtt{ID},\mathtt{T}}))$; however, we simply write $|\mathsf{CoS}| \approx O(R_{\mathtt{ID},\mathtt{T}} D')$ and $|\mathsf{CoS}| \approx O(R_{\mathtt{ID},\mathtt{T}} D)$. We note that there are no adaptively secure RHIBE schemes with the SD.

Table 1: Comparison of adaptively secure $\mathsf{RHIBE}(L)$ schemes

Scheme	$\lvert\mathsf{pp}\rvert$	$\lvert\mathsf{ct}_{\mathtt{ID},\mathtt{T}}\rvert$	$\lvert\mathsf{sk}_{\mathtt{ID}}\rvert$	$\lvert\mathsf{ku}_{\mathtt{ID},\mathtt{T}}\rvert$	$\lvert\mathsf{dk}_{\mathtt{ID},\mathtt{T}}\rvert$
ETW20 [18] + [10,12]	$O(L)$	$O(1)$	$O(\lvert\mathsf{DelK}_{\mathtt{ID}}\rvert(L-\ell)) + O(D(L-\ell))$	$O(R_{\mathtt{ID},\mathtt{T}}D) \times O(L-\ell)$	$O(1)$
ETW20 [18] + [20]	$O(1)$	$O(\ell)$	$O(\lvert\mathsf{DelK}_{\mathtt{ID}}\rvert\ell) + O(D\ell)$	$O(R_{\mathtt{ID},\mathtt{T}}D\ell)$	$O(\ell)$
LK21 [27] + [10,12] (basic)	$O(L)$	$O(\ell D)$	$O(L-\ell)$	$O(R_{\mathtt{ID},\mathtt{T}}D+\ell)$	$O(\ell)$
LK21 [27] + [20] (basic)	$O(1)$	$O(\ell^2 D)$	$O(\ell)$	$O(R_{\mathtt{ID},\mathtt{T}}D\ell+\ell^2)$	$O(\ell^2)$
LK21 [27] + [10,12] (shorter ct)	$O(L+D)$	$O(\ell)$	$O(L+D-\ell)$	$O(R_{\mathtt{ID},\mathtt{T}}D^2+\ell)$	$O(\ell)$
Section 3 + [10,12] (basic)	$O(L)$	$O(\ell D)$	$O(L-\ell)$	$O(R_{\mathtt{ID},\mathtt{T}}D+\ell)$	$O(\ell)$
Section 3 + [20] (basic)	$O(1)$	$O(\ell^2 D)$	$O(\ell)$	$O(R_{\mathtt{ID},\mathtt{T}}D\ell+\ell^2)$	$O(\ell^2)$
Section 4.2 (shorter ct)	$O(L+M)$	$O\left(\ell\frac{D}{M}\right)$	$O(L+M-\ell)$	$O(R_{\mathtt{ID},\mathtt{T}}MD+\ell)$	$O(\ell)$
Section 4.3 (shorter ku)	$O(L+M)$	$O(\ell MD)$	$O(L+M-\ell)$	$O\left(\frac{R_{\mathtt{ID},\mathtt{T}}D}{M}+\ell\right)$	$O(\ell)$

Comparison with ETW20. Let $\mathsf{DelK}_{\mathtt{ID}}$ denote the number of delegation keys in Emura et al.'s scheme. When instantiated by the same HIBE schemes such as [10,12,20], Emura et al.'s scheme has larger $\mathsf{sk}_{\mathtt{ID}}$ and $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$, while our scheme has larger $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$ and $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$. Since $|\mathsf{DelK}_{\mathtt{ID}}|$ grows when $\mathtt{ID}$ creates their children's secret keys and their revocation list $\mathsf{RL}_{\mathtt{ID},\mathtt{T}}$ is updated, the size is not static. Therefore, we believe that our proposed scheme is practically more efficient than Emura et al.'s scheme.

Comparison with LK21. When instantiated by the same HIBE schemes such as [10,12,20], Lee and Kim's basic schemes and our basic schemes in Section 3 have the same asymptotic efficiency. Nevertheless, the concrete efficiency of our schemes is better than Lee and Kim's schemes since we use one HIBE scheme while Lee and Kim used two HIBE schemes for constructing RHIBE as we claimed at the end of Section 1.

We can set the parameter $M \in [D]$ in our shorter ciphertext variant in Section 4.2 whereas we can set the parameter M as an arbitrary non-negative integer in our shorter key udpate variant in Section 4.3. When $M = 1$, both variants achieve the same asymptotic efficiency as Lee and Kim's basis scheme and our basic scheme instantiated by [10,12]. When $M = D$, our shorter ciphertext variant has the same asymptotic efficiency as Lee and Kim's shorter ciphertext variant; however, our shorter cipehrtext variant can take flexible choice of the parameter $M \in [D]$ as opposed to Lee and Kim's scheme. Needless to say, our shorter key update variant has the shortest $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$ than any other adaptively secure schemes. What is more, although we do not summarize in Table 1, our shorter key update variant with a large M achieves the shorter key updates than selectively secure RHIBE schemes with the SD [27,29].

Acknowledgement. We would like to thank anonymous reviewers of Asiacrypt 2019 and PKC 2020.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. EUROCRYPT 2010, LNCS 6110, pp. 553–572. Springer (2010)
2. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. CRYPTO 2010, LNCS 6223, pp. 98–115. Springer (2010)
3. Ambrona, M., Barthe, G., Schmidt, B.: Generic transformations of predicate encodings: Constructions and applications. CRYPTO 2017, LNCS 10401, pp. 36–66. Springer (2017)
4. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. CCS 2008. pp. 417–426. ACM (2008)
5. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. EUROCRYPT 2014, LNCS 8441, pp. 533–556. Springer (2014)

6. Brakerski, Z., Lombardi, A., Segev, G., Vaikuntanathan, V.: Anonymous ibe, leakage resilience and circular security from new assumptions. EUROCRYPT 2018, LNCS 10820, pp. 535–564. Springer (2018)
7. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. J. Cryptology **25**(4), 601–639 (2012)
8. Chang, D., Chauhan, A.K., Kumar, S., Sanadhya, S.K.: Revocable identity-based encryption from codes with rank metric. CT-RSA 2018, LNCS 10808, pp. 435–451. Springer (2018)
9. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. EUROCRYPT 2015, LNCS 9057, pp. 595–624. Springer (2015)
10. Chen, J., Gong, J.: ABE with tag made easy - concise framework and new instantiations in prime-order groups. ASIACRYPT 2017, LNCS 10625, pp. 35–65. Springer (2017)
11. Chen, J., Lim, H.W., Ling, S., Wang, H., Nguyen, K.: Revocable identity-based encryption from lattices. ACISP 2012, LNCS 7372, pp. 390–403. Springer (2012)
12. Chen, J., Wee, H.: Dual system groups and its applications - compact HIBE and more. IACR Cryptology ePrint Archive **2014**, 265 (2014)
13. Cocks, C.C.: An identity based encryption scheme based on quadratic residues. Cryptography and Coding, LNCS 2260, pp. 360–363. Springer (2001)
14. Döttling, N., Garg, S.: From selective IBE to full IBE and selective HIBE. TCC 2017, LNCS 10677, pp. 372–408. Springer (2017)
15. Döttling, N., Garg, S.: Identity-based encryption from the Diffie-Hellman assumption. CRYPTO 2017, LNCS 10401, pp. 537–569. Springer (2017)
16. Emura, K., Seo, J.H., Watanabe, Y.: Efficient revocable identity-based encryption with short public parameters. Theor. Comput. Sci. **863**, 127–155 (2021)
17. Emura, K., Seo, J.H., Youn, T.: Semi-generic transformation of revocable hierarchical identity-based encryption and its DBDH instantiation. IEICE Transactions **99-A**(1), 83–91 (2016)
18. Emura, K., Takayasu, A., Watanabe, Y.: Adaptively secure revocable hierarchical IBE from k-linear assumption. IACR Cryptol. ePrint Arch. **2020**, 886 (2020)
19. Gaborit, P., Hauteville, A., Phan, D.H., Tillich, J.: Identity-based encryption from codes with rank metric. CRYPTO 2017, LNCS 10403, pp. 194–224. Springer (2017)
20. Gong, J., Cao, Z., Tang, S., Chen, J.: Extended dual system group and shorter unbounded hierarchical identity based encryption. Des. Codes Cryptography **80**(3), 525–559 (2016)
21. Hu, Z., Liu, S., Chen, K., Liu, J.K.: Revocable identity-based encryption and server-aided revocable ibe from the computational Diffie-Hellman assumption. Cryptography **2**(4) (2018)
22. Katsumata, S., Matsuda, T., Takayasu, A.: Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance. PKC 2019, LNCS 11443, pp. 441–471. Springer (2019)
23. Langrehr, R., Pan, J.: Tightly secure hierarchical identity-based encryption. PKC 2019, LNCS 11442, pp. 436–465. Springer (2019)
24. Langrehr, R., Pan, J.: Hierarchical identity-based encryption with tight multi-challenge security. PKC 2020, LNCS 12110, pp. 153–183. Springer (2020)
25. Langrehr, R., Pan, J.: Unbounded HIBE with tight security. ASIACRYPT 2020, LNCS 12492, pp. 129–159. Springer (2020)
26. Lee, K.: A generic construction for revocable identity-based encryption with subset difference methods. IACR Cryptology ePrint Archive **2019**, 798 (2019)

27. Lee, K., Kim, J.S.: A generic approach to build revocable hierarchical identity-based encryption. IACR Cryptology ePrint Archive **2021**, 502 (2021)
28. Lee, K., Lee, D.H., Park, J.H.: Efficient revocable identity-based encryption via subset difference methods. Des. Codes Cryptography **85**(1), 39–76 (2017)
29. Lee, K., Park, S.: Revocable hierarchical identity-based encryption with shorter private keys and update keys. Des. Codes Cryptography **86**(10), 2407–2440 (2018)
30. Libert, B., Vergnaud, D.: Adaptive-ID secure revocable identity-based encryption. CT-RSA 2009, LNCS 5473, pp. 1–15. Springer (2009)
31. Ma, X., Lin, D.: Generic constructions of revocable identity-based encryption. Inscrypt 2019, LNCS 12020, pp. 381–396. Springer (2019)
32. Ma, X., Lin, D.: Generic constructions of ribe via subset difference method. IACR Cryptology ePrint Archive **2019**, 1376 (2019)
33. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. CRYPTO 2001, LNCS 2139, pp. 41–62. Springer (2001)
34. Ryu, G., Lee, K., Park, S., Lee, D.H.: Unbounded hierarchical identity-based encryption with efficient revocation. WISA 2015, LNCS 9503, pp. 122–133. Springer (2015)
35. Seo, J.H., Emura, K.: Revocable identity-based encryption revisited: Security model and construction. PKC 2013, LNCS 7778, pp. 216–234. Springer (2013)
36. Seo, J.H., Emura, K.: Revocable hierarchical identity-based encryption. Theor. Comput. Sci. **542**, 44–62 (2014)
37. Seo, J.H., Emura, K.: Revocable hierarchical identity-based encryption via history-free approach. Theor. Comput. Sci. **615**, 45–60 (2016)
38. Takayasu, A., Watanabe, Y.: Lattice-based revocable identity-based encryption with bounded decryption key exposure resistance. ACISP 2017, LNCS 10342, pp. 184–204. Springer (2017)
39. Takayasu, A., Watanabe, Y.: Revocable identity-based encryption with bounded decryption key exposure resistance: Lattice-based construction and more. Theor. Comput. Sci. **849**, 64–98 (2021)
40. Wang, S., Zhang, J., He, J., Wang, H., Li, C.: Simplified revocable hierarchical identity-based encryption from lattices. CANS 2019, LNCS 11829, pp. 99–119. Springer (2019)
41. Waters, B.: Efficient identity-based encryption without random oracles. EUROCRYPT 2005, LNCS 3494, pp. 114–127. Springer (2005)
42. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. CRYPTO 2009, LNCS 5677, pp. 619–636. Springer (2009)
43. Wee, H.: Dual system encryption via predicate encodings. TCC 2014, LNCS 8349, pp. 616–637. Springer (2014)