Adaptively Secure Revocable Hierarchical IBE from k-linear Assumption[*]

Keita Emura[†] Atsushi Takayasu[†] Yohei Watanabe[‡] [§]

July 17, 2020

Abstract

Revocable identity-based encryption (RIBE) is an extension of IBE with an efficient key revocation mechanism. *Revocable hierarchical IBE (RHIBE)* is its further extension with key delegation functionality. Although there are various adaptively secure pairing-based RIBE schemes, all known hierarchical analogues only satisfy *selective security*. In addition, the currently known most efficient adaptively secure RIBE and selectively secure RHIBE schemes rely on *non-standard assumptions*, which are referred to as the augmented DDH assumption and q-type assumptions, respectively. In this paper, we propose a simple but effective design methodology for RHIBE schemes. We provide a generic design framework for RHIBE based on an HIBE scheme with a few properties. Fortunately, several state-of-the-art pairing-based HIBE schemes have the properties. In addition, our construction preserves the sizes of master public keys, ciphertexts, and decryption keys, as well as the complexity assumptions of the underlying HIBE scheme. Thus, we obtain the first RHIBE schemes with *adaptive security* under the standard *k-linear assumption*. We prove adaptive security by developing a new proof technique for RHIBE. Due to the compactness-preserving construction, the proposed R(H)IBE schemes have similar efficiencies to the most efficient existing schemes.

[*] This work is supported by JST CREST Grant Number JPMJCR14D6, JSPS KAKENHI Grant Number JP17K12697, JP18H05289, and MEXT Leading Initiative for Excellent Young Researchers.

[†] National Institute of Information and Communications Technology (NICT), Japan. {k-emura, takayasu}@nict.go.jp

[‡] The University of Electro-Communications, Japan. watanabe@uec.ac.jp

[§] National Institute of Advanced Industrial Science and Technology (AIST), Japan.

Contents

1 Introduction

1.1 Background

Identity-based encryption (IBE), which was proposed by Boneh and Franklin [BF01], is an advanced form of public-key encryption (PKE), where an arbitrary string (e.g., usernames or e-mail addresses) can be used as users' public keys. Thus, unlike traditional PKE, IBE systems do not require a public key infrastructure (PKI) to create certificates for each public key. A well-known extension of IBE is *hierarchical IBE* (HIBE), which has key delegation functionality that allows the decentralization of the power of key creations from the key generation center (KGC) to users. Specifically, users in HIBE form a tree structure, and each user can produce their children's secret keys. As a result, this functionality realizes efficient management of a large number of users. Departing from other constructions [ABB10a, ABB10b, CHKP12, DG17, Zha12], pairing-based schemes are only known constructions for achieving *adaptive security* in the standard model, e.g., [CG17, CW14, GCTC16, LP19, LP20, Lew12, LW10, LW11, Wat09]. Furthermore, several adaptively secure pairing-based schemes have compact ciphertexts [CG17, CW14, LW10, OT15, RS14] or compact master public keys [GCTC16, Lew12, LW11]. Thus, in this paper, we primarily focus on adaptively secure pairing-based schemes in the standard model.

The other IBE variant is *revocable IBE (RIBE)* [BGK08], which enables efficient revocation of identities as required (e.g., if laptops are corrupted or leak their secret keys). There are long-term secret and short-term decryption keys per identity in RIBE, and the long-term key is responsible for updating the decryption keys. Specifically, the decryption key can be updated using the secret key and key update information, which is broadcast by the KGC; however, updating the decryption key fails if the KGC revokes the user when generating the key update. Therefore, RIBE realizes the revocation mechanism. Here, the secret key is only used to generate the decryption key; thus, it can be stored on a more powerful and secure device. However, since decryption keys are used frequently, they tend to be stored on more vulnerable devices. Therefore, it is desirable to ensure security when decryption keys are leaked. Based on this situation, Seo and Emura [SE13b] introduced a new security notion called the decryption key exposure resistance (DKER) which has become the standard security notion for RIBE. In fact, RIBE with the DKER ensures that non-exposed decryption keys can still be used without compromising security even if a number of decryption keys (except that of the target user and the target time period) are exposed. The revocation mechanism can also be considered in the hierarchical setting [SE13a], which is referred to as *revocable HIBE (RHIBE)*. Each RHIBE user is responsible for the key delegation functionality and revocation of their children users. Thus, each RHIBE user also manages a binary tree. An RHIBE secret key has an additional component for key delegation (hereafter, delegation key). As with RIBE, in addition to updating keys, delegation keys can be stored on a more powerful and secure device because key delegations are not performed frequently. In the RHIBE context, there are two types of DKER notions from a historical perspective. The primary (or weaker) DKER notion was proposed by Seo and Emura [SE15b], and then Katsumata et al. [KMT19] introduced a stronger notion. Generally, both notions give the same security guarantee as the DKER in the RIBE setting, i.e., non-exposed decryption keys can still be used without compromising security even if a number of decryption keys are exposed except that of the target user and the target time period. The difference between the weaker and the stronger DKER is which decryption keys are exposed. In the stronger notion, a number of decryption keys are exposed except that of the target user and the target time period, while those of the target-user's ancestor and the target time period are not exposed in the weaker notion. The lattice-based RHIBE [WZH$^+$19] considered the stronger DKER as the standard security notion.

To date, a number of adaptively secure pairing-based RIBE schemes with the DKER have been proposed [GW19, ISW17, LLP17, SE13b, SZSM17, WLXZ14, WES17]. Recently, several generic constructions of RIBE from two-level HIBE [Lee19, ML19a, ML19b] have also been proposed, although they require large ciphertexts whose sizes depend on a length of identities. Among these schemes, the scheme proposed by Watanabe et al. [WES17] and its variant [GW19] are the most efficient because they satisfy prime-order bilinear groups, compact master public keys, and compact ciphertexts simultaneously. However, the security of these schemes relies on the *augmented* DDH assumption, which is a non-standard variant of the traditional DDH assumption. For RHIBE, the situation is even worse. Although several pairing-based schemes with the weaker (or primary) DKER [ESY16, LP18, RLPL15, SE15b][1] have been proposed, they only satisfy selective security. In addition, the security of all schemes (except [ESY16]) is based on q-type assumptions, while the scheme [ESY16] is less efficient than other schemes [LP18, RLPL15, SE15b]. Unfortunately unlike the RIBE case, there is no generic construction for RHIBE schemes. Therefore, there are no adaptively secure RHIBE schemes, even if we ignore the stronger DKER and permit non-standard assumptions.

1.2 Our Contributions

In this paper, we demonstrate significant progress relative to constructing adaptively secure pairing-based RHIBE schemes with the stronger DKER. We extract the core essence of existing schemes and propose a generic construction for an RHIBE scheme from an HIBE scheme with mild requirements that are satisfied by several state-of-the-art pairing-based HIBE schemes [CG17, CW14, GCTC16]. The primary contributions of the proposed generic construction are summarized as follows.

- We develop a new proof technique such that the construction is *adaptiveness-preserving*, i.e., the proposed scheme achieves adaptive security if the underlying HIBE scheme satisfies the same security level.

- Our construction is *compactness-preserving*, i.e., the proposed RHIBE scheme have the same size master public keys, ciphertexts, and decryption keys as the underlying HIBE scheme. Note that instantiations of the proposed scheme suffer from larger delegation keys than those of existing constructions. Informally, each delegation key comprises several sub-delegation keys, which, in previous RHIBE schemes, comprise only $O(1)$ group elements. In contrast, that of our instantiations is a secret key of the underlying HIBE scheme; thus, the size of the latter depends on the level of the given user. However, we can store delegation (and updating) keys on a powerful device; thus, we believe that the larger secret key-size is not a significant issue.

Therefore, we obtain the first adaptively secure RHIBE schemes with the stronger DKER from the standard k-linear assumption in the standard model based on [CG17, CW14, GCTC16]. The definition of RHIBE is complicated; thus, we provide an overview of our results in Section 2.

Table 1 compares the proposed RHIBE schemes with previous ones.[2] Here, we use [CG17, CW14] and [GCTC16] as the underlying HIBE schemes, and we instantiate the proposed RHIBE schemes with compact ciphertexts and compact master public keys. Although we omit the details, the secret key of the proposed schemes based on [CG17, CW14] (resp. [GCTC16]) are larger than those of other schemes by factors $O(L - \ell)$ (resp. $O(\ell)$). The proposed schemes based on [CG17, CW14]

[1] To be precise, these works are prior to Katsumata et al. [KMT19]; thus, they do not consider the stronger DKER.

[2] Note that we consider insider security as the minimum requirement. The definition and necessity of insider security is discussed in Section 1.3.

Table 1: Comparison of pairing-based RHIBE schemes

Scheme	$\lvert\mathsf{MPK}\rvert$	$\lvert\mathsf{ct}_{\mathtt{ID},\mathtt{T}}\rvert$	$\lvert\mathsf{dk}_{\mathtt{ID},\mathtt{T}}\rvert$	adaptive?	DKER	Assumption
SE15[SE15b]	$O(L)$	$O(1)$	$O(1)$	selective	strong	q-type
RLPL15 [RLPL15]	$O(1)$	$O(\ell)$	$O(\ell)$	selective	weak	q-type
LP18 [LP18]	$O(1)$	$O(\ell)$	$O(\ell)$	selective	weak	q-type
ESY16 [ESY16]	$O(L)$	$O(\ell)$	$O(\ell)$	selective	weak	DBDH
Ours+[CG17, CW14]	$O(L)$	$O(1)$	$O(1)$	adaptive	strong	k-Lin
Ours+[GCTC16]	$O(1)$	$O(\ell)$	$O(\ell)$	adaptive	strong	k-Lin

– $\lvert\mathsf{MPK}\rvert$, $\lvert\mathsf{ct}_{\mathtt{ID},\mathtt{T}}\rvert$, and $\lvert\mathsf{dk}_{\mathtt{ID},\mathtt{T}}\rvert$ denote the sizes of the master public key, ciphertext, and decryption key of user $\mathtt{ID}$ at time $\mathtt{T}$, respectively, in terms of the number of group elements.

– L and ℓ denote the maximum hierarchical size and level of $\mathtt{ID}$, respectively.

– q-type, DBDH, k-Lin stand for a q-type assumption, the decisional bilinear Diffie-Hellman assumption, and the k-linear assumption, respectively.

with compact ciphertexts have the same asymptotic efficiency as Seo and Emura's scheme [SE15b] in terms of MPK, $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$, and $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$-size. Similarly, the proposed scheme based on [GCTC16] with compact master public keys has the same asymptotic efficiency as the scheme proposed by Ryu et al. [RLPL15] and that proposed by Lee-Park [LP18]. Note that all of these schemes have better asymptotic efficiencies than the scheme proposed by Emura et al. [ESY16]. As we have claimed, only the proposed schemes achieve adaptive security. To compare DKER, we evaluate whether the existing schemes achieve the stronger DKER because they only have security proofs for the weaker DKER. Note that Seo and Emura's scheme [SE15b] can be modified to achieve the stronger DKER under the stronger assumption.[3] However, we cannot find analogous modifications for the schemes proposed by Ryu et al. [RLPL15], Lee-Park [LP18], and Emura et al. [ESY16]. We also note that there are concrete attacks against these schemes under the stronger DKER model, and that this fact does not imply that the security proofs of these works [ESY16, LP18, RLPL15] contain bugs. While all existing schemes (except that proposed by Emura et al. [ESY16]) are based on q-type assumptions, the proposed schemes are based on the standard k-linear assumption.

Note that the benefit of our results is not limited to RHIBE. Table 2 compares the proposed RIBE scheme to the scheme proposed by Watanabe et al. [WES17] and its variant [GW19]. These existing schemes are modifications of Jutla and Roy's IBE scheme [JR17], we compare them to the proposed scheme based on Chen-Gong's HIBE scheme [CG17], which is an extension of the scheme proposed by Jutla-Roy. Specifically, our instantiation is based on the SXDH assumption, which is a specific case of the k-linear assumption for $k = 1$, while the existing schemes rely on a non-standard augmented DDH assumption. In addition, the reduction loss of the proposed scheme is better than that of the other schemes. Although our instantiation has slightly larger secret keys and key updates, the gap is not significant. In contrast, the proposed scheme has shorter master public keys, ciphertexts, and decryption keys.

[3] Although the security of Seo and Emura's original scheme is based on the q-weak bilinear Diffie-Hellman inversion assumption, that of the modified scheme is based on the q-bilinear Diffie-Hellman exponent assumption [Sha07].

Table 2: Efficiency comparison of adaptive-identity secure RIBE schemes with the DKER in prime-order bilinear groups, compact master public keys, and compact ciphertexts

Scheme	$\lvert\mathsf{MPK}\rvert$	$\lvert\mathsf{ct}_{\mathtt{ID},\mathtt{T}}\rvert$	$\lvert\mathsf{sk}_{\mathtt{ID},\theta}\rvert$
	$(\lvert\mathbb{G}_1\rvert, \lvert\mathbb{G}_2\rvert, \lvert\mathbb{G}_T\rvert)$	$(\lvert\mathbb{G}_1\rvert, \lvert\mathbb{G}_T\rvert, \lvert\mathbb{Z}_p\rvert)$	$(\lvert\mathbb{G}_2\rvert, \lvert\mathbb{Z}_p\rvert)$
WES17 [WES17]	$(6, 10, 1)$	$(4, 1, 1)$	$(5, 0)$
GW19 [GW19]	$(6, 10, 1)$	$(4, 1, 1)$	$(5, 1)$
Ours+ [CG17]	$(5, 7, 1)$	$(3, 1, 1)$	$(7, 0)$

Scheme	$\lvert\mathsf{ku}_{\mathtt{T},\theta}\rvert$	$\lvert\mathsf{dk}_{\mathtt{ID},\mathtt{T}}\rvert$	reduction	assumption
	$\lvert\mathbb{G}_2\rvert$	$(\lvert\mathbb{G}_2\rvert, \lvert\mathbb{Z}_p\rvert)$	loss	
WES17 [WES17]	3	$(6, 0)$	$O(Q^2\lvert\mathcal{T}\rvert)$	ADDH1, DDH2
GW19 [GW19]	3	$(6, 1)$	$O(Q^2\lvert\mathcal{T}\rvert)$	ADDH1, DDH2
Ours+ [CG17]	7	$(5, 0)$	$O(Q(Q + \lvert\mathcal{T}\rvert))$	SXDH

- All schemes use asymmetric bilinear maps $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Groups $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ have prime order p.
- $\lvert\mathsf{MPK}\rvert, \lvert\mathsf{ct}_{\mathtt{ID},\mathtt{T}}\rvert$, and $\lvert\mathsf{dk}_{\mathtt{ID},\mathtt{T}}\rvert$ denote the sizes of the master public key, ciphertext, and decryption key of user ID at time T, respectively, in terms of the number of group elements and $\mathbb{Z}_p$ elements.
- $\lvert\mathsf{sk}_{\mathtt{ID},\theta}\rvert$ and $\lvert\mathsf{ku}_{\mathtt{T},\theta}\rvert$ denote the sizes of the secret key of user ID and key update of time period T associated with a single node θ, respectively, in terms of the number of $\mathbb{G}_2$ elements and $\mathbb{Z}_p$ elements.
- Q and $\lvert\mathcal{T}\rvert$ denote the number of secret key generation queries and the size of the time period space, respectively.
- ADDH1, DDH2, and SXDH represent the augmented decisional Diffie-Hellman assumption in $\mathbb{G}_1$, decisional Diffie-Hellman assumption in $\mathbb{G}_2$, and symmetric external Diffie-Hellman assumption, respectively.

1.3 Related Work

The revocation problem of IBE was first considered by Boneh and Franklin [BF01] with naive and non-scalable solutions, where the size of the key update generated by the KGC is linear in the number of users. Boldyreva et al. [BGK08] utilized a subset cover framework called the complete subtree method [NNL01] and proposed the first RIBE scheme with scalable revocation by reducing the size of the key update logarithmic in the number of users. Libert and Vergnaud [LV09] constructed the first adaptively secure RIBE scheme. In addition, Seo and Emura [SE13b] defined the notion of the DKER, and then proposed the first adaptively secure scheme with the DKER. Note that the above schemes are pairing-based, and various pairing-based improvements [GW19, ISW17, LLP17, SZSM17, WLXZ14, WES17] have been proposed. The RIBE scheme is also constructed under the LWE assumption [CLL$^+$12, TW17a, TW17b], code-based assumption [CCKS18], and CDH assumption without pairing or the factoring assumption [HLCL18] although these schemes do not satisfy the DKER. However, Katsumata et al. [KMT19] proposed a generic construction of RIBE with the DKER from RIBE without the DKER and 2-level HIBE. Since IBE implies selectively secure HIBE [DG17], RIBE without the DKER implies selectively secure RIBE with the DKER. In addition, by extending the concept presented by Katsumata et al., several generic constructions of RIBE with the DKER have been proposed. For example, Ma and Lin [ML19a, ML19b] proposed a generic construction of RIBE with the DKER from 2-level HIBE, and Lee [Lee19] proposed a generic construction of RIBE with the DKER from 2-level HIBE and identity-based revocation. However, these schemes suffer from large ciphertexts.

The concept of RHIBE was first discussed by Seo and Emura [SE13a]; however, their definition was too weak for practical application since it does not satisfy collusion resistance that is the minimum security requirement of HIBE. Thus, Seo and Emura [SE15b] redefined the notion with the DKER and insider security that ensures collusion resistance. Then, several selectively secure pairing-based RHIBE schemes were proposed [ESY16, LP18, RLPL15]. Note that several papers have claimed to construct adaptively secure RHIBE schemes [Lee16, SE15a, WLJW16, XWW$^+$16, XWWT18]; however, their security proofs are incorrect or ignore insider security. Katsumata et al. [KMT19] defined the most strict and rigorous definition of RHIBE and introduced a stronger definition for the DKER. In addition, Katsumata et al. proposed a selectively secure RHIBE scheme from the LWE assumption, and Wang et al. [WZH$^+$19] proposed a more efficient selectively secure scheme in the standard model and adaptively secure scheme in the random oracle model under the same assumption.

1.4 Organization

The remainder of this paper is organized as follows. Section 2 gives an overview of the proposed construction. In Section 3, we provide a definition of RHIBE, and in Section 4, we introduce the additional properties of RHIBE, which are used to construct RHIBE. Section 5 describes the construction of RHIBE, and, in Section 6, we prove the adaptive security of the scheme.

2 Technical Overview

In this section, we provide a technical overview of our result. We first present an overview of the definition of RHIBE and the complete subtree (CS) method [NNL01] Section 2.1, which is a popular technique to efficiently achieve revocation functionality. We then present the selectively secure RHIBE scheme with the weaker DKER of Seo-Emura (SE) [SE15b] and an overview of its security proof in Section 2.2. Then, we provide a proof of the adaptive security of Watanabe et al.'s (WES) RIBE with the DKER and discuss the difficulties related to combining them to construct adaptively secure RHIBE in Section 2.3.[4] In Section 2.4, we present the proposed approach to construct adaptively secure RHIBE with the weaker DKER. We then discuss how the scheme is modified to achieve the stronger DKER.

2.1 Preliminaries

Notations. Let $\mathbb{N}$ be the set of all natural numbers. For non-negative integers $a, b \in \mathbb{N}$ with $a \leq b$, we define $[a, b] := \{a, a+1, \ldots, b\}$ and $[a] := [1, a]$. In addition, as a special case, $[a, b] = \emptyset$ for $a > b$. Let lowercase and uppercase bold letters $\mathbf{a}$ and $\mathbf{A}$ denote a column vector and a matrix, respectively, where $\mathbf{a}^\top$ and $\mathbf{A}^\top$ denote their transposes. For a finite set S, let $x \leftarrow_R S$ denote sampling x from S uniformly at random. For two algorithms $\mathsf{A}(\cdot)$ and $\mathsf{B}(\cdot)$, let $\mathsf{A}(\cdot) \approx \mathsf{B}(\cdot)$ denote that their outputs follow the same distribution.

Let $\mathcal{T}$ denote a time period space. According to convention, the size of $\mathcal{T}$ is polynomially bounded by the security parameter λ. Let ℓ-dimensional vector $\mathtt{ID} := (\mathtt{id}_1, \ldots, \mathtt{id}_\ell)$ denote an

[4] Recall that the difference between the weaker and the stronger DKER only appears in the hierarchical setting. Thus, the approach proposed by Watanabe et al. does not provide a pathway to achieve the stronger DKER.

identity at level-ℓ, and let $\mathcal{I}$ be an identity space of level-1, which is determined by only the security parameter λ; therefore, an identity space at level-ℓ is $\mathcal{I}^{\ell}$. Here, we use $|\mathtt{ID}| := \ell$ to denote the hierarchical level of the identity. For convenience, we consider kgc as a "root" user and let $\mathcal{I}^0 := \{\mathsf{kgc}\}$. In addition, $\mathcal{I}_{\mathtt{ID}} \subset \mathcal{I}^{|\mathtt{ID}|+1}$ denotes a set of level-($|\mathtt{ID}| + 1$) identities whose direct ancestor is $\mathtt{ID}$. In other words, $\mathcal{I}_{\mathtt{ID}} := \{\mathtt{ID}' \in \mathcal{I}^{|\mathtt{ID}|+1} \mid \forall \mathtt{id}_{|\mathtt{ID}|+1} \in \mathcal{I}, \mathtt{ID}' = (\mathtt{ID}, \mathtt{id}_{|\mathtt{ID}|+1})\}$. We also define several notations for the prefix of an identity $\mathtt{ID} = (\mathtt{id}_1, \cdots, \mathtt{id}_{|\mathtt{ID}|})$ in the following. For a non-negative integer $\ell \leq |\mathtt{ID}|$, an ℓ-dimensional prefix of $\mathtt{ID}$ is denoted $\mathtt{ID}_{[\ell]} := (\mathtt{id}_1, \ldots, \mathtt{id}_\ell)$. Here, a direct ancestor of $\mathtt{ID}$ is denoted $\mathsf{pa}(\mathtt{ID}) := \mathtt{ID}_{[|\mathtt{ID}|-1]}$, and $\mathtt{ID}_{[0]} := \mathsf{kgc}$. In addition, $\mathsf{prefix}^+(\mathtt{ID}) := \{\mathtt{ID}_{[1]}, \mathtt{ID}_{[2]}, \ldots, \mathtt{ID}_{|\mathtt{ID}|-1}(= \mathsf{pa}(\mathtt{ID})), \mathtt{ID}\}$ denotes a set of all prefixes of $\mathtt{ID}$ and itself.

Overview of RHIBE. Here, we provide an overview of RHIBE, which appears to be complicated for beginners. Note that this complexity stems from the existence of three types of keys, i.e., *secret keys* $\mathsf{sk}_{\mathtt{ID}}$, *key updates* $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$, and *decryption keys* $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$. In addition, a secret key contains a *delegation key* $\mathsf{delk}_{\mathtt{ID}} \in \mathsf{sk}_{\mathtt{ID}}$ that is responsible for key delegation in our description, although a delegation key $\mathsf{delk}_{\mathtt{ID}}$ does not explicitly appear in the syntax. Furthermore, a delegation key $\mathsf{delk}_{\mathtt{ID}}$ is updated during system execution.

At the time of RHIBE system launch, the KGC creates a master public key MPK and KGC's secret key $\mathsf{sk}_{\mathsf{kgc}}$. The ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$ depends on the receiver's identity $\mathtt{ID} \in \mathcal{I}^{\leq L}$ and time period $\mathtt{T} \in \mathcal{T}$, where L denotes the maximum level of the hierarchy. Here, the configuration of level-1 users $\mathtt{ID} \in \mathcal{I}$ is the same as that of RIBE. All level-1 users $\mathtt{ID}$ are given their secret key $\mathsf{sk}_{\mathtt{ID}}$ by the KGC, and the secret keys $\mathsf{sk}_{\mathtt{ID}}$ are insufficient to decrypt ciphertexts $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$. The KGC manages a revocation list $\mathsf{RL}_{\mathsf{kgc},\mathtt{T}} \subset \mathcal{I}$ of level-1 users who will be revoked at time period $\mathtt{T}$. Then, at each time period $\mathtt{T}$, the KGC broadcasts key update $\mathsf{ku}_{\mathsf{kgc},\mathtt{T}}$ for level-1 users. Here, by combining secret keys $\mathsf{sk}_{\mathtt{ID}}$ and key update $\mathsf{ku}_{\mathsf{kgc},\mathtt{T}}$, only non-revoked users are able to derive decryption keys $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ that can decrypt ciphertexts $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$.

The basic configuration of level-ℓ users $\mathtt{ID} \in \mathcal{I}^{\ell}$ for $\ell \geq 2$ is essentially the same. Here, all level-ℓ users $\mathtt{ID}$ are given their secret key $\mathsf{sk}_{\mathtt{ID}}$ by their level-($\ell - 1$) parent users $\mathsf{pa}(\mathtt{ID})$. For this purpose, parent users $\mathsf{pa}(\mathtt{ID})$ may update their delegation keys $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID})}$, which are parts of the secret keys, by themselves. Note that the secret keys $\mathsf{sk}_{\mathtt{ID}}$ themselves are insufficient to decrypt ciphertexts $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$. The parent users $\mathsf{pa}(\mathtt{ID})$ manage the revocation lists $\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} \subset \mathcal{I}_{\mathsf{pa}(\mathtt{ID})}$ of their children users who will be revoked at time period $\mathtt{T}$. Then, at each time period $\mathtt{T}$, parent users $\mathsf{pa}(\mathtt{ID})$ attempt to broadcast key update $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ for their children users. In this case, parent users $\mathsf{pa}(\mathtt{ID})$ can derive key updates $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ only when they are not revoked by their parent users $\mathsf{pa}(\mathsf{pa}(\mathtt{ID}))$ in the same time period. Note that parent users $\mathsf{pa}(\mathtt{ID})$ may update their delegation keys $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID})}$ by themselves to create key updates $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$. Given the parent user $\mathsf{pa}(\mathtt{ID})$'s key update $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$, only non-revoked users $\mathtt{ID}$ can derive decryption keys $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ by combining their secret keys $\mathsf{sk}_{\mathtt{ID}}$ and key updates $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ broadcast by parent users $\mathsf{pa}(\mathtt{ID})$.

In the following, we briefly describe the security model. Here, let $\mathtt{ID}^\star$ and $\mathtt{T}^\star$ denote the challenge identity and challenge time period, respectively. An RHIBE adversary can receive all $\mathsf{sk}_{\mathtt{ID}}$ (which contains $\mathsf{delk}_{\mathtt{ID}}$) and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ under the condition that they cannot derive $\mathsf{dk}_{\mathtt{ID}^\star,\mathtt{T}^\star}$. Here, all such secret keys include $\mathsf{sk}_{\mathtt{ID}}$ for $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star)$ as opposed to non-revocable HIBE. Similarly, all such key updates include $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ for $\mathsf{pa}(\mathtt{ID}) \in \mathsf{prefix}^+(\mathtt{ID}^\star) \land \mathtt{T} = \mathtt{T}^\star$. To prevent the adversary from deriving $\mathsf{dk}_{\mathtt{ID}^\star,\mathtt{T}^\star}$, if an adversary receives $\mathsf{sk}_{\mathtt{ID}^\star_{[\ell]}}$ for some $\ell \in [|\mathtt{ID}^\star|]$, then the identity $\mathtt{ID}^\star_{[\ell]}$ or one of its ancestors must be revoked by $\mathtt{T}^\star$. In addition to $\mathsf{sk}_{\mathtt{ID}}$ and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$, the adversary in the weaker/stronger DKER model can also receive $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$. In the weaker DKER model, the adversary can receive all $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ except for $\mathtt{ID} \in (\mathtt{ID}^\star_{[\ell]})_{\ell \in [|\mathtt{ID}^\star|]} \land \mathtt{T} = \mathtt{T}^\star$. In addition, the adversary in the stronger DKER model can receive all $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$, with the exception of $(\mathtt{ID}, \mathtt{T}) = (\mathtt{ID}^\star, \mathtt{T}^\star)$. Thus, compared to the weaker DKER model, the adversary in the stronger DKER model can receive

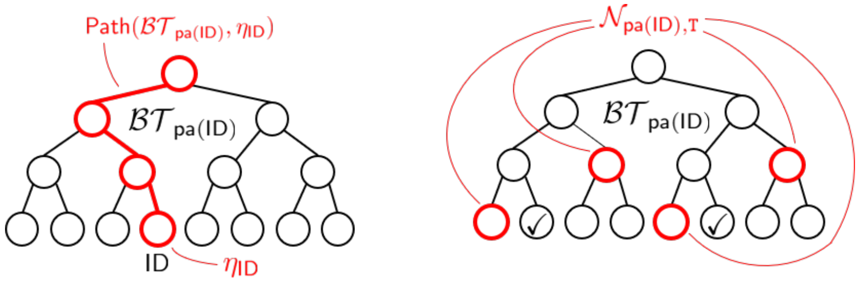
$\mathsf{Path}(\mathcal{BT}_{\mathsf{pa(ID)}}, \eta_{\mathsf{ID}})$
$\mathcal{BT}_{\mathsf{pa(ID)}}$
ID
η_{ID}
$\mathcal{N}_{\mathsf{pa(ID),T}}$
$\mathcal{BT}_{\mathsf{pa(ID)}}$
✓
✓

Figure 1: Diagrams of $\mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})$ and $\mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ for the $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ with 8 leaves. "✓" in leaf nodes means that an identity $\mathtt{ID}'$ assigned to the leaf node is revoked, i.e., $\mathtt{ID}' \in \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$.

additional decryption keys $(\mathsf{dk}_{\mathtt{ID}^{\star}_{[\ell]},\mathtt{T}^{\star}})_{\ell\in[|\mathtt{ID}^{\star}|-1]}$.

Complete Subtree Method. In our scheme, all parent users $\mathsf{pa}(\mathtt{ID}) \in \mathcal{I}^{\leq L-1}$, including the KGC, manage their own binary trees $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ and assign their child $\mathtt{ID} \in \mathcal{I}_{\mathsf{pa}(\mathtt{ID})}$ to a distinct leaf node denoted $\eta_{\mathtt{ID}}$. Here, let $\mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})$ denote a path from the root node to a leaf node $\eta_{\mathtt{ID}}$. The CS method ensures that there is an efficient algorithm that can output a set of nodes $\mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} \subseteq \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ by taking assigned leaves $(\eta_{\mathtt{ID}})_{\mathtt{ID}\in\mathcal{I}_{\mathsf{pa}(\mathtt{ID})}}$ of $\mathsf{pa}(\mathtt{ID})$'s child users and a set of child users $\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} \subseteq \mathcal{I}_{\mathsf{pa}(\mathtt{ID})}$ from which a member will be revoked as input. The set $\mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ satisfies the following properties.

- If $\mathtt{ID} \in \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$, $\mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}}) \cap \mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} = \emptyset$.
- If $\mathtt{ID} \notin \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$, $\mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}}) \cap \mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} \neq \emptyset$.

Figure 1 shows diagrams of $\mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})$ and $\mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$.

2.2 Selectively Secure RHIBE Scheme

Here, we present an overview of SE RHIBE and its security proof for selective security.

Construction. The SE RHIBE uses a symmetric bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ of prime order p. Here, g denotes the generator of $\mathbb{G}$. First, we present master public key MPK, ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$, and decryption key $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$.

$$\begin{aligned}
&\mathsf{MPK} = (g, g_1 := g^{\alpha}, g_2, (h_i)_{i\in[L+1]}, (h'_i)_{i\in[2]}), \qquad \mathsf{sk}_{\mathsf{kgc}} = (g_2^{\alpha}, \mathsf{delk}_{\mathsf{kgc}}),\\
&\qquad // \ (g_2, (h_i)_{i\in[L+1]}, (h'_i)_{i\in[2]}) \leftarrow_R \mathbb{G}^{L+4}, \quad \alpha \leftarrow_R \mathbb{Z}_p\\
&\mathsf{ct}_{\mathtt{ID},\mathtt{T}} = \left(g^s, (h_1^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+1})^s, ((h'_1)^{\mathtt{T}} h'_2)^s, \mathsf{M} \cdot e(g_1, g_2)^s\right), \qquad // \ s \leftarrow_R \mathbb{Z}_p\\
&\mathsf{dk}_{\mathtt{ID},\mathtt{T}} = \left(g_2^{\alpha} \cdot (h_1^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+1})^r \cdot ((h'_1)^{\mathtt{T}} h'_2)^t, g^r, g^t, (h_i^r)_{i\in[|\mathtt{ID}|+1,L]}\right). \qquad // \ (r,t) \leftarrow_R \mathbb{Z}_p^2
\end{aligned}$$

The description of the KGC's delegation key $\mathsf{delk}_{\mathsf{kgc}}$ will be given later. Here, we refer to $\mathsf{MSK} = g_2^{\alpha}$ a master secret key. When we write $\mathsf{ct}_{\mathtt{ID},\mathtt{T}} = (c_1, c_2, c_3, c_T)$ and $\mathsf{dk}_{\mathtt{ID},\mathtt{T}} = (dk_1, dk_2, dk_3, (dk'_i)_{i\in[|\mathtt{ID}|+1,L]})$, we can recover a plaintext M by computing $c_T \cdot e(c_2, dk_2) \cdot e(c_3, dk_3)/e(c_1, dk_1)$. Generally, the scheme is a concatenation of the Boneh-Boyen-Goh (BBG) HIBE [BBG05] and Boneh-Boyen (BB) IBE [BB04] with the same generator g, where the former and latter are used to encode an identity $\mathtt{ID}$ and time period $\mathtt{T}$, respectively.

Next, we present secret key $\mathsf{sk}_{\mathtt{ID}}$ and key update $\mathsf{ku}_{\mathtt{T}}$:

$$\mathsf{sk}_{\mathtt{ID}} = \left(\left(g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}} \cdot (h_1^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+1})^{r_{\mathtt{ID},\theta}}, g^{r_{\mathtt{ID},\theta}}, (h_i^{r_{\mathtt{ID},\theta}})_{i\in[|\mathtt{ID}|+1,L]}\right)_{\theta\in\mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})},\eta_{\mathtt{ID}})}, \mathsf{delk}_{\mathtt{ID}}\right),$$

$$// \; r_{\mathtt{ID},\theta} \leftarrow_R \mathbb{Z}_p$$

$$\mathsf{delk}_{\mathsf{pa}(\mathtt{ID})} = \left(g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}\right)_{\theta \in \mathcal{AN}_{\mathsf{pa}(\mathtt{ID})}}, \qquad // \; \alpha_{\mathsf{pa}(\mathtt{ID}),\theta} \leftarrow_R \mathbb{Z}_p$$

$$\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} = \left(\begin{array}{c} g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}} \cdot (h_1^{\mathtt{id}_1} \cdots h_{|\mathsf{pa}(\mathtt{ID})|}^{\mathtt{id}_{|\mathsf{pa}(\mathtt{ID})|}} h_{L+1})^{r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}} \cdot ((h'_1)^{\mathtt{T}} h'_2)^{t_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}}, \\ g^{r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}}, g^{t_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}}, \; (h_i^{r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}})_{i \in [|\mathsf{pa}(\mathtt{ID})|+1, L]} \end{array} \right)_{\theta \in \mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}},$$

$$// \; (r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}, t_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}) \leftarrow_R \mathbb{Z}_p^2$$

where $\mathcal{AN}_{\mathsf{pa}(\mathtt{ID})} \subset \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ is a set of all activated nodes θ. Here, the activated nodes denote nodes θ associated with $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$ used to create all $\mathsf{sk}_{\mathtt{ID},\theta}$ and/or $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$ at least once.[5] Each $\mathsf{sk}_{\mathtt{ID}}$ comprises sub-secret keys $\mathsf{sk}_{\mathtt{ID},\theta}$ associated with nodes θ in their parent's binary tree $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ and sub-delegation keys $\mathsf{delk}_{\mathtt{ID},\theta}$ associated with nodes θ in their own binary tree $\mathcal{BT}_{\mathtt{ID}}$. Similarly, each $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ comprises sub-key updates $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$ associated with nodes θ in $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$.

Correctness. We confirm that the above scheme satisfies correctness as follows.

- All parent users $\mathsf{pa}(\mathtt{ID})$ can compute $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID})}$ by themselves and can compute $\mathsf{sk}_{\mathtt{ID}}$ using $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID})}$.

- When non-revoked users $\mathtt{ID}$ are given $\mathsf{sk}_{\mathtt{ID}}$ and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$, they can compute $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$. Recall that the CS method ensures that there is a node $\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}}) \cap \mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ for non-revoked users $\mathtt{ID} \notin \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$. Specifically, when we have $\mathsf{sk}_{\mathtt{ID},\theta} = (sk_1, sk_2, sk'_{|\mathtt{ID}|+1}, \ldots, sk'_L)$ and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta} = (ku_1, ku_2, ku_3, ku'_{|\mathsf{pa}(\mathtt{ID})|+1} = ku'_{|\mathtt{ID}|}, \ldots, ku'_L)$, users $\mathtt{ID}$ sample $(r', t') \leftarrow_R \mathbb{Z}_p^2$ and can compute $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ as follows:

$$\begin{aligned} dk_1 &= sk_1 \cdot ku_1 \cdot (ku'_{|\mathtt{ID}|})^{\mathtt{id}_{|\mathtt{ID}|}} \cdot (h_1^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+1})^{r'} \cdot ((h'_1)^{\mathtt{T}} h'_2)^{t'} \\ &= g_2^{\alpha + \alpha_{\mathsf{pa}(\mathtt{ID}),\theta} - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}} \cdot (h_1^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+1})^{r_{\mathtt{ID},\theta} + r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta} + r'} \cdot ((h'_1)^{\mathtt{T}} h'_2)^{t_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta} + t'} \\ &= g_2^{\alpha} \cdot (h_1^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+1})^{r} \cdot ((h'_1)^{\mathtt{T}} h'_2)^{t} \end{aligned}$$

and

$$dk_2 = sk_2 \cdot ku_2 \cdot g^{r'} = g^{r_{\mathtt{ID},\theta} + r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta} + r'} = g^r, \qquad dk_3 = ku_3 \cdot g^{t'} = g^{t_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta} + t'} = g^t$$
$$dk'_i = sk'_i \cdot ku'_i \cdot h_i^{r'} = h_i^{r_{\mathtt{ID},\theta} + r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta} + r'} = h_i^r \quad \text{for } i \in [|\mathtt{ID}| + 1, L].$$

- When non-revoked users $\mathtt{ID}$ are given $\mathsf{sk}_{\mathtt{ID}}$ and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$, they can compute $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$. Specifically, users $\mathtt{ID}$ derive $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$, sample $(r'_{\theta'}, t'_{\theta'}) \leftarrow_R \mathbb{Z}_p^2$, and then compute for each $\theta' \in \mathcal{N}_{\mathtt{ID},\mathtt{T}}$,

$$\begin{aligned} ku_1 &= \mathsf{delk}_{\mathtt{ID},\theta'}^{-1} \cdot dk_1 \cdot (h_1^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+1})^{r'_{\theta'}} \cdot ((h'_1)^{\mathtt{T}} h'_2)^{t'_{\theta'}} \\ &= g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}} \cdot (h_1^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+1})^{r + r'_{\theta'}} \cdot ((h'_1)^{\mathtt{T}} h'_2)^{t + t'_{\theta'}} \\ &= g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}} \cdot (h_1^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+1})^{r_{\mathtt{ID},\mathtt{T},\theta'}} \cdot ((h'_1)^{\mathtt{T}} h'_2)^{t_{\mathtt{ID},\mathtt{T},\theta'}} \end{aligned}$$

and

$$ku_2 = dk_2 \cdot g^{r'_{\theta'}} = g^{r + r'_{\theta'}} = g^{r_{\mathtt{ID},\mathtt{T},\theta'}}, \qquad ku_3 = dk_3 \cdot g^{t'_{\theta'}} = g^{t + t'_{\theta'}} = g^{t_{\mathtt{ID},\mathtt{T},\theta'}},$$
$$ku'_i = dk'_i \cdot h_i^{r'_{\theta'}} = h_i^{r + r'_{\theta'}} = h_i^{r_{\mathtt{ID},\mathtt{T},\theta'}} \quad \text{for } i \in [|\mathtt{ID}| + 1, L].$$

[5] In other words, each parent user $\mathsf{pa}(\mathtt{ID})$ updates $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID})} = (\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{AN}_{\mathsf{pa}(\mathtt{ID})}}$ by adding $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta'}$ when they create $\mathsf{sk}_{\mathtt{ID},\theta'}$ and/or $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta'}$ for $\theta' \notin \mathcal{AN}_{\mathsf{pa}(\mathtt{ID})}$. This procedure also updates $\mathcal{AN}_{\mathsf{pa}(\mathtt{ID})} \leftarrow \mathcal{AN}_{\mathsf{pa}(\mathtt{ID})} \cup \{\theta'\}$.

Security. Intuitively, the scheme is secure because revoked users $\mathtt{ID}$ do not appear to be able to compute dk_1 correctly because the randomness $\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}$ in the exponents between $\mathsf{sk}_{\mathtt{ID}}$ and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ do not vanish. Seo and Emura formally proved *selective* security by reducing the security of the BBG HIBE [BBG05] to that of SE RHIBE. Given the BBG master public key $(g, g_1, g_2, (h_i)_{i\in[L+1]})$, the reduction algorithm creates the BB master public key $(h'_i)_{i\in[2]}$ with the BB trapdoor, which allows the reduction algorithm to create BB secret key for $\mathtt{T} \neq \mathtt{T}^\star$ in the same manner as the security proof [BB04]. Note that the reduction algorithm does not know the master secret key g_2^α, i.e., the BBG maser secret key.

SE *Node Division.* The most crucial point of the proof is *node division.* Specifically, the reduction algorithm creates $\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$, and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$ in distinct ways depending on the division. Here, let $\mathcal{AN}$ denote a set of all activated nodes that appear during the security game, i.e., $\mathcal{AN}$ is the union of $\mathcal{AN}_{\mathsf{pa}(\mathtt{ID})}$ for all parent users $\mathsf{pa}(\mathtt{ID})$, including kgc, that appear during the game. Let $(\mathtt{ID}^\star, \mathtt{T}^\star)$ be the target identity and target time period that the reduction algorithm knows at the beginning of the security game. Let $\ell^\star$ be the minimum hierarchical level of the target user's ancestor whose secret key $\mathsf{sk}_{\mathtt{ID}^\star_{[\ell^\star]}}$ is received by the RHIBE adversary.[6] To prevent trivial attacks, the adversary must not be able to receive $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell]}),\mathtt{T}^\star,\theta}$ if it shares the same θ with $\mathsf{sk}_{\mathtt{ID}^\star_{[\ell]},\theta}$ for $\ell \in [\ell^\star, |\mathtt{ID}^\star|]$.[7] Here, the user $\mathtt{ID}^\star_{[\ell^\star]}$ (or one of its ancestors) must be revoked by time period $\mathtt{T}^\star$. At the beginning of the game, the reduction algorithm guesses the $\ell^\star$ value and determines leaf nodes $\eta_{\mathtt{ID}^\star_{[\ell]}} \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell]})}$ for $\ell \in [\ell^\star]$ to which $\ell^\star$ users $\mathtt{ID}^\star_{[1]}, \mathtt{ID}^\star_{[2]}, \ldots, \mathtt{ID}^\star_{[\ell^\star]}$ will be assigned. Then, the reduction algorithm divides $\mathcal{AN}$ into three mutually exclusive subsets $\mathcal{SE}^{(1)}_{\ell^\star}, \mathcal{SE}^{(2)}_{\ell^\star}$, and $\mathcal{SE}^{(3)}_{\ell^\star}$ as follows:

$$\mathcal{SE}^{(1)}_{\ell^\star} := \left\{\theta : \left(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell]})} \quad \text{for } \ell \in [\ell^\star - 1]\right) \vee \left(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell^\star]})} \setminus \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell^\star]})}, \eta_{\mathtt{ID}^\star_{[\ell^\star]}})\right)\right\},$$

$$\mathcal{SE}^{(2)}_{\ell^\star} := \left\{\theta : \left(\theta \in \mathcal{BT}_{\mathtt{ID}^\star_{[\ell]}} \quad \text{for } \ell \in [\ell^\star, |\mathtt{ID}^\star|]\right) \vee \left(\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell^\star]})}, \eta_{\mathtt{ID}^\star_{[\ell^\star]}})\right)\right\},$$

$$\mathcal{SE}^{(3)}_{\ell^\star} := \mathcal{AN} \setminus (\mathcal{SE}^{(1)}_{\ell^\star} \cup \mathcal{SE}^{(2)}_{\ell^\star}) = \mathcal{AN} \setminus \left(\bigcup_{\ell=0}^{|\mathtt{ID}^\star|} \mathcal{BT}_{\mathtt{ID}^\star_{[\ell]}}\right).$$

Note that $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell^\star]})} = \mathcal{BT}_{\mathtt{ID}^\star_{[\ell^\star-1]}}$. A graphical overview of SE node division is presented in Appendix A (Figure 2). Note that the reduction algorithm can perform this division only when it knows $\mathtt{ID}^\star$. In addition, all $\mathsf{sk}_{\mathtt{ID}^\star_{[\ell]},\theta}$ for $\ell \in [\ell^\star - 1]$ and $\ell \in [\ell^\star, |\mathtt{ID}^\star|]$ are associated with nodes $\theta \in \mathcal{SE}^{(1)}_{\ell^\star}$ and $\theta \in \mathcal{SE}^{(2)}_{\ell^\star}$, respectively. We summarize the properties of the node division as follows.

- All $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{SE}^{(1)}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$. The adversary receives no $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta\in\mathcal{SE}^{(1)}_{\ell^\star}}$. The adversary may receive $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{SE}^{(1)}_{\ell^\star}}$ for $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star) \wedge \mathtt{T} = \mathtt{T}^\star$.
- All $(\mathsf{ku}_{\mathtt{ID},\mathtt{T},\theta})_{\theta\in\mathcal{SE}^{(2)}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{T} \neq \mathtt{T}^\star$. The adversary may receive $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta\in\mathcal{SE}^{(2)}_{\ell^\star}}$ for $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star)$.

[6] Note that there may be an adversary that does not receive $\mathsf{sk}_{\mathtt{ID}}$ for any $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star)$. Here, we ignore such adversaries and assume that an adversary always receives $\mathsf{sk}_{\mathtt{ID}}$ for some $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star)$.

[7] Otherwise, the adversary can create $\mathsf{dk}_{\mathtt{ID}^\star,\mathtt{T}^\star}$ from $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell]}),\mathtt{T}^\star,\theta}$ and $\mathsf{sk}_{\mathtt{ID}^\star_{[\ell]},\theta}$. Note that, according to the definition of $\ell^\star$, the adversary can receive $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell]}),\mathtt{T}^\star,\theta}$ even if it shares the same θ with $\mathsf{sk}_{\mathtt{ID}^\star_{[\ell]},\theta}$ for $\ell \in [\ell^\star - 1]$ because the adversary does not know the secret keys.

- All $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{SE}^{(3)}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{ID} \notin \mathsf{prefix}^{+}(\mathtt{ID}^\star)$.

Here, we do not analyze why the properties hold for the node division. Refer to [SE15b] for the corresponding analysis.

Key Creations. Then, the reduction algorithm changes the distribution of $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{SE}^{(1)}_{\ell^\star}}$ as $g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$, which cannot be computed by the reduction algorithm. Although the reduction algorithm cannot compute $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta} = g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$, this is not a problem because the adversary cannot obtain $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{SE}^{(1)}_{\ell^\star}}$. Note that an adversary cannot detect the change because the distributions of $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ and $g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ are the same, where $\alpha_{\mathsf{pa}(\mathtt{ID}),\theta} \leftarrow_R \mathbb{Z}_p$. As a result, the $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ of $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{SE}^{(1)}_{\ell^\star}}$ and $g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ terms of $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{SE}^{(1)}_{\ell^\star}}$ are replaced by $g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ and $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$, respectively. Therefore, the reduction algorithm can create $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{SE}^{(1)}_{\ell^\star}}$ using $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ sampled by itself. To compute $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{SE}^{(1)}_{\ell^\star}}$, the reduction algorithm receives BBG secret keys $\left(g_2^{\alpha} \cdot (h_1^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+1})^r, g^r, (h_i^r)_{i \in [|\mathtt{ID}|+1, L]}\right)$ and modifies them using $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$. The reduction algorithm can receive the BBG secret key because all $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{SE}^{(1)}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{ID} \notin \mathsf{prefix}^{+}(\mathtt{ID}^\star)$. Here, $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{SE}^{(2)}_{\ell^\star} \cup \mathcal{SE}^{(3)}_{\ell^\star}}$ is not changed; thus, the reduction algorithm can create $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{SE}^{(2)}_{\ell^\star} \cup \mathcal{SE}^{(3)}_{\ell^\star}}$ in the same manner as the real scheme. The rest of information that the reduction algorithm should be able to create is $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{SE}^{(2)}_{\ell^\star}}$ and $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{SE}^{(3)}_{\ell^\star}}$. To broadcast $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{SE}^{(3)}_{\ell^\star}}$, the reduction algorithm proceeds in the same manner as $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{SE}^{(1)}_{\ell^\star}}$, i.e., it receives BBG secret keys and modifies them using $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$. As in the case of $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{SE}^{(1)}_{\ell^\star}}$, this operation is effective because all $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{SE}^{(3)}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{ID} \notin \mathsf{prefix}^{+}(\mathtt{ID}^\star)$. To broadcast $(\mathsf{ku}_{\mathtt{ID},\mathtt{T},\theta})_{\theta \in \mathcal{SE}^{(2)}_{\ell^\star}}$, the reduction algorithm first creates BB secret keys $\left(g_2^{\alpha} \cdot ((h'_1)^{\mathtt{T}} h'_2)^t, g^t\right)$ with the BB trapdoor and modifies them using $g_2^{\alpha_{\mathtt{ID},\theta}}$. The operation is effective because all $(\mathsf{ku}_{\mathtt{ID},\mathtt{T},\theta})_{\theta \in \mathcal{SE}^{(2)}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{T} \neq \mathtt{T}^\star$.

2.3 Adaptively Secure RIBE Scheme

Adaptively secure non-hierarchical RIBE schemes have been constructed in the same manner. For example, Watanabe et al. [WES17] constructed the WES RIBE by concatenating the modified Jutla-Roy (JR) IBE [JR17] and BB IBE [BB04] with the same generators (g_1, g_2) of asymmetric bilinear groups, where the former and latter are used to encode $\mathtt{ID}$ and time period $\mathtt{T}$, respectively. They proved adaptive security by reducing the security of the JR IBE to that of the WES RIBE.[8]. Although the BB IBE only achieves selective security, they exploited the fact that the time period space $\mathcal{T}$ is polynomially bounded, and their reduction algorithm guessed the target time period $\mathtt{T}^\star$ with reduction loss $|\mathcal{T}|$ at the beginning of the game. To perform node division without the knowledge of $\mathtt{ID}^\star$, the reduction algorithm guesses the number $Q^\star$ on which the RIBE adversary makes a secret key query on $\mathtt{ID}^\star$ with reduction loss Q, which is the number of the RIBE adversary's key queries, and assigns the $Q^\star$-th queried user to uniformly selected predetermined node $\eta^\star$. Here,

[8] Note that the original JR IBE, which is secure under the SXDH assumption, is not compatible with the reduction. Here, Watanabe et al. had to modify the scheme and rely on the non-standard variant of the DDH assumption.

the operation can divide all nodes in the same way as the SE node division for $L = 1$ and prove adaptive security in the same manner as the above selectively secure RHIBE.

Unfortunately, this approach is not scalable for hierarchical settings. Specifically, to prove the adaptive security of RIBE, the reduction loss Q is sufficient to divide all nodes because the reduction algorithm must select only a single predetermined leaf node $\eta_{\mathtt{ID}^\star}$ to which the target $\mathtt{ID}^\star$ will be assigned. In the hierarchical case, the reduction algorithm must select $\ell^\star$ predetermined leaf nodes $\eta_{\mathtt{ID}^\star_{[\ell]}} \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell]})}$ for $\ell \in [\ell^\star]$. In other words, the reduction algorithm must guess $\ell^\star$ numbers $Q^\star_1, \ldots, Q^\star_{\ell^\star}$ on which the RHIBE adversary makes secret key queries on $\mathtt{ID}^\star_{[1]}, \ldots, \mathtt{ID}^\star_{[\ell^\star]}$, respectively. Therefore, the approach results in reduction loss $Q^{\ell^\star}$. In other words, even if we replace the BBG HIBE of the SE RHIBE with an adaptively secure HIBE scheme, it appears to be difficult to achieve adaptive security with the proof techniques of SE RHIBE and WES RIBE in a straightforward manner. Thus, a new approach is required to prove the adaptive security of RHIBE.

2.4 Proposed Approach for Adaptively Secure RHIBE Scheme

Here, we provide an overview of the proposed RHIBE scheme. Note that the WES node division is not scalable in the hierarchical setting. First, we employ a new node division to prove the adaptive security of RHIBE. Then, we modify the SE RHIBE to obtain an adaptively secure RHIBE scheme with the weaker DKER because the scheme is not compatible with the new node division. Finally, we discuss how we achieve the stronger DKER.

Adaptive Node Division. As discussed in Section 2.2, SE node division does not work in the adaptive security setting because it requires $\mathtt{ID}^\star$. In addition, it appears that WES node division cannot be extended to the hierarchical setting. Here, we employ a new node division to prove the adaptive security of RHIBE. Specifically, as the SE RHIBE case, our reduction algorithm guesses $\ell^\star$, i.e., the minimum hierarchical level of the target user's ancestor whose secret key $\mathsf{sk}_{\mathtt{ID}^\star_{[\ell^\star]}}$ is received by the RHIBE adversary. In addition, as the WES RIBE case, our reduction algorithm guesses only a single number $Q^\star$ on which the RHIBE adversary makes a secret key query on $\mathtt{ID}^\star_{[\ell^\star]}$ with reduction loss Q. In other words, the guess is scalable because the reduction algorithm does not guess the numbers $Q^\star_1, \ldots, Q^\star_{\ell^\star-1}$ on which the RHIBE adversary makes a secret key query on $\mathtt{ID}^\star_{[1]}, \ldots, \mathtt{ID}^\star_{[\ell^\star-1]}$. Then, the reduction algorithm divides a set of all activated nodes $\mathcal{AN}$ into two mutually exclusive subsets $\mathcal{SK}_{\ell^\star}$ and $\mathcal{KU}_{\ell^\star}$, which are defined later. Unlike the SE RHIBE case, the reduction algorithm does not set leaf nodes $\eta_{\mathtt{ID}^\star_{[\ell]}} \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell]})}$ for $\ell \in [\ell^\star]$ in advance. In turn, whenever the reduction algorithm creates binary trees $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ of all level-$(\ell^\star - 1)$ users $\mathsf{pa}(\mathtt{ID})$ such that $|\mathtt{ID}| = \ell^\star$, it sets leaves $\eta^\star_{\mathsf{pa}(\mathtt{ID})} \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ to which the $Q^\star$-th queried user will be assigned in advance. Note that, although there are numerous (but polynomially many) leaves $\eta^\star_{\mathsf{pa}(\mathtt{ID})}$, only one of them will be used for the $Q^\star$-th query. Then, our node division is defined as follows:

$$\mathcal{SK}_{\ell^\star} := \left\{ \theta : \begin{array}{c} \big(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})} \quad \text{for} \quad |\mathtt{ID}| \le \ell^\star - 1\big) \vee \\ \Big(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})} \setminus \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta^\star_{\mathsf{pa}(\mathtt{ID})}) \quad \text{for} \quad |\mathtt{ID}| = \ell^\star\Big) \end{array} \right\},$$

$$\mathcal{KU}_{\ell^\star} := \mathcal{AN} \setminus \mathcal{SK}_{\ell^\star} = \left\{ \theta : \begin{array}{c} \big(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})} \quad \text{for} \quad |\mathtt{ID}| \ge \ell^\star + 1\big) \vee \\ \Big(\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta^\star_{\mathsf{pa}(\mathtt{ID})}) \quad \text{for} \quad |\mathtt{ID}| = \ell^\star\Big) \end{array} \right\}.$$

A graphical overview of our node division is presented in Appendix A (Figure 3). Note that the reduction algorithm can perform node division without knowledge of $\mathtt{ID}^\star$; thus, the division is compatible with the adaptive security setting. In particular, all $\mathsf{sk}_{\mathtt{ID}^\star_{[\ell]},\theta}$ for $\ell \in [\ell^\star - 1]$ and

$\ell \in [\ell^\star, |\mathtt{ID}^\star|]$ are associated with nodes $\theta \in \mathcal{SK}_{\ell^\star}$ and $\theta \in \mathcal{KU}_{\ell^\star}$, respectively. We summarize the property of the node division in the following.

- All $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$. The adversary may receive $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ for $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star) \wedge \mathtt{T} = \mathtt{T}^\star$.

- All $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star) \vee \mathtt{T} \neq \mathtt{T}^\star$. The adversary may receive $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ for $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star)$.

Here, we omit the analysis of why the properties hold (refer to Section 6 for the corresponding analysis). Intuitively, $\mathcal{SK}_{\ell^\star}$ and $\mathcal{KU}_{\ell^\star}$ take similar roles as $\mathcal{SE}^{(1)}_{\ell^\star}$ and $\mathcal{SE}^{(2)}_{\ell^\star} \cup \mathcal{SE}^{(3)}_{\ell^\star}$, respectively. In particular, all $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$ as in the case of $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{SE}^{(1)}_{\ell^\star}}$. Similarly, all $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star) \vee \mathtt{T} \neq \mathtt{T}^\star$, while all $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{SE}^{(2)}_{\ell^\star}}$ and $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{SE}^{(3)}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{T} \neq \mathtt{T}^\star$ and $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$, respectively. Although we omit the details, the reduction algorithm can create all $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ and all $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$ in the same manner as SE node division. However, a problem arises in $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ and $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$ due to the slight difference between node divisions; therefore, the modification of the node division is insufficient to prove the security of the SE RHIBE for two reasons. First, the RHIBE adversary receives no $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta\in\mathcal{SE}^{(1)}_{\ell^\star}}$; thus, the reduction algorithm for SE node division sets $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta} = g_2^{\alpha-\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}})_{\theta\in\mathcal{SE}^{(1)}_{\ell^\star}}$, which are elements that cannot be computed by the reduction algorithm itself. In addition, our node division allows the adversary to receive $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$; therefore, the reduction algorithm cannot set $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$ in the same manner as the SE node division. Thus, the reduction algorithm cannot answer $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ even in the selective security model. Second, the reduction algorithm for SE node division employs distinct methods to create $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$ depending on whether $\mathtt{T} \neq \mathtt{T}^\star$ or $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$ holds (i.e., whether $\theta \in \mathcal{SE}^{(2)}_{\ell^\star}$ or $\theta \in \mathcal{SE}^{(3)}_{\ell^\star}$). However, here, the reduction algorithm must know $(\mathtt{ID}^\star, \mathtt{T}^\star)$. In addition, $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$ satisfies $\mathtt{T} \neq \mathtt{T}^\star \vee \mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$; therefore, the reduction algorithm without knowing $(\mathtt{ID}^\star, \mathtt{T}^\star)$ does not have a way to distinguish which of condition $\mathtt{T} \neq \mathtt{T}^\star$ or $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$ holds. Thus, the reduction algorithm can only answer $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$ in the selective security model.[9] Therefore, the SE RHIBE is incompatible with our node division.

Modified RHIBE Scheme. To avoid the issue, we modify the SE RHIBE scheme. Here, we use BBG HIBE to explain the proposed approach because we believe it facilitates understanding of the proposed approach compared to SE RHIBE. Note that we extract the required properties of HIBE to construct adaptively secure RHIBE and replace the BBG HIBE with adaptively secure HIBE schemes later in this section. Although SE RHIBE is a concatenation of level-L BBG HIBE and BB IBE, the proposed RHIBE scheme is based on level-$(L+1)$ BBG HIBE and utilizes its algebraic property. First, we set $\mathsf{MPK}, \mathsf{sk}_{\mathsf{kgc}}, \mathsf{ct}_{\mathtt{ID},\mathtt{T}}$, and $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$, which are very similar to those of BBG HIBE, as follows:

$$\begin{aligned}
&\mathsf{MPK} = (g, g_1 := g^\alpha, g_2, (h_i)_{i\in[L+2]}), \qquad \mathsf{sk}_{\mathsf{kgc}} = (g_2^\alpha, \mathsf{delk}_{\mathsf{kgc}}),\\
&\qquad // \ (g_2, (h_i)_{i\in[L+2]}) \leftarrow_R \mathbb{G}^{L+3}, \quad \alpha \leftarrow_R \mathbb{Z}_p
\end{aligned}$$

[9] Note that we can avoid this issue by guessing $\mathtt{T}^\star$ with reduction loss $|\mathcal{T}|$ as the proofs of the above adaptively secure RIBE. However, our construction does not require the guess; thus, it avoids this issue. That is why the proposed RIBE scheme achieves tighter reduction than the WES RIBE [WES17] and its variant [GW19], as shown in Table 2.

$$\begin{aligned}\mathsf{ct}_{\mathtt{ID},\mathtt{T}} &= \Big(g^{s}, (h_1^{\mathtt{T}} h_2^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|+1}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+2})^{s}, \mathsf{M} \cdot e(g_1, g_2)^{s}\Big), \qquad // \ s \leftarrow_R \mathbb{Z}_p \\ \mathsf{dk}_{\mathtt{ID},\mathtt{T}} &= \Big(g_2^{\alpha} \cdot (h_1^{\mathtt{T}} h_2^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|+1}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+2})^{r}, g^{r}, (h_i^{r})_{i \in [|\mathtt{ID}|+2, L+1]}\Big). \qquad // \ r \leftarrow_R \mathbb{Z}_p^2\end{aligned}$$

Here, it is easy to verify that $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ enables users $\mathtt{ID}$ to decrypt $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$. Then, we describe $\mathsf{delk}_{\mathsf{kgc}}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID})}$, and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ as follows:

$$\mathsf{delk}_{\mathsf{kgc}} = \big(g_2^{\alpha_{\mathsf{kgc},\theta}}\big)_{\theta \in \mathcal{AN}_{\mathsf{kgc}}}, \qquad // \ \alpha_{\mathsf{kgc},\theta} \leftarrow_R \mathbb{Z}_p$$

$$\mathsf{sk}_{\mathtt{ID}} = \bigg(\Big(g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}} \cdot (h_2^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|+1}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+2})^{r_{\mathtt{ID},\theta}}, g^{r_{\mathtt{ID},\theta}}, (h_i^{r_{\mathtt{ID},\theta}})_{i \in \{1\} \cup [|\mathtt{ID}|+2, L+1]}\Big)_{\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})}, \mathsf{delk}_{\mathtt{ID}}\bigg),$$

$$// \ r_{\mathtt{ID},\theta} \leftarrow_R \mathbb{Z}_p$$

$$\mathsf{delk}_{\mathsf{pa}(\mathtt{ID})} = \Big(g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}} \cdot (h_2^{\mathtt{id}_1} \cdots h_{|\mathtt{ID}|+1}^{\mathtt{id}_{|\mathtt{ID}|}} h_{L+2})^{u_{\mathsf{pa}(\mathtt{ID}),\theta}}, g^{u_{\mathsf{pa}(\mathtt{ID}),\theta}}, (h_i^{u_{\mathsf{pa}(\mathtt{ID}),\theta}})_{i \in \{1\} \cup [|\mathtt{ID}|+2, L+1]}\Big)_{\theta \in \mathcal{AN}_{\mathsf{pa}(\mathtt{ID})}},$$

$$// \ (\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}, u_{\mathsf{pa}(\mathtt{ID}),\theta}) \leftarrow_R \mathbb{Z}_p^2$$

$$\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} = \begin{pmatrix} g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}} \cdot (h_1^{\mathtt{T}} h_2^{\mathtt{id}_1} \cdots h_{|\mathsf{pa}(\mathtt{ID})|+1}^{\mathtt{id}_{|\mathsf{pa}(\mathtt{ID})|}} h_{L+2})^{r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}}, \\ g^{r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}}, \ (h_i^{r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}})_{i \in [|\mathsf{pa}(\mathtt{ID})|+2, L]} \end{pmatrix}_{\theta \in \mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}}.$$

$$// \ r_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta} \leftarrow_R \mathbb{Z}_p$$

Here, $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$ of the proposed scheme follows a similar distribution to $\mathsf{sk}_{\mathtt{ID},\theta}$, and that of SE RHIBE follows a similar distribution to MSK. This is why the $\mathsf{sk}_{\mathtt{ID}}$ of the proposed scheme is larger than that of the existing RHIBE schemes. It is easy to check that the scheme satisfies the correctness: $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID})}$ can be created without any secret information, $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID})}$ enables users $\mathsf{pa}(\mathtt{ID})$ to create $\mathsf{sk}_{\mathtt{ID}}$, $\mathsf{sk}_{\mathtt{ID}}$ and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ enable non-revoked users $\mathtt{ID}$ to create $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ and $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$.

Key Creations. Next, we observe how our construction avoids the previous issue to prove adaptive security. As with Seo-Emura's proof, the reduction algorithm changes the distribution of factor $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ of $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{SK}_{\ell^\star}}$ to $g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$, which cannot be computed by the reduction algorithm *by itself*. As with the SE proof, here, the adversary cannot detect the change. Thus, the $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ terms of $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{SK}_{\ell^\star}}$ and $g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ of $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{SK}_{\ell^\star}}$ are replaced by $g_2^{\alpha - \alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ and $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$, respectively. Although we omit the specifics, the reduction algorithm can create all $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{SK}_{\ell^\star}}$ and all $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{KU}_{\ell^\star}}$ in the same manner as SE RHIBE. In addition, the reduction algorithm of the proposed scheme can also compute $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{SK}_{\ell^\star}}$ and $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{KU}_{\ell^\star}}$. First, the modification of $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$, which follows a distribution that is similar to $\mathsf{sk}_{\mathtt{ID},\theta}$, allows the reduction algorithm to create $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{SK}_{\ell^\star}}$. Here, the reduction algorithm receives a special form of BBG secret keys $\Big(g_2^{\alpha} \cdot (h_2^{\mathtt{id}_1} \cdots h_{|\mathsf{pa}(\mathtt{ID})|+1}^{\mathtt{id}_{|\mathsf{pa}(\mathtt{ID})|}} h_{L+2})^{r}, g^{r}, (h_i^{r})_{i \in \{1\} \cup [|\mathsf{pa}(\mathtt{ID})|+2, L+1]}\Big)$ of $\mathsf{pa}(\mathtt{ID})$, where $\mathsf{pa}(\mathtt{ID})$ is encoded in levels from 2 to $|\mathsf{pa}(\mathtt{ID})| + 1$, and modifies them using $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$. In this case, we exploit the fact that all $(\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{SK}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{ID} \notin \mathsf{prefix}^{+}(\mathtt{ID}^\star)$. Next, the modification of the proposed scheme, which is not a concatenation of BBG HIBE and BB IBE, i.e., it is based on only BBG HIBE, enables the reduction algorithm to create $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{KU}_{\ell^\star}}$. For this purpose, the reduction algorithm receives BBG secret keys $\Big(g_2^{\alpha} \cdot (h_1^{\mathtt{T}} h_2^{\mathtt{id}_1} \cdots h_{|\mathsf{pa}(\mathtt{ID})|+1}^{\mathtt{id}_{|\mathsf{pa}(\mathtt{ID})|}} h_{L+2})^{r}, g^{r}, (h_i^{r})_{i \in [|\mathsf{pa}(\mathtt{ID})|+2, L+1]}\Big)$ of $(\mathtt{T}, \mathsf{pa}(\mathtt{ID}))$, where $\mathtt{T}$ and $\mathsf{pa}(\mathtt{ID})$ are encoded in levels 1 and from 2 to $|\mathsf{pa}(\mathtt{ID})| + 1$, respectively, and modifies them using $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$. Here, we exploit the fact that all $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{KU}_{\ell^\star}}$ received by the RHIBE adversary satisfy $\mathtt{ID} \notin \mathsf{prefix}^{+}(\mathtt{ID}^\star) \vee \mathtt{T} \neq \mathtt{T}^\star$. Therefore, the reduction algorithm can create all $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID})}$, $\mathsf{sk}_{\mathtt{ID}}$, and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ received by the adversary in a security proof.

Achieving the Stronger DKER. The above scheme is sufficient to achieve the weaker DKER; however, it is insufficient to achieve the stronger DKER. Here, we describe how the reduction algorithm creates $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$. First, the reduction algorithm creates $\mathsf{sk}_{\mathtt{ID},\theta}$ and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$ with the same θ as described above and can derive *most* decryption keys $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$. Here, the only exception is $(\mathsf{dk}_{\mathtt{ID}^\star_{[\ell]},\mathtt{T}^\star})_{\ell\in[\ell^\star-1]}$. Recall that all $(\mathsf{sk}_{\mathtt{ID}^\star_{[\ell]},\theta})_{\ell\in[\ell^\star-1]}$ are associated with nodes $\theta\in\mathcal{SK}_{\ell^\star}$. As observed above, the reduction algorithm can create all $(\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ by itself. However, the method to create $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ relies on the condition $\mathtt{ID}\in\mathsf{prefix}^+(\mathtt{ID}^\star)$. In other words, the reduction algorithm cannot create $(\mathsf{sk}_{\mathtt{ID}^\star_{[\ell]},\theta})_{\ell\in[\ell^\star-1]}$. Thus, the reduction algorithm receives the decryption key as a BBG secret key $\left(g_2^{\alpha}\cdot(h_1^{\mathtt{T}}h_2^{\mathtt{id}_1}\cdots h_{|\mathtt{ID}|+1}^{\mathtt{id}_{|\mathtt{ID}|}}h_{L+2})^r, g^r, (h_i^r)_{i\in[|\mathtt{ID}|+2,L+1]}\right)$ of $(\mathtt{T},\mathtt{ID})$, where $\mathtt{T}$ and $\mathtt{ID}$ are encoded in levels 1 and from 2 to $|\mathtt{ID}|+1$, respectively. The reduction algorithm requires that $\mathtt{ID}\notin\mathsf{prefix}^+(\mathtt{ID}^\star)\vee\mathtt{T}\neq\mathtt{T}^\star$ to receive the BBG secret keys; thus, it can only answer $(\mathsf{dk}_{\mathtt{ID}^\star_{[\ell]},\mathtt{T}})_{\ell\in[\ell^\star-1]}$ when $\mathtt{T}\neq\mathtt{T}^\star$. Thus, the above scheme only achieves the weaker DKER because the reduction algorithm for the stronger DKER has to be able to create $(\mathsf{dk}_{\mathtt{ID}^\star_{[\ell]},\mathtt{T}^\star})_{\ell\in[\ell^\star-1]}$. Moreover, the above scheme has a trivial attack in the stronger DKER model since one of $(\mathsf{dk}_{\mathtt{ID}^\star_{[\ell]},\mathtt{T}^\star})_{\ell\in[\ell^\star-1]}$ enables an adversary to decrypt challenge ciphertext $\mathsf{ct}_{\mathtt{ID}^\star,\mathtt{T}^\star}$.

To prevent this trivial attack, we modify the decryption key as follows:

$$\mathsf{dk}_{\mathtt{ID},\mathtt{T}}=\left(g_2^{\alpha}\cdot(h_1^{\mathtt{T}}h_2^{\mathtt{id}_1}\cdots h_{|\mathtt{ID}|+1}^{\mathtt{id}_{|\mathtt{ID}|}}h_{L+2})^r, g^r\right)\qquad //\ r\leftarrow_R\mathbb{Z}_p^2.$$

Here, it is easy to check that the decryption key correctly decrypts a ciphertext. Although this is a simple modification by removing $(h_i)_{i\in[|\mathtt{ID}|+2,L+1]}$, it prevents an adversary with $(\mathsf{dk}_{\mathtt{ID}^\star_{[\ell]},\mathtt{T}^\star})_{\ell\in[\ell^\star-1]}$ from decrypting challenge ciphertext $\mathsf{ct}_{\mathtt{ID}^\star,\mathtt{T}^\star}$. Thus, the reduction algorithm receives the special BBG secret keys and can answer $(\mathsf{dk}_{\mathtt{ID}^\star_{[\ell]},\mathtt{T}^\star})_{\ell\in[\ell^\star-1]}$ in the stronger DKER model.

Extracting Essential Properties of HIBE Required for Proposed Approach. The above modification describes the proposed approach based on BBG HIBE. However, some readers may wonder whether the approach truly provides adaptively secure RHIBE schemes because we employed the special algebraic/security properties of BBG HIBE. Thus, we exploit essential properties, i.e., *MSK evaluatability*, *first-level wildcarded SK*, and *prefix decryption restriction*, which enable HIBE schemes to be compatible with the proposed approach. Then, we verify that not only the BBG HIBE but also several adaptively secure HIBE schemes [CG17, CW14, GCTC16] satisfy the properties. As a result, we can construct adaptively secure RHIBE schemes based on the HIBE schemes.

3 RHIBE

In this section, we provide a definition for RHIBE. Note that the content of this section is primarily based on [KMT19].

Syntax. An RHIBE scheme Π consists of six algorithms (Setup, $\mathsf{Encrypt}$, GenSK, KeyUp, GenDK, $\mathsf{Decrypt}$) defined as follows.

- $\mathsf{Setup}(1^\lambda, L)\to(\mathsf{MPK},\mathsf{sk}_{\mathsf{kgc}})$: This is the *setup* algorithm that takes security parameter 1^λ and the maximum depth of the hierarchy $L\in\mathbb{N}$ as input, and outputs a master public key MPK and the KGC's secret key $\mathsf{sk}_{\mathsf{kgc}}$.
- $\mathsf{Encrypt}(\mathsf{MPK},\mathtt{ID},\mathtt{T},\mathsf{M})\to\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$: This is the *encryption* algorithm, which takes MPK, an identity $\mathtt{ID}\in\mathcal{I}^{|\mathtt{ID}|}$, time period $\mathtt{T}\in\mathcal{T}$, and a plaintext $\mathsf{M}\in\mathcal{M}$ as input, and outputs a ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$.

- $\mathsf{GenSK}(\mathsf{MPK}, \mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}, \mathtt{ID}) \to (\mathsf{sk}_{\mathtt{ID}}, \mathsf{sk}'_{\mathsf{pa}(\mathtt{ID})})$: This is the *secret key generation* algorithm, which takes MPK, a parent's secret key $\mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}$, and an identity $\mathtt{ID} \in \mathcal{I}_{\mathsf{pa}(\mathtt{ID})}$ as input, and outputs $\mathsf{sk}_{\mathtt{ID}}$ for $\mathtt{ID}$ and the "updated" $\mathsf{sk}'_{\mathsf{pa}(\mathtt{ID})}$.
- $\mathsf{KeyUp}(\mathsf{MPK}, \mathtt{T}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{RL}_{\mathtt{ID},\mathtt{T}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}) \to (\mathsf{ku}_{\mathtt{ID},\mathtt{T}}, \mathsf{sk}'_{\mathtt{ID}})$: The *key update information generation* algorithm takes MPK, $\mathtt{T} \in \mathcal{T}$, $\mathsf{sk}_{\mathtt{ID}}$ for $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$, revocation list $\mathsf{RL}_{\mathtt{ID},\mathtt{T}} \subseteq \mathcal{I}_{\mathtt{ID}}$, and a parent's key update $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ as input, and outputs $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$ and the "updated" $\mathsf{sk}'_{\mathtt{ID}}$. As a special case, we define $\mathsf{ku}_{\mathsf{pa}(\mathsf{kgc}),\mathtt{T}} := \perp$ for all $\mathtt{T} \in \mathcal{T}$.
- $\mathsf{GenDK}(\mathsf{MPK}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}) \to \mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ or $\perp$: This is the *decryption key generation* algorithm, which takes MPK, $\mathsf{sk}_{\mathtt{ID}}$ for $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$, and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ as input, and outputs a decryption key $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ for $\mathtt{T} \in \mathcal{T}$ or the special symbol $\perp$, indicating that $\mathtt{ID}$ or some of its ancestors have been revoked.
- $\mathsf{Decrypt}(\mathsf{MPK}, \mathsf{dk}_{\mathtt{ID},\mathtt{T}}, \mathsf{ct}_{\mathtt{ID},\mathtt{T}}) \to \mathsf{M}$: This is the *decryption* algorithm, which takes MPK, $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$, and $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$ as input, and outputs the decryption result M.

Correctness. We require ciphertext $\mathsf{ct}_{\mathtt{ID},\mathtt{T}}$ to be decrypted properly by a correctly-generated decryption key $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ for the same $\mathtt{ID}$ and $\mathtt{T}$ when $\mathtt{ID}$ is not revoked at $\mathtt{T}$. In other words, for all $\lambda \in \mathbb{N}$, $L \in \mathbb{N}$, $(\mathsf{MPK}, \mathsf{sk}_{\mathsf{kgc}}) \leftarrow \mathsf{Setup}(1^\lambda, L)$, $\ell \in [L]$, $\mathtt{ID} \in (\mathcal{I})^\ell$, $\mathtt{T} \in \mathcal{T}$, $\mathsf{M} \in \mathcal{M}$, $\mathsf{RL}_{\mathsf{kgc},\mathtt{T}} \subseteq \mathcal{I}$, $\mathsf{RL}_{\mathtt{ID}_{[1]},\mathtt{T}} \subseteq \mathcal{I}_{\mathtt{ID}_{[1]}}, \ldots, \mathsf{RL}_{\mathtt{ID}_{[\ell-1]},\mathtt{T}} \subseteq \mathcal{I}_{\mathtt{ID}_{[\ell-1]}}$, if $\mathtt{ID}' \notin \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}'),\mathtt{T}}$ holds for all $\mathtt{ID}' \in \mathsf{prefix}^+(\mathtt{ID})$. Then, we require $\mathsf{M}' = \mathsf{M}$ to hold after executing the following procedures.

(1) $(\mathsf{ku}_{\mathsf{kgc},\mathtt{T}}, \mathsf{sk}_{\mathsf{kgc}}) \leftarrow \mathsf{KeyUp}(\mathsf{MPK}, \mathtt{T}, \mathsf{sk}_{\mathsf{kgc}}, \mathsf{RL}_{\mathsf{kgc},\mathtt{T}}, \perp)$.

(2) For all $\mathtt{ID}' \in \mathsf{prefix}^+(\mathtt{ID})$ (in short-to-long order), execute the following (2.1) and (2.2):

(2.1) $(\mathsf{sk}_{\mathtt{ID}'}, \mathsf{sk}'_{\mathsf{pa}(\mathtt{ID}')}) \leftarrow \mathsf{GenSK}(\mathsf{MPK}, \mathsf{sk}_{\mathsf{pa}(\mathtt{ID}')}, \mathtt{ID}')$.

(2.2) $(\mathsf{ku}_{\mathtt{ID}',\mathtt{T}}, \mathsf{sk}'_{\mathtt{ID}'}) \leftarrow \mathsf{KeyUp}(\mathsf{MPK}, \mathtt{T}, \mathsf{sk}_{\mathtt{ID}'}, \mathsf{RL}_{\mathtt{ID}',\mathtt{T}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}'),\mathtt{T}})$.[10]

(3) $\mathsf{dk}_{\mathtt{ID},\mathtt{T}} \leftarrow \mathsf{GenDK}(\mathsf{MPK}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$.[11]

(4) $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{MPK}, \mathtt{ID}, \mathtt{T}, \mathsf{M})$.

(5) $\mathsf{M}' \leftarrow \mathsf{Decrypt}(\mathsf{MPK}, \mathsf{dk}_{\mathtt{ID},\mathtt{T}}, \mathsf{ct})$.

Security Definition. Let Π be an RHIBE scheme. Adaptive security with the stronger DKER is defined using a game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$. The game is parameterized by security parameter λ and polynomial $L = L(\lambda)$ representing the maximum hierarchical depth. In this game, global counter $\mathtt{T}_{\mathsf{cu}}$ is initialized as 1 to denote the "current time period. In addition, $\mathcal{C}$'s responses to $\mathcal{A}$'s queries are controlled by $\mathtt{T}_{\mathsf{cu}}$. Intuitively, $\mathcal{A}$ can receive all secret keys, key updates, and decryption keys if they are insufficient to derive $\mathsf{dk}_{\mathtt{ID}^\star,\mathtt{t}^\star}$ for target tuple $(\mathtt{ID}^\star, \mathtt{t}^\star)$. The game proceeds as follows.

First, $\mathcal{C}$ runs $(\mathsf{MPK}, \mathsf{sk}_{\mathsf{kgc}}) \leftarrow \mathsf{Setup}(1^\lambda, L)$ and prepares $\mathtt{SKList}$, which initially contains $(\mathsf{kgc}, \mathsf{sk}_{\mathsf{kgc}})$, and into which pairs of $(\mathtt{ID}, \mathsf{sk}_{\mathtt{ID}})$ generated during the game are stored. When a new $\mathsf{sk}_{\mathtt{ID}}$ is generated or existing ones are updated by executing GenSK or KeyUp, $\mathcal{C}$ stores $(\mathtt{ID}, \mathsf{sk}_{\mathtt{ID}})$ or updates them in $\mathtt{SKList}$. Note that we omit discussion of this addition/update. Then, $\mathcal{C}$ executes $(\mathsf{ku}_{\mathsf{kgc},1}, \mathsf{sk}'_{\mathsf{kgc}}) \leftarrow \mathsf{KeyUp}(\mathsf{MPK}, \mathtt{T}_{\mathsf{cu}} = 1, \mathsf{sk}_{\mathsf{kgc}}, \mathsf{RL}_{\mathsf{kgc},1} = \emptyset, \perp)$ to generate a key update for the initial time period $\mathtt{T}_{\mathsf{cu}} = 1$ and gives $(\mathsf{MPK},\ \mathsf{ku}_{\mathsf{kgc},1})$ to $\mathcal{A}$.

Then, $\mathcal{A}$ may adaptively make the following five types of a query to $\mathcal{C}$.

Secret Key Generation Query: Upon a query $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$ from $\mathcal{A}$, $\mathcal{C}$ checks if $(\mathtt{ID}, *) \notin \mathtt{SKList}$ and $(\mathsf{pa}(\mathtt{ID}), \mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}) \in \mathtt{SKList}$ for some $\mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}$, and then returns $\perp$ to $\mathcal{A}$ if this is *not* the case. Otherwise, $\mathcal{C}$ executes $(\mathsf{sk}_{\mathtt{ID}}, \mathsf{sk}'_{\mathsf{pa}(\mathtt{ID})}) \leftarrow \mathsf{GenSK}(\mathsf{MPK}, \mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}, \mathtt{ID})$. If $|\mathtt{ID}| = 1$,

[10] If $|\mathtt{ID}'| = L$, this step is skipped.

[11] Here, $\mathsf{sk}_{\mathtt{ID}}$ is the latest secret key, i.e., the result of Step (2).

or $2 \leq |\mathtt{ID}| \leq L-1$ and $\mathsf{pa}(\mathtt{ID}) \notin \mathsf{RL}_{\mathsf{pa}(\mathsf{pa}(\mathtt{ID})),\mathtt{T}_{\mathtt{cu}}}$, then $\mathcal{C}$ executes $(\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}}, \mathsf{sk}'_{\mathtt{ID}}) \leftarrow \mathsf{KeyUp}(\mathsf{MPK}, \mathtt{T}_{\mathtt{cu}}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{RL}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}} := \emptyset, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}_{\mathtt{cu}}})$ and returns $\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}}$ to $\mathcal{A}$. If $2 \leq |\mathtt{ID}| \leq L$ and $\mathsf{pa}(\mathtt{ID}) \in \mathsf{RL}_{\mathsf{pa}(\mathsf{pa}(\mathtt{ID})),\mathtt{T}_{\mathtt{cu}}}$, then $\mathcal{C}$ executes $\mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}_{\mathtt{cu}}} \leftarrow \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}_{\mathtt{cu}}} \cup \{\mathtt{ID}\}$ and returns nothing to $\mathcal{A}$.

Note that all $\mathtt{ID}$ in the following queries (except the challenge query) must be "activated", in the sense that $\mathsf{sk}_{\mathtt{ID}}$ has already been generated via this query; thus, $(\mathtt{ID}, \mathsf{sk}_{\mathtt{ID}}) \in \mathtt{SKList}$.

Secret Key Reveal Query: Until the challenge query, upon a query $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$ from $\mathcal{A}$, $\mathcal{C}$ finds $\mathsf{sk}_{\mathtt{ID}}$ from $\mathtt{SKList}$ and returns it to $\mathcal{A}$. After the challenge query, $\mathcal{C}$ checks

- If $\mathtt{T}_{\mathtt{cu}} \geq \mathtt{T}^{\star}$ and $\mathtt{ID} \in \mathsf{prefix}^{+}(\mathtt{ID}^{\star})$, then $\mathtt{ID}' \in \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}'),\mathtt{T}^{\star}}$ for some $\mathtt{ID}' \in \mathsf{prefix}^{+}(\mathtt{ID})$.

If this condition is *not* satisfied, then $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$; otherwise, $\mathcal{C}$ finds $\mathsf{sk}_{\mathtt{ID}}$ from $\mathtt{SKList}$ and returns it to $\mathcal{A}$.

Revoke & Key Update Query: Until the challenge query, upon a query $\mathsf{RL} \subseteq \mathcal{I}^{\leq L}$ (denoting the set of identities to be revoked in the next time period) from $\mathcal{A}$, $\mathcal{C}$ determines if the following conditions are satisfied simultaneously.

- $\mathsf{RL}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}} \subseteq \mathsf{RL}$ for all $\mathtt{ID} \in \mathcal{I}^{\leq L-1}$ that appear in $\mathtt{SKList}$.[12]
- For all identities $\mathtt{ID}$ such that $(\mathtt{ID}, *) \in \mathtt{SKList}$ and $\mathtt{ID}' \in \mathsf{prefix}^{+}(\mathtt{ID})$, if $\mathtt{ID}' \in \mathsf{RL}$, then $\mathtt{ID} \in \mathsf{RL}$.

After the challenge query, $\mathcal{C}$ also checks

- $\mathtt{ID}' \in \mathsf{RL}$ if $\mathtt{T}_{\mathtt{cu}} = \mathtt{T}^{\star} - 1$ and $\mathsf{sk}_{\mathtt{ID}'}$ for some $\mathtt{ID}' \in \mathsf{prefix}^{+}(\mathtt{ID}^{\star})$ has been revealed previously by the secret key reveal query.

If these conditions are *not* satisfied, then $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ increments the current time period by $\mathtt{T}_{\mathtt{cu}} \leftarrow \mathtt{T}_{\mathtt{cu}} + 1$ and executes the following operations (1) and (2) for all "activated" and non-revoked identities $\mathtt{ID}$, i.e., $\mathtt{ID} \in \mathcal{I}^{\leq L-1} \cup \{\mathsf{kgc}\}$, $(\mathtt{ID}, *) \in \mathtt{SKList}$ and $\mathtt{ID} \notin \mathsf{RL}$, in breadth-first order in the identity hierarchy.

(1) Set $\mathsf{RL}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}} \leftarrow \mathsf{RL} \cap \mathcal{I}_{\mathtt{ID}}$, where we define $\mathcal{I}_{\mathsf{kgc}} := \mathcal{I}$.

(2) Run $(\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}}, \mathsf{sk}'_{\mathtt{ID}}) \leftarrow \mathsf{KeyUp}(\mathsf{MPK}, \mathtt{T}_{\mathtt{cu}}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{RL}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}_{\mathtt{cu}}})$, where $\mathsf{ku}_{\mathsf{pa}(\mathsf{kgc}),\mathtt{T}_{\mathtt{cu}}} := \perp$.

Finally, $\mathcal{C}$ returns all of the generated $\{\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}}\}_{(\mathtt{ID},*) \in \mathtt{SKList} \setminus \mathsf{RL}}$ to $\mathcal{A}$.

Decryption Key Reveal Query: Until the challenge query, upon a query $(\mathtt{ID}, \mathtt{T}) \in \mathcal{I}^{|\mathtt{ID}|} \times \mathcal{T}$ from $\mathcal{A}$, $\mathcal{C}$ checks

- If $\mathtt{T} \leq \mathtt{T}_{\mathtt{cu}}$ holds.

After the challenge query, $\mathcal{C}$ also checks

- If $(\mathtt{ID}, \mathtt{T}) \neq (\mathtt{ID}^{\star}, \mathtt{T}^{\star})$ holds.[13]

If these conditions are *not* satisfied, then $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ finds $\mathsf{sk}_{\mathtt{ID}}$ from $\mathtt{SKList}$, runs $\mathsf{dk}_{\mathtt{ID},\mathtt{T}} \leftarrow \mathsf{GenDK}(\mathsf{MPK}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$, and returns $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ to $\mathcal{A}$.

Challenge Query: Note that $\mathcal{A}$ is permitted to make this query exactly once. Upon a query $(\mathtt{ID}^{\star}, \mathtt{T}^{\star}, \mathsf{M}_0^{\star}, \mathsf{M}_1^{\star})$ such that $|\mathsf{M}_0^{\star}| = |\mathsf{M}_1^{\star}|$ from $\mathcal{A}$, $\mathcal{C}$ determines if the following conditions are satisfied simultaneously.

- If $\mathtt{T}^{\star} \leq \mathtt{T}_{\mathtt{cu}}$, $\mathcal{A}$ has not submitted $(\mathtt{ID}^{\star}, \mathtt{T}^{\star})$ as a decryption key reveal query.
- If $\mathtt{T}^{\star} \leq \mathtt{T}_{\mathtt{cu}}$ and $\mathsf{sk}_{\mathtt{ID}}$ for $\mathtt{ID} \in \mathsf{prefix}^{+}(\mathtt{ID}^{\star})$ has been revealed to $\mathcal{A}$, then $\mathtt{ID} \in \mathsf{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}^{\star}-1}$.

[12] This check ensures that previously revoked identities remain revoked in the next time period.

[13] This is the condition of the stronger DKER. The condition of the weaker DKER is replaced by $(\mathtt{ID}, \mathtt{T}) \neq (\mathtt{ID}', \mathtt{T}^{\star})$ for all $\mathtt{ID}' \in \mathsf{prefix}^{+}(\mathtt{ID}^{\star})$.

If these conditions are *not* satisfied, then $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ selects a bit $b \in \{0,1\}$ uniformly at random, runs $\mathsf{ct}^\star \leftarrow \mathsf{Encrypt}(\mathsf{MPK}, \mathtt{ID}^\star, \mathtt{T}^\star, \mathsf{M}_b^\star)$, and returns the challenge ciphertext $\mathsf{ct}^\star$ to $\mathcal{A}$.

At some point, $\mathcal{A}$ outputs $b' \in \{0,1\}$ as its guess for b and terminates.

This completes the description of the game. In this game, $\mathcal{A}$'s adaptive security advantage is defined by $\mathsf{Adv}^{\mathtt{RHIBE}}_{\Pi,L,\mathcal{A}}(\lambda) := 2 \cdot |\Pr[b' = b] - 1/2|$.

Definition 1. *We say that an RHIBE scheme* Π *of depth* L *satisfies* adaptive security *if the advantage* $\mathsf{Adv}^{\mathtt{RHIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$ *is negligible for all PPT adversaries* $\mathcal{A}$*.*

4 Pairing-based HIBE

In this section, we describe HIBE and its additional properties that are used in our construction. The additional properties can be achieved in most existing pairing-based HIBE constructions; therefore, HIBE with those properties can be considered an abstraction of existing pairing-based HIBE constructions. We briefly review the definition of *plain* HIBE in Section 4.1, and in Section 4.2, we discuss additional properties of HIBE. Finally, in Section 4.3, we describe how state-of-the-art HIBE schemes satisfy these properties.

4.1 Plain HIBE

A plain HIBE scheme $\mathsf{H}.\Pi$ with depth L consists of four algorithms ($\mathsf{H.Setup}, \mathsf{H.Encrypt}, \mathsf{H.GenSK}, \mathsf{H.Decrypt}$) defined as follows.

- $\mathsf{H.Setup}(1^\lambda, L) \to (\mathsf{H.MPK}, \mathsf{H.MSK})$: The *setup* algorithm takes security parameter λ and maximum hierarchical depth L as input, and it outputs a master public key $\mathsf{H.MPK}$ and a smaster secret key $\mathsf{H.MSK}$.
- $\mathsf{H.Encrypt}(\mathsf{H.MPK}, \mathtt{ID}, \mathsf{M}) \to \mathsf{H.ct}_{\mathtt{ID}}$: The *encryption* algorithm takes $\mathsf{H.MPK}$, an identity $\mathtt{ID} \in \mathsf{H}.\mathcal{I}^{|\mathtt{ID}|}$, and a plaintext M as input, and outputs a ciphertext $\mathsf{H.ct}_{\mathtt{ID}}$.
- $\mathsf{H.GenSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{\mathtt{ID}'}, \mathtt{ID} := (\mathtt{ID}', \mathtt{id})) \to \mathsf{H.sk}_{\mathtt{ID}}$: The *secret key generation* algorithm takes $\mathsf{H.MPK}$, a secret key $\mathsf{H.sk}_{\mathtt{ID}'}$, and an identity $\mathtt{ID} \in \mathsf{H}.\mathcal{I}_{\mathtt{ID}'}$ as input, and outputs a secret key $\mathsf{H.sk}_{\mathtt{ID}}$. Here, the second input $\mathsf{H.sk}_{\mathtt{ID}'}$ can be replaced by $\mathsf{H.MSK}$.
- $\mathsf{H.Decrypt}(\mathsf{H.MPK}, \mathsf{H.sk}_{\mathtt{ID}}, \mathsf{H.ct}_{\mathtt{ID}}) \to \mathsf{M}$: The *decryption* algorithm takes $\mathsf{H.MPK}$, $\mathsf{H.sk}_{\mathtt{ID}}$, and $\mathsf{H.ct}_{\mathtt{ID}}$ as input, and outputs the decryption result M.

The correctness of HIBE ensures that the $\mathsf{H.Decrypt}$ algorithm outputs a correct decryption result. In addition, most pairing-based HIBE schemes further satisfy the condition that $\mathsf{H.GenSK}(\mathsf{H.MPK}, \mathsf{H.MSK}, \mathtt{ID}) \approx \mathsf{H.GenSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{\mathtt{ID}'}, \mathtt{ID})$ even when given $\mathsf{H.sk}_{\mathtt{ID}'}$. The security of HIBE ensures that it is difficult for an adversary who obtains polynomially many secret keys $\mathsf{H.sk}_{\mathtt{ID}}$ such that $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$ to extract secret information from $\mathsf{H.ct}_{\mathtt{ID}^\star}$.

By definition, an HIBE system has three natural restrictions. First, only the KGC that knows $\mathsf{H.MSK}$ can create a secret key $\mathsf{H.sk}_{\mathtt{ID}}$ for any $\mathtt{ID} \in \mathsf{H}.\mathcal{I}^{|\mathtt{ID}|}$. Next, a user with $\mathsf{H.sk}_{\mathtt{ID}}$ can create $\mathsf{H.sk}_{\mathtt{ID}'}$ iff $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}')$. Thus, finally, an adversary cannot obtain $\mathsf{H.sk}_{\mathtt{ID}}$ for $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star)$ in the security game, where $\mathtt{ID}^\star$ is a target identity.

4.2 Properties Extracted from Existing HIBE Constructions

We presented the proposed approach to construct RHIBE schemes in Section 2.4. That explanation was specific to BBG HIBE because we rely on several special algebraic/security properties of the scheme, which the definition of plain HIBE does not capture. To construct adaptively secure RHIBE schemes, here, we introduce three additional properties of HIBE, i.e., *MSK evaluatability*, *first-level wildcarded SK*, and *prefix decryption restriction*, to abstract the properties to realize the proposed approach. Note that the extracted properties can be achieved by most pairing-based constructions, as discussed in Section 4.3.

MSK Evaluatability. In Section 2.4, users or the reduction algorithm performed the following operations.

- All parent users $\mathsf{pa}(\mathtt{ID})$ compute $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ to create $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$, where $\alpha_{\mathsf{pa}(\mathtt{ID}),\theta} \leftarrow_R \mathbb{Z}_p$. Here, although the reduction algorithm may change the computation to $g_2^{\alpha-\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$, it does not change the distribution.
- Each user $\mathtt{ID}$ employs $\mathsf{sk}_{\mathtt{ID},\theta}$ and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$, which are the BBG secret keys whose first factors of sk_1 and ku_1 are $g_2^{\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$ and $g_2^{\alpha-\alpha_{\mathsf{pa}(\mathtt{ID}),\theta}}$, respectively, and derives $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$, which is a BBG secret key whose first factor of dk_1 is g_2^{α}. In addition, the distribution of $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ is independent of $\mathsf{sk}_{\mathtt{ID},\theta}$ and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$.

Note that these operations are not supported by plain HIBE; therefore, we introduce *MSK evaluatability*. In the following, we use notation $\mathsf{H.sk}_{\mathtt{ID}}[\mathsf{H.MSK}]$ rather than $\mathsf{H.sk}_{\mathtt{ID}}$, to explicitly describe the MSK-component of $\mathsf{H.sk}_{\mathtt{ID}}$. Here, let $\mathsf{H}.\mathcal{MSK}$ denote an MSK space of HIBE, which is a finite multiplicative abelian group. Intuitively, MSK evaluatability allows anyone to perform the following operations.

1. $\mathsf{H.MPK}$ can be used to sample $\widehat{\mathsf{H.MSK}}$, which we refer to as a *pseudo-MSK*, randomly from $\mathsf{H}.\mathcal{MSK}$. Note that $\widehat{\mathsf{H.MSK}} \neq \mathsf{H.MSK}$ holds with overwhelming probability.

2. $\mathsf{H.sk}_{\mathtt{ID}}[\widehat{\mathsf{H.MSK}}_1]$ and $\mathsf{H.sk}_{\mathtt{ID}}[\widehat{\mathsf{H.MSK}}_2]$ for the same $\mathtt{ID} \in \mathcal{I}$ under arbitrary $(\widehat{\mathsf{H.MSK}}_1, \widehat{\mathsf{H.MSK}}_2) \in \mathsf{H}.\mathcal{MSK}^2$ can be merged into $\mathsf{H.sk}_{\mathtt{ID}}[f(\widehat{\mathsf{H.MSK}}_1, \widehat{\mathsf{H.MSK}}_2)]$ by evaluating their MSK-parts with function $f : \mathsf{H}.\mathcal{MSK} \times \mathsf{H}.\mathcal{MSK} \to \mathsf{H}.\mathcal{MSK}$. Here, $\widehat{\mathsf{H.MSK}}_1$ and $\widehat{\mathsf{H.MSK}}_2$ are arbitrary elements in $\mathsf{H}.\mathcal{MSK}$ including $\mathsf{H.MSK}$.

We formally define MSK evaluatability in Definition 2.

Definition 2 (MSK Evaluatability)**.** *Let* $\mathsf{H}.\Pi$ *be an HIBE scheme. We say that* $\mathsf{H}.\Pi$ *supports MSK evaluatability w.r.t. a function class* $\mathcal{F}$ *if there exist two probabilistic algorithms* $\mathsf{H.SampMSK}$ *and* $\mathsf{H.EvalMSK}$*.*

- $\mathsf{H.SampMSK}(\mathsf{H.MPK}) \to \widehat{\mathsf{H.MSK}}$*: The* pseudo-MSK sampling *algorithm takes* $\mathsf{H.MPK}$ *as input and outputs* $\widehat{\mathsf{H.MSK}} \in \mathsf{H}.\mathcal{MSK}$*.*
- $\mathsf{H.EvalMSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{\mathtt{ID}}[\widehat{\mathsf{H.MSK}}_1], \mathsf{H.sk}_{\mathtt{ID}}[\widehat{\mathsf{H.MSK}}_2], f) \to \mathsf{H.sk}_{\mathtt{ID}}[f(\widehat{\mathsf{H.MSK}}_1, \widehat{\mathsf{H.MSK}}_2)]$*: The* MSK evaluation *algorithm takes* $\mathsf{H.sk}_{\mathtt{ID}}[\widehat{\mathsf{H.MSK}}_1]$ *and* $\mathsf{H.sk}_{\mathtt{ID}}[\widehat{\mathsf{H.MSK}}_2]$ *for the same* $\mathtt{ID}$*, under* $(\widehat{\mathsf{H.MSK}}_1, \widehat{\mathsf{H.MSK}}_2) \in \mathsf{H}.\mathcal{MSK}^2$*, and a function* $f \in \mathcal{F}$ *as input, and then outputs* $\mathsf{H.sk}_{\mathtt{ID}}[f(\widehat{\mathsf{H.MSK}}_1, \widehat{\mathsf{H.MSK}}_2)]$*.*

In addition, these algorithms satisfy the following requirements.

▷ **Pseudo-MSK Indistinguishability***: For any* $f \in \mathcal{F}$ *and any* $\widehat{\mathsf{H.MSK}} \in \mathsf{H}.\mathcal{MSK}$*, given* $\widehat{\mathsf{H.MSK}}$*, it holds that*

$$\mathsf{H.SampMSK}(\mathsf{H.MPK}) \approx f(\widehat{\mathsf{H.MSK}}, \mathsf{H.SampMSK}(\mathsf{H.MPK})).$$

▷ **Evaluation Invariance**: *For any* $f \in \mathcal{F}$, *any* $(\widehat{\mathsf{H.MSK}}_1, \widehat{\mathsf{H.MSK}}_2) \in \mathsf{H}.\mathcal{MSK}^2$, *and any* $\mathtt{ID} \in \mathcal{I}$, *given* $\mathsf{H.sk}_{\mathtt{ID}}[\widehat{\mathsf{H.MSK}}_1]$ *and* $\mathsf{H.sk}_{\mathtt{ID}}[\widehat{\mathsf{H.MSK}}_2]$, *it holds that*

$$\mathsf{H.GenSK}(\mathsf{H.MPK}, f(\widehat{\mathsf{H.MSK}}_1, \widehat{\mathsf{H.MSK}}_2), \mathtt{ID})$$
$$\approx \mathsf{H.EvalMSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{\mathtt{ID}}[\widehat{\mathsf{H.MSK}}_1], \mathsf{H.sk}_{\mathtt{ID}}[\widehat{\mathsf{H.MSK}}_2], f).$$

Remark 1. *Although we have defined MSK evaluatability for general* $\mathcal{F}$, *we consider a simple class* $\mathcal{F} := \{\mathtt{mul}, \mathtt{div}\}$ *throughout the paper, where* $\mathtt{mul}$ *and* $\mathtt{div}$ *indicate multiplication and division, respectively. In other words,* $\mathtt{mul}(\widehat{\mathsf{H.MSK}}_1, \widehat{\mathsf{H.MSK}}_2) = \widehat{\mathsf{H.MSK}}_1 \cdot \widehat{\mathsf{H.MSK}}_2$ *and* $\mathtt{div}(\widehat{\mathsf{H.MSK}}_1, \widehat{\mathsf{H.MSK}}_2) = \widehat{\mathsf{H.MSK}}_1 / \widehat{\mathsf{H.MSK}}_2$. *Therefore, we omit "w.r.t.* $\mathcal{F}$*" and simply state MSK evaluatability for simplicity.*

Remark 2. *Although we have defined pseudo-MSK indistinguishability for a general distribution* $\mathsf{H.SampMSK}(\mathsf{H.MPK})$, *we consider only the case where* $\mathsf{H.SampMSK}(\mathsf{H.MPK})$ *is a uniform distribution on* $\mathsf{H}.\mathcal{MSK}$ *throughout the paper. When* $\mathcal{F} = \{\mathtt{mul}, \mathtt{div}\}$, *the uniformity of* $\mathsf{H.SampMSK}(\mathsf{H.MPK})$ *implies pseudo-MSK indistinguishability. In addition,* $\mathsf{H.SampMSK}(\mathsf{H.MPK}) \neq \mathsf{H.MSK}$ *holds with high probability.*

First-level wildcarded SK. In Section 2.4, we assumed the following special encoding/security requirement of BBG HIBE:

- $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$ is a BBG secret key, where $\mathsf{pa}(\mathtt{ID})$ is encoded in levels from 2 to $|\mathsf{pa}(\mathtt{ID})| + 1$. $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$ is used to create $\mathsf{sk}_{\mathtt{ID},\theta}$ which is a BBG secret key, where $\mathtt{ID}$ is encoded in levels from 2 to $|\mathtt{ID}| + 1$. In addition, $\mathsf{sk}_{\mathtt{ID},\theta}$ is used to create $\mathsf{ku}_{\mathtt{ID},\mathtt{T},\theta'}$, which is a BBG secret key, where $\mathtt{T}$ and $\mathtt{ID}$ are encoded in levels 1 and from 2 to $|\mathtt{ID}| + 1$, respectively.
- Note that the BBG HIBE does not degrade security even if $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$ for $\mathsf{pa}(\mathtt{ID}) \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$ and $\mathsf{sk}_{\mathtt{ID},\theta}$ for $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$ are revealed.

These encodings/security requirements are not supported by plain HIBE; thus, we introduce the *first-level wildcarded SK*. This concept is similar to that of wildcarded IBE [ABC+11] and wicked IBE [AKN07]. Roughly speaking, if an HIBE scheme supports first-level wildcarded SK, a wildcard "$*$" can be used for the first level of identity vectors when generating secret keys, whereas wicked IBE allows wildcards at each level of identity vectors for secret keys.[14] Thus, first-level wildcarded SK enables all parent users to allow more flexible delegation procedures than plain HIBE. This "wildcard" can be replaced with any element identity $\mathtt{id} \in \mathsf{H}.\mathcal{I}$; therefore, $\mathsf{H.sk}_{(*,\mathtt{id}_1,\ldots,\mathtt{id}_{|\mathtt{ID}|})}$ can be used to derive $\mathsf{H.sk}_{(\mathtt{T},\mathtt{id}_1,\ldots,\mathtt{id}_{|\mathtt{ID}|})}$ for any $\mathtt{T} \in \mathsf{H}.\mathcal{I}$ and $\mathsf{H.sk}_{(*,\mathtt{id}_1,\ldots,\mathtt{id}_{|\mathtt{ID}|+1})}$ for any $\mathtt{id}_{|\mathtt{ID}|+1} \in \mathsf{H}.\mathcal{I}$. In other words, an identity space $\mathsf{H}.\mathcal{I}^{\leq L}$ for secret keys becomes $\mathsf{W}.\mathcal{I}^{\leq L} := (\mathsf{H}.\mathcal{I} \cup \{*\}) \times \mathsf{H}.\mathcal{I}^{\leq L-1}$. Note that identity space $\mathsf{H}.\mathcal{I}^{\leq L}$ for ciphertexts is the same as that of the original HIBE, i.e., wildcards are *not* permitted for ciphertexts at all levels. Here, for simplicity, we define the following notations regarding to this notion.

(I) For any $\mathtt{ID} := (\mathtt{id}_1, \mathtt{id}_2, \ldots, \mathtt{id}_{|\mathtt{ID}|}) \in \mathsf{W}.\mathcal{I}^{|\mathtt{ID}|}$, let

$$\mathsf{W}.\mathcal{I}_{\mathtt{ID}} := \begin{cases} \mathsf{H}.\mathcal{I}_{\mathtt{ID}} \cup \{\mathtt{ID}' \in \mathsf{H}.\mathcal{I}^{|\mathtt{ID}|} \mid \forall \mathtt{id}'_1 \in \mathsf{H}.\mathcal{I}, \mathtt{ID}' = (\mathtt{id}'_1, \mathtt{id}_2 \ldots, \mathtt{id}_{|\mathtt{ID}|})\} & \text{if } \mathtt{id}_1 = *, \\ \mathsf{H}.\mathcal{I}_{\mathtt{ID}} & \text{otherwise.} \end{cases}$$

(II) Let $\mathsf{W.prefix}^+((\mathtt{id}_1, \ldots, \mathtt{id}_\ell)) := \mathsf{prefix}^+((\mathtt{id}_1, \ldots, \mathtt{id}_\ell)) \cup \{(*, \mathtt{id}_2, \ldots, \mathtt{id}_\ell)\}$ be a wildcarded prefix for an identity $(\mathtt{id}_1, , \ldots, \mathtt{id}_\ell) \in \mathsf{H}.\mathcal{I}^\ell$.

[14] In this sense, our notion is more closely related to wicked IBE rather than wildcarded IBE because wildcards are associated with a ciphertext/secret key in wildcarded/wicked IBE. Nonetheless, we use "wildcarded" to express this intuitive property.

Here, we examined a toy example. Assume $\mathsf{H}.\mathcal{I} = \{0,1\}$. For $\mathtt{ID} = (*,1)$, we have $\mathsf{W}.\mathcal{I}_{\mathtt{ID}} = \mathsf{H}.\mathcal{I}_{\mathtt{ID}} \cup \{(0,1),(1,1)\}$, where $\mathsf{H}.\mathcal{I}_{\mathtt{ID}} = \{(*,1,0),(*,1,1)\}$, and, for $\mathtt{ID} = (0,1)$, we have $\mathsf{W}.\mathsf{prefix}^{+}(\mathtt{ID}) = \mathsf{prefix}^{+}(\mathtt{ID}) \cup \{(*,1)\} = \{0,(0,1),(*,1)\}$.

Formally, first-level wildcarded SK is defined as follows.

Definition 3 (First-level Wildcarded SK)**.** *Let* $\mathsf{H}.\Pi$ *be an HIBE scheme. We say that* $\mathsf{H}.\Pi$ *supports first-level wildcarded SK if* $\mathsf{H}.\mathsf{GenSK}$ *can be modified as follows.*

- $\mathsf{H}.\mathsf{GenSK}(\mathsf{H}.\mathsf{MPK}, \mathsf{H}.\mathsf{sk}_{\mathtt{ID}'}, \mathtt{ID} \in \mathsf{W}.\mathcal{I}_{\mathtt{ID}}) \to \mathsf{H}.\mathsf{sk}_{\mathtt{ID}}$*: This is the same as* $\mathsf{H}.\mathsf{GenSK}$ *in plain HIBE, with the exception that it takes* $\mathtt{ID} \in \mathsf{W}.\mathcal{I}^{\leq L}$ *as input rather than* $\mathtt{ID} \in \mathsf{H}.\mathcal{I}^{\leq L}$*.*

The following correctness and wildcarded secret key query are satisfied.

▷ **Correctness***: For any* $\mathtt{ID}' \in \mathsf{W}.\mathcal{I}^{\leq L-1}$ *and* $\mathtt{ID} \in \mathsf{W}.\mathcal{I}_{\mathtt{ID}'}$*, given* $\mathsf{H}.\mathsf{sk}_{\mathtt{ID}'}$*, it holds that*

$$\mathsf{H}.\mathsf{GenSK}(\mathsf{H}.\mathsf{MPK}, \mathsf{H}.\mathsf{MSK}, \mathtt{ID}') \approx \mathsf{H}.\mathsf{GenSK}(\mathsf{H}.\mathsf{MPK}, \mathsf{H}.\mathsf{sk}_{\mathtt{ID}}, \mathtt{ID}').$$

▷ **Wildcarded Secret Key Query***: In the security game, an adversary is allowed to receive secret keys* $\mathsf{H}.\mathsf{sk}_{\mathtt{ID}}$ *for any* $\mathtt{ID} \in \mathsf{W}.\mathcal{I}^{\leq L} \setminus \mathsf{W}.\mathsf{prefix}^{+}(\mathtt{ID}^\star)$*.*

Prefix Decryption Restriction. In Section 2.4, we utilized the following special form of a BBG secret key.

- $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ is a special form of an BBG secret key that does not contain $(h_i^r)_{i\in[|\mathtt{ID}|+2,L+1]}$.
- Note that BBG HIBE does not degrade security even if $(\mathsf{dk}_{\mathtt{ID}^\star_{[\ell]},\mathtt{T}^\star})_{\ell\in[|\mathtt{ID}^\star|-1]}$ are revealed.

The special form of the secret keys is not supported by plain HIBE; thus, we introduce the *prefix decryption restriction*. This property degrades the decryption capability of an HIBE secret key. Note that a similar concept called limited delegation was introduced by Shacham [Sha07].[15] If an HIBE scheme supports the prefix decryption restriction, there is an HIBE decryption key $\mathsf{H}.\mathsf{dk}_{\mathtt{ID}}$, which can be considered an HIBE secret key for $\mathtt{ID}$ without delegation functionality. In contrast to an HIBE secret key $\mathsf{H}.\mathsf{sk}_{\mathtt{ID}}$, a decryption key $\mathsf{H}.\mathsf{dk}_{\mathtt{ID}}$ for $\mathtt{ID} \in \mathsf{H}.\mathcal{I}^{\leq L}$ can decrypt ciphertext $\mathsf{H}.\mathsf{ct}_{\mathtt{ID}'}$ iff $\mathtt{ID} = \mathtt{ID}'$. In other words, $\mathsf{H}.\mathsf{dk}_{\mathtt{ID}}$ cannot decrypt ciphertext $\mathsf{H}.\mathsf{ct}_{\mathtt{ID}'}$ for $\mathtt{ID}' \in \mathsf{H}.\mathcal{I}_{\mathtt{ID}}$, whereas $\mathsf{H}.\mathsf{sk}_{\mathtt{ID}}$ can perform this decryption.

Formally, the prefix decryption restriction is defined as follows.

Definition 4 (Prefix Decryption Restriction)**.** *An HIBE scheme supports the prefix decryption restriction if there exists* $\mathsf{H}.\mathsf{GenDK}$ *as follows.*

- $\mathsf{H}.\mathsf{GenDK}(\mathsf{H}.\mathsf{MPK}, \mathsf{H}.\mathsf{sk}_{\mathtt{ID}}) \to \mathsf{H}.\mathsf{dk}_{\mathtt{ID}}$*: The* decryption key generation *algorithm takes* $\mathsf{H}.\mathsf{MPK}$ *and* $\mathsf{H}.\mathsf{sk}_{\mathtt{ID}}$ *as input and outputs a decryption key* $\mathsf{H}.\mathsf{dk}_{\mathtt{ID}}$*.*

The following correctness and decryption key query are satisfied.

▷ **Correctness:** *For any* $\mathtt{ID} \in \mathsf{H}.\mathcal{I}^{\leq L}$*, a correctly-generated ciphertext* $\mathsf{H}.\mathsf{ct}_{\mathtt{ID}}$ *can be decrypted correctly with* $\mathsf{H}.\mathsf{dk}_{\mathtt{ID}}$*. In addition, for any* $\mathtt{ID} \in \mathsf{H}.\mathcal{I}^{\leq L}$ *given* $\mathsf{H}.\mathsf{sk}_{\mathtt{ID}}$*, it holds that*

$$\mathsf{H}.\mathsf{GenDK}(\mathsf{H}.\mathsf{MPK}, \mathsf{H}.\mathsf{GenSK}(\mathsf{H}.\mathsf{MPK}, \mathsf{H}.\mathsf{MSK}, \mathtt{ID})) \approx \mathsf{H}.\mathsf{GenDK}(\mathsf{H}.\mathsf{MPK}, \mathsf{H}.\mathsf{sk}_{\mathtt{ID}}).$$

▷ **Decryption Key Query:** *In the security game, an adversary is permitted to receive decryption keys* $\mathsf{H}.\mathsf{dk}_{\mathtt{ID}}$ *for any* $\mathtt{ID} \in \mathsf{H}.\mathcal{I}^{\leq L} \setminus \{\mathtt{ID}^\star\}$*.*

[15] Shacham define an HIBE secret key $\mathsf{H}.\mathsf{sk}_{\mathtt{ID}}$ with limited delegation such that it can derive a secret key $\mathsf{H}.\mathsf{sk}_{\mathtt{ID}'}$ of a suffix identity $\mathtt{ID}'$ only when $|\mathtt{ID}'|$ is less than or equal to limited bound $|\mathtt{ID}'| = L' < L$.

4.3 Concrete Examples

Here, we demonstrate that the Chen-Wee (CW) HIBE [CW14][16] satisfies the above properties because we consider it to be conceptually simpler to understand than other state-of-the-art HIBE schemes [CG17, GCTC16]. The CW HIBE is designed on an asymmetric bilinear group of prime-order p equipped with a non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Here, we use the implicit notation $[a]_1 \coloneqq g_1^a \in \mathbb{G}_1$, $[a]_2 \coloneqq g_2^a \in \mathbb{G}_2$, and $[a]_T \coloneqq e(g_1, g_2)^a \in \mathbb{G}_T$ [EHK+17]. In addition, for a vector $\mathbf{a} \coloneqq (a_1, \ldots, a_d) \in \mathbb{Z}_p^k$, we use the notation $[\mathbf{a}]_1 \coloneqq ([a_1]_1, \ldots, [a_d]_1) \in \mathbb{G}_1^k$. The analogous notation is used for $[\mathbf{a}]_2, [\mathbf{a}]_T$ and a matrix $[\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{A}]_T$. If $\mathbf{A}$ and $\mathbf{B}$ are matrices with compatible dimensions, let $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{A}^\top \mathbf{B}]_T$. First, the CW HIBE samples two matrices $\mathbf{A}$ and $\mathbf{B}$ in $\mathbb{Z}_p^{(k+1)\times k}$ from a matrix distribution $\mathcal{D}_k$ [EHK+17], $\mathbf{W}_1, \ldots, \mathbf{W}_{L+1} \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+1)}$, and $\mathbf{k} \leftarrow_R \mathbb{Z}_p^{k+1}$. Then, $\mathsf{H.Setup}(1^\lambda, L)$ outputs $\mathsf{H.MPK}$ and $\mathsf{H.MSK}$:

$$\mathsf{H.MPK} \coloneqq \left(\begin{matrix} [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, \ldots, [\mathbf{W}_{L+1}^\top \mathbf{A}]_1 \\ [\mathbf{B}]_2, [\mathbf{W}_1 \mathbf{B}]_2, \ldots, [\mathbf{W}_{L+1}\mathbf{B}]_2 \end{matrix} ; [\mathbf{A}^\top \mathbf{k}]_T \right), \quad \mathsf{H.MSK} \coloneqq [\mathbf{k}]_2.$$

Thus, the master secret key space $\mathsf{H}.\mathcal{MSK}$ of the scheme is $\mathbb{G}_2^{k+1}$. This scheme has $\mathsf{H.ct}_{\mathtt{ID}^\star}$ and $\mathsf{H.sk}_{\mathtt{ID}}$:

$$\mathsf{H.ct}_{\mathtt{ID}^\star} \coloneqq \left(c_0 \coloneqq [\mathbf{As}]_1, c_1 \coloneqq [(\mathtt{id}_1^\star \mathbf{W}_1^\top + \cdots + \mathtt{id}_{|\mathtt{ID}^\star|}^\star \mathbf{W}_{|\mathtt{ID}^\star|}^\top + \mathbf{W}_{L+1}^\top)\mathbf{As}]_1, c_2 \coloneqq \mathsf{M} \cdot [\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}]_T \right),$$

$$\mathsf{H.sk}_{\mathtt{ID}} \coloneqq \left(\begin{matrix} \mathsf{sk}_{\mathtt{ID},0} \coloneqq [\mathbf{Br}]_2, \\ \mathsf{sk}_{\mathtt{ID},1} \coloneqq [\mathbf{k}]_2 \cdot [(\mathtt{id}_1 \mathbf{W}_1 + \cdots + \mathtt{id}_{|\mathtt{ID}|} \mathbf{W}_{|\mathtt{ID}|} + \mathbf{W}_{L+1})\mathbf{Br}]_2 \\ \mathsf{sk}_{\mathtt{ID},|\mathtt{ID}|+1} \coloneqq [\mathbf{W}_{|\mathtt{ID}|+1}\mathbf{Br}]_2, \ldots, \mathsf{sk}_{\mathtt{ID},L} \coloneqq [\mathbf{W}_L \mathbf{Br}]_2 \end{matrix} \right),$$

where $\mathbf{s}, \mathbf{r} \leftarrow_R \mathbb{Z}_p^k$. It is easy to verify the correctness of the scheme. Specifically, the decryption works as $\mathsf{M} = c_2 \cdot e(c_1, \mathsf{sk}_{\mathtt{ID}^\star,0}) / e(c_0, \mathsf{sk}_{\mathtt{ID}^\star,1})$, where $\mathsf{sk}_{\mathtt{ID}^\star,|\mathtt{ID}^\star|+1}, \ldots, \mathsf{sk}_{\mathtt{ID}^\star,L}$ are not used. They are responsible only for key delegation.

Prior to examining whether CW HIBE satisfies the properties discussed in Section 4.2, we first describe the crucial part in its security proof. Chen and Wee proved adaptive security using Waters' dual system encryption methodology [Wat09]. To overcome the main step (i.e., changing secret keys $\mathsf{H.sk}_{\mathtt{ID}}$ queried by the adversary to be semi-functional), they observed that the following distributions are uniformly random in $\mathbb{Z}_p^{L-|\mathtt{ID}|+2}$:

$$\{\zeta_1 \mathtt{id}_1^\star + \cdots + \zeta_{|\mathtt{ID}^\star|} \mathtt{id}_{|\mathtt{ID}^\star|}^\star + \zeta_{L+1},\ \zeta_1 \mathtt{id}_1 + \cdots + \zeta_{|\mathtt{ID}|} \mathtt{id}_{|\mathtt{ID}|} + \zeta_{L+1}, (\zeta_i)_{i \in [|\mathtt{ID}|+1, L]}\},$$

where $\zeta_i \leftarrow_R \mathbb{Z}_p$ for all $i \in [L+1]$. Intuitively, $\zeta_1 \mathtt{id}_1^\star + \cdots + \zeta_{|\mathtt{ID}^\star|} \mathtt{id}_{|\mathtt{ID}^\star|}^\star + \zeta_{L+1}$ is a random semi-functional component of ciphertext element c_1, and the other $\zeta_1 \mathtt{id}_1 + \cdots + \zeta_{|\mathtt{ID}|} \mathtt{id}_{|\mathtt{ID}|} + \zeta_{L+1}$ and $(\zeta_i)_{i \in [|\mathtt{ID}|+1, L]}$ are random semi-functional components of secret key elements $\mathsf{sk}_{\mathtt{ID},1}$ and $(\mathsf{sk}_{\mathtt{ID},i})_{i \in [|\mathtt{ID}|+1,L]}$, respectively. Here, the uniformity claim holds when $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$. In particular, when $|\mathtt{ID}^\star| < |\mathtt{ID}|$, the claim holds because the second element $\zeta_1 \mathtt{id}_1 + \cdots + \zeta_{|\mathtt{ID}|} \mathtt{id}_{|\mathtt{ID}|} + \zeta_{L+1}$ distributes uniformly random in $\mathbb{Z}_p$ even when $\mathtt{id}_i = \mathtt{id}_i^\star$ for $i \in [|\mathtt{ID}^\star|]$ because the random $(\zeta_{|\mathtt{ID}^\star|+1}, \ldots, \zeta_{|\mathtt{ID}|})$ are independent of the other elements. The claim also holds for $|\mathtt{ID}^\star| \geq |\mathtt{ID}|$ because $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$, which implies that $\mathtt{id}_{i^\star}^\star \neq \mathtt{id}_{i^\star}$ holds for some $i^\star \in [|\mathtt{ID}|]$. Therefore, $\{\zeta_0 + \zeta_{i^\star} \mathtt{id}_{i^\star}^\star, \zeta_0 + \zeta_{i^\star} \mathtt{id}_{i^\star}\}$ distributes uniformly in $\mathbb{Z}_p^2$.

[16] To be precise, we use Chen-Gay-Wee's instantiation of a dual system group [CGW15] as Chen-Wee's HIBE scheme. In addition, the scheme proposed by Gong et al. satisfies prefix decryption restriction only if $n \geq 2$, where n is a predetermined parameter to manage the efficiency trade-off in their scheme, and we note that this flexible parameter n is crucial. It is difficult to prove that other unbounded HIBE schemes [Lew12, LW11] have the prefix decryption restriction because they do not have such a parameter.

MSK Evaluatability. Since the master secret key space of the CW HIBE is $\mathbb{G}_2^{k+1}$, we define the $\mathsf{H.SampMSK}$ and $\mathsf{H.SampMSK}$ algorithms as follows.

- $\mathsf{H.SampMSK}(\mathsf{H.MPK}) \to \widehat{\mathsf{H.MSK}}$: It samples $\widehat{\mathbf{k}} \leftarrow_R \mathbb{Z}_p^{k+1}$ and outputs $\widehat{\mathsf{H.MSK}} := [\widehat{\mathbf{k}}]_2$.
- $\mathsf{H.EvalMSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{\mathtt{ID}}[[\widehat{\mathbf{k}}_1]_2], \mathsf{H.sk}_{\mathtt{ID}}[[\widehat{\mathbf{k}}_2]_2], f) \to \mathsf{H.sk}_{\mathtt{ID}}[f([\widehat{\mathbf{k}}_1]_2, [\widehat{\mathbf{k}}_2]_2)]$: On input
  - $\mathsf{H.sk}_{\mathtt{ID}}[[\widehat{\mathbf{k}}_1]_2] = (\mathsf{H.sk}^{(1)}_{\mathtt{ID},0}, \mathsf{H.sk}^{(1)}_{\mathtt{ID},1}, (\mathsf{H.sk}^{(1)}_{\mathtt{ID},i})_{i\in[|\mathtt{ID}|+1,L]})$,
  - $\mathsf{H.sk}_{\mathtt{ID}}[[\widehat{\mathbf{k}}_2]_2] = (\mathsf{H.sk}^{(2)}_{\mathtt{ID},0}, \mathsf{H.sk}^{(2)}_{\mathtt{ID},1}, (\mathsf{H.sk}^{(2)}_{\mathtt{ID},i})_{i\in[|\mathtt{ID}|+1,L]})$,

  it samples $\mathbf{r}' \leftarrow_R \mathbb{Z}_p^k$ and outputs $\mathsf{H.sk}_{\mathtt{ID}}[f([\widehat{\mathbf{k}}_1]_2, [\widehat{\mathbf{k}}_2]_2)] = (\mathsf{H.sk}_{\mathtt{ID},0}, \mathsf{H.sk}_{\mathtt{ID},1}, (\mathsf{H.sk}_{\mathtt{ID},i})_{i\in[|\mathtt{ID}|+1,L]})$, where
  - if $f = \mathtt{mul}$: $\mathsf{sk}_{\mathtt{ID},0} = \mathsf{sk}^{(1)}_{\mathtt{ID},0} \cdot \mathsf{sk}^{(2)}_{\mathtt{ID},0} \cdot [\mathbf{B}\mathbf{r}']_2$, $\mathsf{sk}_{\mathtt{ID},1} = \mathsf{sk}^{(1)}_{\mathtt{ID},1} \cdot \mathsf{sk}^{(2)}_{\mathtt{ID},1} \cdot [(\mathtt{id}_1\mathbf{W}_1 + \cdots + \mathtt{id}_{|\mathtt{ID}|}\mathbf{W}_{|\mathtt{ID}|} + \mathbf{W}_{L+1})\mathbf{B}\mathbf{r}']_2$, and $\mathsf{sk}_{\mathtt{ID},i} = \mathsf{sk}^{(1)}_{\mathtt{ID},i} \cdot \mathsf{sk}^{(2)}_{\mathtt{ID},i} \cdot [\mathbf{W}_i\mathbf{B}\mathbf{r}']_2$;
  - if $f = \mathtt{div}$: $\mathsf{sk}_{\mathtt{ID},0} = (\mathsf{sk}^{(1)}_{\mathtt{ID},0}/\mathsf{sk}^{(2)}_{\mathtt{ID},0}) \cdot [\mathbf{B}\mathbf{r}']_2$, $\mathsf{sk}_{\mathtt{ID},1} = (\mathsf{sk}^{(1)}_{\mathtt{ID},1}/\mathsf{sk}^{(2)}_{\mathtt{ID},1}) \cdot [(\mathtt{id}_1\mathbf{W}_1 + \cdots + \mathtt{id}_{|\mathtt{ID}|}\mathbf{W}_{|\mathtt{ID}|} + \mathbf{W}_{L+1})\mathbf{B}\mathbf{r}']_2$, and $\mathsf{sk}_{\mathtt{ID},i} = (\mathsf{sk}^{(1)}_{\mathtt{ID},i}/\mathsf{sk}^{(2)}_{\mathtt{ID},i}) \cdot [\mathbf{W}_i\mathbf{B}\mathbf{r}']_2$.

These algorithms satisfy the requirements in Definition 2 as follows:

▷ **Pseudo-MSK Indistinguishability**: For any $[\widehat{\mathbf{k}}]_2 \in \mathbb{G}_2^{k+1}$, given $[\widehat{\mathbf{k}}]_2$, $\mathtt{mul}([\widehat{\mathbf{k}}]_2, [\widehat{\mathbf{k}}]_2) = [\widehat{\mathbf{k}}]_2 \cdot [\widehat{\mathbf{k}}]_2 = [\widehat{\mathbf{k}} + \widehat{\mathbf{k}}]_2$ and $\mathtt{div}([\widehat{\mathbf{k}}]_2, [\widehat{\mathbf{k}}]_2) = [\widehat{\mathbf{k}}]_2/[\widehat{\mathbf{k}}]_2 = [\widehat{\mathbf{k}} - \widehat{\mathbf{k}}]_2$, where $\widehat{\mathbf{k}} \leftarrow_R \mathbb{Z}_p^{k+1}$, are uniformly distributed in $\mathbb{G}_2^{k+1}$.

▷ **Evaluation Invariance**: For any $[\widehat{\mathbf{k}}_1]_2, [\widehat{\mathbf{k}}_2]_2 \in \mathsf{H}.\mathcal{MSK}$, and any $\mathtt{ID} \in \mathcal{I}$, given $\mathsf{H.sk}_{\mathtt{ID}}[[\widehat{\mathbf{k}}_1]_2], \mathsf{H.sk}_{\mathtt{ID}}[[\widehat{\mathbf{k}}_2]_2]$, the master secret key part of $\mathsf{H.sk}_{\mathtt{ID}}[f([\widehat{\mathbf{k}}_1]_2, [\widehat{\mathbf{k}}_2]_2)] \leftarrow \mathsf{H.EvalMSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{\mathtt{ID}}[[\widehat{\mathbf{k}}_1]_2], \mathsf{H.sk}_{\mathtt{ID}}[[\widehat{\mathbf{k}}_2]_2], f)$ is $f([\widehat{\mathbf{k}}_1]_2, [\widehat{\mathbf{k}}_2]_2)$ for both $f \in \{\mathtt{mul}, \mathtt{div}\}$. Then, the distribution is the same as $\mathsf{H.GenSK}(\mathsf{H.MPK}, f([\widehat{\mathbf{k}}_1]_2, [\widehat{\mathbf{k}}_2]_2), \mathtt{ID})$ due to the uniformly random $\mathbf{r}' \in \mathbb{Z}_p^k$.

Remark 3 (Difficulty achieving MSK evaluatability from lattices and traditional groups)**.** *Obviously, it appears to be difficult for lattice-based HIBE schemes [ABB10a, ABB10b, CHKP12, Zha12] and the pairing-free HIBE scheme [DG17] to satisfy MSK evaluatability, which is the primary reason we exclusively focus on pairing-based instantiations.*

First-level Wildcarded SK. The CW HIBE can be modified easily such that it equips first-level wildcarded SK. Here, we define a secret key $\mathsf{H.sk}_{(*,\mathtt{ID})}[[\widehat{\mathbf{k}}]_2] := (\mathsf{sk}_{(*,\mathtt{ID}),0}, \mathsf{sk}_{(*,\mathtt{ID}),1}, (\mathsf{sk}_{(*,\mathtt{ID}),i})_{i\in[\ell+1,L]}, \mathsf{sk}_{(*,\mathtt{ID}),*}) \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, [\widehat{\mathbf{k}}]_2, (*, \mathtt{ID}))$ for $\mathtt{ID} = (\mathtt{id}_2, \ldots, \mathtt{id}_\ell)$ under a (pseudo-)master secret key $[\widehat{\mathbf{k}}]_2 \in \mathbb{G}_2^{k+1}$ as

- $\mathsf{sk}_{(*,\mathtt{ID}),0} = [\mathbf{B}\mathbf{r}]_2$, $\boxed{\mathsf{sk}_{(*,\mathtt{ID}),1} = [\widehat{\mathbf{k}}]_2 \cdot [(\mathtt{id}_2\mathbf{W}_2 + \cdots + \mathtt{id}_\ell\mathbf{W}_\ell + \mathbf{W}_{L+1})\mathbf{B}\mathbf{r}]_2,}$ $\mathsf{sk}_{(*,\mathtt{ID}),\ell+1} = [\mathbf{W}_{\ell+1}\mathbf{B}\mathbf{r}]_2$, …, $\mathsf{sk}_{(*,\mathtt{ID}),L} = [\mathbf{W}_L\mathbf{B}\mathbf{r}]_2$, $\boxed{\mathsf{sk}_{(*,\mathtt{ID}),*} = [\mathbf{W}_1\mathbf{B}\mathbf{r}]_2}$.

Note that $\mathsf{sk}_{(*,\mathtt{ID}),0}$ and $(\mathsf{sk}_{(*,\mathtt{ID}),i})_{i\in[\ell+1,L]}$ are the same as the original secret key. In addition, we can execute $\mathsf{H.GenSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(*,\mathtt{ID})}, \mathtt{ID}')$ by sampling $\mathbf{r}' \leftarrow_R \mathbb{Z}_p^k$ and computing

- $\mathsf{H.sk}_{\mathtt{ID}'} := (\mathsf{sk}_{\mathtt{ID}',0}, \mathsf{sk}_{\mathtt{ID}',1}, (\mathsf{sk}_{\mathtt{ID}',i})_{i\in[\ell+2,L]}, \mathsf{sk}_{\mathtt{ID}',*})$ if $\mathtt{ID}' = (*, \mathtt{ID}, \mathtt{id}_{\ell+1})$: $\mathsf{sk}_{\mathtt{ID}',1} = \mathsf{sk}_{(*,\mathtt{ID}),1} \cdot [(\mathbf{W}_{L+1} + \mathtt{id}_2\mathbf{W}_2 + \cdots + \mathtt{id}_\ell\mathbf{W}_\ell)\mathbf{B}\mathbf{r}']_2 \cdot (\mathsf{sk}_{(*,\mathtt{ID}),\ell+1} \cdot [\mathbf{W}_{\ell+1}\mathbf{B}\mathbf{r}']_2)^{\mathtt{id}_{\ell+1}}$, $\mathsf{sk}_{\mathtt{ID}',*} = \mathsf{sk}_{(*,\mathtt{ID}),*} \cdot [\mathbf{W}_1\mathbf{B}\mathbf{r}']_2$;
- $\mathsf{H.sk}_{\mathtt{ID}'} := (\mathsf{sk}_{\mathtt{ID}',0}, \mathsf{sk}_{\mathtt{ID}',1}, (\mathsf{sk}_{\mathtt{ID}',i})_{i\in[\ell+1,L]})$ if $\mathtt{ID}' = (\mathtt{id}_1, \mathtt{ID})$: $\mathsf{sk}_{\mathtt{ID}',1} = \mathsf{sk}_{(*,\mathtt{ID}),1} \cdot [(\mathbf{W}_{L+1} + \mathtt{id}_2\mathbf{W}_2 + \cdots + \mathtt{id}_\ell\mathbf{W}_\ell)\mathbf{B}\mathbf{r}']_2 \cdot (\mathsf{sk}_{(*,\mathtt{ID}),*} \cdot [\mathbf{W}_1\mathbf{B}\mathbf{r}']_2)^{\mathtt{id}_1}$,

where $\mathsf{sk}_{\mathtt{ID}',0}$ and $(\mathsf{sk}_{\mathtt{ID}',i})_{i\in[\ell+1,L]}$ are computed in the same manner as the original delegation procedure.

The above $\mathsf{sk}_{\mathtt{ID}}$ satisfies the correctness in Definition 3 due to the uniformly random $\mathbf{r}' \leftarrow_R \mathbb{Z}_p^k$. We demonstrate that CW HIBE permits a wildcarded secret key query in Definition 3, i.e., the reduction algorithm can change a wildcarded secret key $\mathsf{H.sk}_{(*,\mathtt{ID})}$ to semi-functional. Note that it is sufficient to demonstrate that the following distribution is a uniform distribution in $\mathbb{Z}_p^{L-|\mathtt{ID}|+3}$:

$$\{\zeta_1\mathtt{id}_1^\star + \cdots + \zeta_{|\mathtt{ID}^\star|}\mathtt{id}_{|\mathtt{ID}^\star|}^\star + \zeta_{L+1},\ \zeta_1, \zeta_2\mathtt{id}_2 + \cdots + \zeta_{|\mathtt{ID}|}\mathtt{id}_{|\mathtt{ID}|} + \zeta_{L+1}, (\zeta_i)_{i\in[|\mathtt{ID}|+1,L]}\},$$

where $\zeta_i \leftarrow_R \mathbb{Z}_p$ for all $i \in [L+1]$. Here, ζ_1 is a random semi-functional component of a new secret key element $\mathsf{sk}_{(*,\mathtt{ID}),*}$, and $\zeta_2\mathtt{id}_2 + \cdots + \zeta_{|\mathtt{ID}|}\mathtt{id}_{|\mathtt{ID}|} + \zeta_{L+1}$ is a random semi-functional component of a secret key element $\mathsf{sk}_{\mathtt{ID},1}$. Since $[\mathtt{id}_1\mathbf{W}_1\mathbf{Br}]_1$ is omitted, a random semi-functional component of $\mathsf{sk}_{\mathtt{ID},1}$ differs from that of an HIBE secret key $\mathsf{H.sk}_{(\mathtt{id}_1,\ldots,\mathtt{id}_{|\mathtt{ID}|})}$ by $\zeta_1\mathtt{id}_1$. The claim of uniformity holds thanks to the fact that $(*, \mathtt{id}_2, \ldots, \mathtt{id}_{|\mathtt{ID}|}) \notin \mathsf{W.prefix}^+(\mathtt{ID}^\star) \Leftrightarrow (\mathtt{id}_2, \ldots, \mathtt{id}_{|\mathtt{ID}|}) \notin \mathsf{H.prefix}^+((\mathtt{id}_2^\star, \ldots, \mathtt{id}_{|\mathtt{ID}^\star|}^\star))$.

Prefix Decryption Restriction. Recall that $\mathsf{sk}_{\mathtt{ID},|\mathtt{ID}|+1}, \ldots, \mathsf{sk}_{\mathtt{ID},L}$ are not used during decryption and only are responsible for key delegation; thus, we define the GenDK algorithm as follows:

- $\mathsf{H.GenDK}(\mathsf{H.MPK}, \mathsf{H.sk}_{\mathtt{ID}} = (\mathsf{sk}_{\mathtt{ID},0}, \mathsf{sk}_{\mathtt{ID},1}, (\mathsf{sk}_{\mathtt{ID},i})_{i\in[\ell+1,L]})) \to \mathsf{H.dk}_{\mathtt{ID}}$: It samples $\mathbf{r}' \leftarrow_R \mathbb{Z}_p^k$ and outputs $\mathsf{H.dk}_{\mathtt{ID}} = (\mathsf{H.dk}_{\mathtt{ID},0}, \mathsf{H.dk}_{\mathtt{ID},1})$ by computing $\mathsf{dk}_{\mathtt{ID},0} = \mathsf{sk}_{\mathtt{ID},0} \cdot [\mathbf{Br}']_2$, $\mathsf{dk}_{\mathtt{ID},1} = \mathsf{sk}_{\mathtt{ID},1} \cdot [(\mathbf{W}_{L+1} + \mathtt{id}_1\mathbf{W}_1 + \cdots + \mathtt{id}_{|\mathtt{ID}|}\mathbf{W}_{|\mathtt{ID}|})\mathbf{Br}']_2$.

Obviously, $\mathsf{H.dk}_{\mathtt{ID}}$ is able to decrypt $\mathsf{H.ct}_{\mathtt{ID}}$ correctly. In addition, the above $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ satisfies the correctness in Definition 4 thanks to the uniformly random $\mathbf{r}' \leftarrow_R \mathbb{Z}_p^k$. Finally, we demonstrate that CW HIBE allows the decryption key query in Definition 4, i.e., the reduction algorithm can change a decryption key $\mathsf{H.dk}_{\mathtt{ID}}$ to be semi-functional. Here, it is sufficient to demonstrate that the following distribution is a uniform distribution in $\mathbb{Z}_p^2$:

$$\{\zeta_0 + \zeta_1\mathtt{id}_1^\star + \cdots + \zeta_{|\mathtt{ID}^\star|}\mathtt{id}_{|\mathtt{ID}^\star|}^\star,\ \zeta_0 + \zeta_1\mathtt{id}_1 + \cdots + \zeta_{|\mathtt{ID}|}\mathtt{id}_{|\mathtt{ID}|}\},$$

where $\zeta_i \leftarrow_R \mathbb{Z}_p$ for all $i \in [\max\{|\mathtt{ID}^\star|, |\mathtt{ID}|\}]$. Note that there are no $(\zeta_i)_{i\in[|\mathtt{ID}|+1,L]}$ that are random semi-functional components of $(\mathsf{sk}_{\mathtt{ID},i})_{i\in[|\mathtt{ID}|+1,L]}$. Thus, the claim of uniformity holds because $\mathtt{ID} \neq \mathtt{ID}^\star$.

5 Construction

In this section, we present the proposed RHIBE scheme with depth L from an HIBE scheme with depth $L+1$ supporting *MSK evaluatability*, *first-level wildcarded SK*, and *prefix decryption restriction* using the CS method. Here, we assume $\mathcal{I} \subseteq \mathsf{H}.\mathcal{I}$ and $\mathcal{T} \subseteq \mathsf{H}.\mathcal{I}$. The master public key, ciphertexts, and decryption keys of the proposed RHIBE scheme are the same as those of the underlying HIBE scheme.

Complete Subtree Method. As claimed in Section 2.1 and Section 2.2, $\mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}}) \subset \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ and $\mathcal{AN}_{\mathsf{pa}(\mathtt{ID})} \subset \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ denote a path from the root node to a leaf node $\eta_{\mathtt{ID}}$ and a set of activated nodes θ associated with $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$ that have been used to create all $\mathsf{sk}_{\mathtt{ID},\theta}$ and/or $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$ at least once. For all parent users $\mathsf{pa}(\mathtt{ID}) \in \mathcal{I}^{\leq L-1}$ and kgc, we introduce the following notations to indicate the subsets of $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$: $\mathcal{L}_{\mathsf{pa}(\mathtt{ID})}$ is a set of all leaf nodes; $\mathcal{AL}_{\mathsf{pa}(\mathtt{ID})} \subset \mathcal{L}_{\mathsf{pa}(\mathtt{ID})}$ is a set of leaf nodes to which some identities have already been assigned; $\mathcal{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} \subset \mathcal{AL}_{\mathsf{pa}(\mathtt{ID})}$ is a set of leaf nodes for users revoked by a time period $\mathtt{T}$.

In this paper, we use four algorithms $(\mathsf{CS.SetUp}, \mathsf{CS.Assign}, \mathsf{CS.Cover}, \mathsf{CS.Match})$ to describe the CS method.

- $\mathsf{CS.SetUp}(1^\lambda, \mathsf{pa}(\mathtt{ID})) \rightarrow \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$: The setup algorithm takes the security parameter 1^λ and a parent identity $\mathsf{pa}(\mathtt{ID}) \in \mathcal{I}^{\leq L-1}$ as input, and outputs the description of a binary tree $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ for $\mathsf{pa}(\mathtt{ID})$.
- $\mathsf{CS.Assign}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \mathcal{AL}_{\mathsf{pa}(\mathtt{ID})}, \mathtt{ID}) \rightarrow (\eta_{\mathtt{ID}}, \mathcal{AL}'_{\mathsf{pa}(\mathtt{ID})})$: The assign algorithm takes binary tree $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$, a set of leaf nodes $\mathcal{AL}_{\mathsf{pa}(\mathtt{ID})}$, and an identity $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$, and samples a leaf node $\eta_{\mathtt{ID}} \leftarrow_R \mathcal{L}_{\mathsf{pa}(\mathtt{ID})} \setminus \mathcal{AL}_{\mathsf{pa}(\mathtt{ID})}$ uniformly at random. It assigns $\mathtt{ID}$ to $\eta_{\mathtt{ID}}$ and updates $\mathcal{AL}'_{\mathsf{pa}(\mathtt{ID})} \leftarrow \mathcal{AL}_{\mathsf{pa}(\mathtt{ID})} \cup \{\eta_{\mathtt{ID}}\}$. Finally, it outputs $\eta_{\mathtt{ID}}$ and $\mathcal{AL}'_{\mathsf{pa}(\mathtt{ID})}$.
- $\mathsf{CS.Cover}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \mathcal{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}) \rightarrow \mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$: The cover algorithm takes a binary tree $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ and a set of leaf nodes $\mathcal{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$, and outputs a set of nodes $\mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$.
- $\mathsf{CS.Match}(\mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}, \eta_{\mathtt{ID}}) \rightarrow \theta$ or $\perp$: The matching algorithm takes a set of nodes $\mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ output by $\mathsf{CS.Cover}$ and a leaf node $\eta_{\mathtt{ID}}$ as input, and outputs $\theta \in \mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} \cap \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})$ if such a node exists; otherwise, it outputs an invalid symbol $\perp$.

The CS method ensures that the $\mathsf{CS.Match}$ algorithm outputs θ and $\perp$ for all leaf nodes $\eta_{\mathtt{ID}} \in \mathcal{L}_{\mathsf{pa}(\mathtt{ID})} \setminus \mathcal{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$ and $\eta_{\mathtt{ID}} \in \mathcal{RL}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$, respectively. In addition, the CS method allows us to realize scalable revocation because $|\mathcal{N}_{\mathtt{ID}}| = O(|\mathcal{RL}_{\mathtt{ID}}| \log(|\mathcal{L}_{\mathtt{ID}}|/|\mathcal{RL}_{\mathtt{ID}}|))$ holds.

Construction. Note that $\mathsf{sk}_{\mathtt{ID},\theta}$, $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$, $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$, and $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ of the proposed RHIBE scheme are HIBE secret keys or an HIBE decryption key:

$$\begin{aligned} &\mathsf{sk}_{\mathtt{ID},\theta} = \mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}], && \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta} = \mathsf{H.sk}_{(*,\mathsf{pa}(\mathtt{ID}))}\left[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}\right], \\ &\mathsf{ku}_{\mathtt{ID},\mathtt{T},\theta} = \mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}], && \mathsf{dk}_{\mathtt{ID},\mathtt{T}} = \mathsf{H.dk}_{(\mathtt{T},\mathtt{ID})}. \end{aligned}$$

It is easy to verify that $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$ can derive $\mathsf{sk}_{\mathtt{ID},\theta}$, $\mathsf{sk}_{\mathtt{ID},\theta}$, $\mathsf{delk}_{\mathtt{ID},\theta}$, and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$ can derive $\mathsf{ku}_{\mathtt{ID},\mathtt{T},\theta}$, and $\mathsf{sk}_{\mathtt{ID},\theta}$ and $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta}$ can derive $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ due to MSK evaluatability, first-level wildcarded SK, and prefix decryption restriction. The proposed scheme is presented in the following.

- $\mathsf{Setup}(1^\lambda, L) \rightarrow (\mathsf{MPK}, \mathsf{sk}_{\mathsf{kgc}})$: Run $(\mathsf{H.MPK}, \mathsf{H.MSK}) \leftarrow \mathsf{H.Setup}(1^\lambda, L+1)$ and $\mathcal{BT}_{\mathsf{kgc}} \leftarrow \mathsf{CS.SetUp}(1^\lambda, \mathsf{kgc})$, then output $\mathsf{MPK} := \mathsf{H.MPK}$ and $\mathsf{sk}_{\mathsf{kgc}} := (\mathsf{H.MSK}, \mathcal{BT}_{\mathsf{kgc}})$.
- $\mathsf{Encrypt}(\mathsf{MPK}, \mathtt{ID}, \mathtt{T}, \mathsf{M}) \rightarrow \mathsf{ct}_{\mathtt{ID},\mathtt{T}}$: Parse $\mathsf{MPK} = \mathsf{H.MPK}$. Run
  - · $\mathsf{H.ct}_{(\mathtt{T},\mathtt{ID})} \leftarrow \mathsf{H.Encrypt}(\mathsf{H.MPK}, (\mathtt{T}, \mathtt{ID}), \mathsf{M})$,

  then output $\mathsf{ct}_{\mathtt{ID},\mathtt{T}} := \mathsf{H.ct}_{(\mathtt{T},\mathtt{ID})}$.
- $\mathsf{GenSK}(\mathsf{MPK}, \mathsf{sk}_{\mathsf{pa}(\mathtt{ID})}, \mathtt{ID}) \rightarrow (\mathsf{sk}_{\mathtt{ID}}, \mathsf{sk}'_{\mathsf{pa}(\mathtt{ID})})$: Parse $\mathsf{MPK} = \mathsf{H.MPK}$ and

$$\triangleright\ \mathsf{sk}_{\mathsf{pa}(\mathtt{ID})} = \begin{cases} (\mathsf{H.MSK}, \mathcal{BT}_{\mathsf{kgc}}, (\theta, \mathsf{delk}_{\mathsf{kgc},\theta})_{\theta \in \mathcal{AN}_{\mathsf{kgc}}}) & \text{if } \mathsf{pa}(\mathtt{ID}) = \mathsf{kgc}, \\ \begin{pmatrix} (\theta, \mathsf{sk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathsf{pa}(\mathtt{ID}))}, \eta_{\mathsf{pa}(\mathtt{ID})})}, \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \\ (\theta, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{AN}_{\mathsf{pa}(\mathtt{ID})}} \end{pmatrix} & \text{otherwise.} \end{cases}$$

  Proceed as follows.

  1 (**User Assignment**). Run $(\eta_{\mathtt{ID}}, \mathcal{BT}'_{\mathsf{pa}(\mathtt{ID})}) \leftarrow \mathsf{CS.Assign}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \mathtt{ID})$.

  2 (**Delegation Key Generation**). For each node $\theta \in \mathsf{Path}(\eta_{\mathtt{ID}}) \setminus \mathcal{AN}_{\mathsf{pa}(\mathtt{ID})}$, proceeds as follows.

  2-1 (**Pseudo-MSK Sampling**). Run
  - · $\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta} \leftarrow \mathsf{H.SampMSK}(\mathsf{H.MPK})$.

  2-2 (**Sub-delegation Key Generation**). If $\mathsf{pa}(\mathtt{ID}) = \mathsf{kgc}$, skip this step. Otherwise, run
  - · $\mathsf{H.sk}_{(*,\mathsf{pa}(\mathtt{ID}))}\left[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}\right] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}, (*, \mathsf{pa}(\mathtt{ID})))$.

2-3 (**Update**). Proceed as follows.

- $\cdot$ Set $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta} := \begin{cases} \widehat{\mathsf{H.MSK}}_{\mathsf{kgc},\theta} & \text{if } \mathsf{pa}(\mathtt{ID}) = \mathsf{kgc}, \\ \mathsf{H.sk}_{(*,\mathsf{pa}(\mathtt{ID}))}\Big[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}\Big] & \text{otherwise.} \end{cases}$
- $\cdot$ Update $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$ by $\mathcal{AN}'_{\mathsf{pa}(\mathtt{ID})} \leftarrow \mathcal{AN}_{\mathsf{pa}(\mathtt{ID})} \cup \{\theta\}$.

3 (**Sub-secret Key Generation**). For each $\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})$, parse

$\triangleright$ $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta} = \mathsf{H.sk}_{(*,\mathsf{pa}(\mathtt{ID}))}\Big[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}\Big]$,

and run

$\cdot$ $\mathsf{H.sk}_{(*,\mathtt{ID})}\Big[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}\Big] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(*,\mathsf{pa}(\mathtt{ID}))}\Big[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}\Big], (*, \mathtt{ID}))$,

then set $\mathsf{sk}_{\mathtt{ID},\theta} := \mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}]$.

Finally, run $\mathcal{BT}_{\mathtt{ID}} \leftarrow \mathsf{CS.SetUp}(1^{\lambda}, \mathtt{ID})$, and output $\mathsf{sk}_{\mathtt{ID}}$ and an updated $\mathsf{sk}'_{\mathsf{pa}(\mathtt{ID})}$, where

– $\mathsf{sk}_{\mathtt{ID}} := ((\theta, \mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})}, \mathcal{BT}_{\mathtt{ID}})$,

– $$\mathsf{sk}'_{\mathsf{pa}(\mathtt{ID})} = \begin{cases} (\mathsf{H.MSK}, \mathcal{BT}'_{\mathsf{kgc}}, (\theta, \mathsf{delk}_{\mathsf{kgc},\theta})_{\theta \in \mathcal{AN}'_{\mathsf{kgc}}}) & \text{if } \mathsf{pa}(\mathtt{ID}) = \mathsf{kgc}, \\ \left(\begin{array}{c} (\theta, \mathsf{sk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathsf{pa}(\mathtt{ID}))}, \eta_{\mathsf{pa}(\mathtt{ID})})}, \mathcal{BT}'_{\mathsf{pa}(\mathtt{ID})}, \\ (\theta, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{AN}'_{\mathsf{pa}(\mathtt{ID})}} \end{array} \right) & \text{otherwise.} \end{cases}$$

- $\mathsf{KeyUp}(\mathsf{MPK}, \mathtt{T}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{RL}_{\mathtt{ID},\mathtt{T}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}) \to (\mathsf{ku}_{\mathtt{ID},\mathtt{T}}, \mathsf{sk}'_{\mathtt{ID}})$: Parse $\mathsf{MPK} = \mathsf{H.MPK}$ and

  $\triangleright$ $\mathsf{sk}_{\mathtt{ID}} = \begin{cases} (\mathsf{H.MSK}, \mathcal{BT}_{\mathsf{kgc}}, (\theta, \mathsf{delk}_{\mathsf{kgc},\theta})_{\theta \in \mathcal{AN}_{\mathsf{kgc}}}) & \text{if } \mathtt{ID} = \mathsf{kgc}, \\ ((\theta, \mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})}, \mathcal{BT}_{\mathtt{ID}}, (\theta, \mathsf{delk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{AN}_{\mathtt{ID}}}) & \text{otherwise,} \end{cases}$

  $\triangleright$ $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} = \begin{cases} \bot & \text{if } \mathtt{ID} = \mathsf{kgc}, \\ (\theta, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}} & \text{otherwise.} \end{cases}$

  Proceeds as follows.

  1 (**Finding $\mathcal{N}_{\mathtt{ID},\mathtt{T}}$**). Run

  $\cdot$ $\mathcal{N}_{\mathtt{ID},\mathtt{T}} \leftarrow \mathsf{CS.Cover}(\mathcal{BT}_{\mathtt{ID}}, \mathcal{RL}_{\mathtt{ID},\mathtt{T}})$.

  2 (**Delegation Key Generation**). For each node $\theta \in \mathcal{N}_{\mathtt{ID},\mathtt{T}} \setminus \mathcal{AN}_{\mathtt{ID}}$, perform Steps 2-1, 2-2, and 2-3 of GenSK.

  3 (**Sub-key Update Generation**). Run $\mathsf{GenDK}(\mathsf{MPK}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}})$ until Step 2 to obtain $\mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\mathsf{H.MSK}]$. Then, for each $\theta \in \mathcal{N}_{\mathtt{ID},\mathtt{T}}$, parse

  $\triangleright$ $\mathsf{delk}_{\mathtt{ID},\theta} = \begin{cases} \widehat{\mathsf{H.MSK}}_{\mathsf{kgc},\theta} & \text{if } \mathtt{ID} = \mathsf{kgc}, \\ \mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}] & \text{otherwise.} \end{cases}$

  If $\mathtt{ID} = \mathsf{kgc}$, run

  $\cdot$ $\mathsf{H.sk}_{\mathtt{T}}\left[\frac{\mathsf{H.MSK}}{\widehat{\mathsf{H.MSK}}_{\mathsf{kgc},\theta}}\right] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \frac{\mathsf{H.MSK}}{\widehat{\mathsf{H.MSK}}_{\mathsf{kgc},\theta}}, \mathtt{T})$,

  and set $\mathsf{ku}_{\mathsf{kgc},\mathtt{T},\theta} := \mathsf{H.sk}_{\mathtt{T}}[\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathsf{kgc},\theta}]$. Otherwise, run

  $\cdot$ $\mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}], (\mathtt{T}, \mathtt{ID}))$,

  $\cdot$ $\mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}\left[\frac{\mathsf{H.MSK}}{\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}}\right]$

  $\leftarrow \mathsf{H.EvalMSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\mathsf{H.MSK}], \mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}], \mathtt{div})$,

  and set $\mathsf{ku}_{\mathtt{ID},\mathtt{T},\theta} := \mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}]$.

Finally, output $\mathsf{ku}_{\mathtt{ID},\mathtt{T}}$ and an updated secret key $\mathsf{sk}'_{\mathtt{ID}}$, where

- $\mathsf{ku}_{\mathtt{ID},\mathtt{T}} = (\theta, \mathsf{ku}_{\mathtt{ID},\mathtt{T},\theta})_{\theta \in \mathcal{N}_{\mathtt{ID},\mathtt{T}}}$,
- $\mathsf{sk}'_{\mathtt{ID}} = \begin{cases} (\mathsf{H.MSK}, \mathcal{BT}'_{\mathsf{kgc}}, (\theta, \mathsf{delk}_{\mathsf{kgc},\theta})_{\theta \in \mathcal{AN}'_{\mathsf{kgc}}}) & \text{if } \mathtt{ID} = \mathsf{kgc}, \\ ((\theta, \mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})}, \mathcal{BT}'_{\mathtt{ID}}, (\theta, \mathsf{delk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{AN}'_{\mathtt{ID}}}) & \text{otherwise.} \end{cases}$

- $\mathsf{GenDK}(\mathsf{MPK}, \mathsf{sk}_{\mathtt{ID}}, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}) \to \mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ or $\perp$: Parse $\mathsf{MPK} = \mathsf{H.MPK}$ and
  - ▷ $\mathsf{sk}_{\mathtt{ID}} = ((\theta, \mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})}, \mathcal{BT}_{\mathtt{ID}}, (\theta, \mathsf{delk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{AN}_{\mathtt{ID}}})$,
  - ▷ $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}} = (\theta, \mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta})_{\theta \in \mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}}$.

  Proceeds as follows.

  1 (**Finding the Common Node** θ). Run $\mathsf{CS.Match}(\mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}, \eta_{\mathtt{ID}})$ to find $\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}$. If it outputs $\perp$, then output $\perp$.

  2 (**Secret Key Generation under** MSK). Parse

  ▷ $\mathsf{sk}_{\mathtt{ID},\theta} = \mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}]$,

  ▷ $\mathsf{ku}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T},\theta} = \mathsf{H.sk}_{(\mathtt{T},\mathsf{pa}(\mathtt{ID}))}\left[\frac{\mathsf{H.MSK}}{\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}}\right]$.

  Compute $\mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\mathsf{H.MSK}]$ as follows:

  · $\mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}], (\mathtt{T}, \mathtt{ID}))$,

  · $\mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}\left[\frac{\mathsf{H.MSK}}{\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}}\right] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(\mathtt{T},\mathsf{pa}(\mathtt{ID}))}\left[\frac{\mathsf{H.MSK}}{\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}}\right], (\mathtt{T}, \mathtt{ID}))$,

  · $\mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\mathsf{H.MSK}]$

  $\leftarrow \mathsf{H.EvalMSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}], \mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}\left[\frac{\mathsf{H.MSK}}{\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}}\right], \mathtt{mul})$,

  3 (**Decryption Key Generation**). Run

  · $\mathsf{H.dk}_{(\mathtt{T},\mathtt{ID})} \leftarrow \mathsf{H.GenDK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(\mathtt{T},\mathtt{ID})}[\mathsf{H.MSK}])$.

  Finally, output $\mathsf{dk}_{\mathtt{ID},\mathtt{T}} := \mathsf{H.dk}_{(\mathtt{T},\mathtt{ID})}$.

$\mathsf{Decrypt}(\mathsf{MPK}, \mathsf{dk}_{\mathtt{ID},\mathtt{T}}, \mathsf{ct}_{\mathtt{ID},\mathtt{T}}) \to \mathsf{M}$: Parse $\mathsf{dk}_{\mathtt{ID},\mathtt{T}} = \mathsf{H.dk}_{(\mathtt{T},\mathtt{ID})}$ and $\mathsf{ct}_{\mathtt{ID},\mathtt{T}} = \mathsf{H.ct}_{(\mathtt{T},\mathtt{ID})}$. Run and output

· $\mathsf{M} \leftarrow \mathsf{H.Decrypt}(\mathsf{H.MPK}, \mathsf{H.dk}_{(\mathtt{T},\mathtt{ID})}, \mathsf{H.ct}_{(\mathtt{T},\mathtt{ID})})$.

Correctness. The correctness of the CS method ensures that a non-revoked user can find node $\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}),\eta_{\mathtt{ID}}}) \cap \mathcal{N}_{\mathsf{pa}(\mathtt{ID}),\mathtt{T}}$. Then, the correctness of HIBE and first-level wildcarded SK, the prefix decryption restriction, and evaluation invariance ensure that $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ is an HIBE decryption key $\mathsf{H.dk}_{(\mathtt{T},\mathtt{ID})}$ under $\mathsf{H.MSK}$. Again, the correctness of prefix decryption restriction ensures that the $\mathsf{H.Decrypt}$ algorithm outputs M correctly.

6 Security

In this section, we prove the following theorem.

Theorem 1. *If the underlying HIBE scheme with hierarchical depth $L+1$ supporting MSK evaluatability, first-level wildcarded SK, and prefix decryption restriction satisfies adaptive (resp. selective) security, then the RHIBE scheme with hierarchical depth L also satisfies adaptive (resp. selective) security. Specifically, if there exists an adversary $\mathcal{A}$ to break adaptive (resp. selective) security of the RHIBE scheme with advantage $\mathsf{Adv}^{\mathtt{RHIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$, then there exists a reduction algorithm $\mathcal{B}$ to break* *adaptive (resp. selective) security of the underlying HIBE scheme with advantage* $\mathsf{Adv}^{\mathtt{HIBE}}_{\mathsf{H}.\Pi,L+1,\mathcal{B}}(\lambda)$ *such that* $\mathsf{Adv}^{\mathtt{RHIBE}}_{\Pi,L,\mathcal{A}}(\lambda) \leq O(LQ) \cdot \mathsf{Adv}^{\mathtt{HIBE}}_{\mathsf{H}.\Pi,L+1,\mathcal{B}}(\lambda)$ *(resp.* $\mathsf{Adv}^{\mathtt{RHIBE}}_{\Pi,L,\mathcal{A}}(\lambda) \leq O(L) \cdot \mathsf{Adv}^{\mathtt{HIBE}}_{\mathsf{H}.\Pi,L+1,\mathcal{B}}(\lambda)$*), where Q denotes the number of $\mathcal{A}$'s secret key generation queries.*

Proof of Theorem 1. In the following, we prove the theorem for adaptive security. From this point forward, we divide $\mathcal{A}$'s attack strategy into $L+1$ types and consider the following *Type-$\ell^\star$ strategy* for each $\ell^\star \in [L+1]$.

Type-$\ell^\star$ strategy: For $\ell^\star \in [L]$, $\mathcal{A}$ makes a secret key reveal query on $\mathtt{ID}^\star_{[\ell^\star]}$ but not on any $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star_{[\ell^\star]}) \backslash \{\mathtt{ID}^\star_{[\ell^\star]}\}$ $(= \mathsf{prefix}^+(\mathtt{ID}^\star_{[\ell^\star-1]}))$. In addition, as a special case for $\ell^\star = L+1$, $\mathcal{A}$ does not make secret key reveal queries on any $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star)$.

To reduce the security of HIBE to that of RHIBE, the reduction algorithm $\mathcal{B}$ picks $\ell^\star \leftarrow_R [L+1]$ and assumes that $\mathcal{A}$ follows the attack strategy of Type-$\ell^\star$. If this condition does not hold, $\mathcal{B}$ aborts the game and outputs a random bit. Here, the reduction suffers from $O(L)$ reduction loss. In the following, we provide a proof against $\mathcal{A}$ of the Type-$\ell^\star$ strategy (denoted $\mathcal{A}_{\ell^\star}$) for a fixed $\ell^\star$.

Now, we employ $\mathcal{A}_{\ell^\star}$ as a building block and construct a reduction algorithm $\mathcal{B}_{\ell^\star}$ against an $(L+1)$-level HIBE scheme. Note that $\mathcal{B}_{\ell^\star}$ will set

$$(\mathtt{T}^\star, \mathtt{ID}^\star)$$

as the challenge identity in the HIBE security game; thus, $\mathcal{B}_{\ell^\star}$ does not make secret key reveal queries on $\mathtt{ID} \in \mathsf{W.prefix}^+(\mathtt{ID}^\star)$ during the HIBE security game. First, $\mathcal{B}_{\ell^\star}$ is given an HIBE's master public key $\mathsf{H.MPK}$ from an HIBE challenger $\mathcal{C}$. Whenever new pseudo-MSKs $\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}$ are generated, $\mathcal{B}_{\ell^\star}$ stores $(\theta, \widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta})$ in $\mathtt{MSKList}$. Then, $\mathcal{B}_{\ell^\star}$ executes $\mathcal{BT}_{\mathsf{kgc}} \leftarrow \mathsf{CS.SetUp}(1^\lambda, \mathsf{kgc})$, produces $\mathsf{ku}_{\mathsf{kgc},1}$, and sends $(\mathsf{MPK} = \mathsf{H.MPK}, \mathsf{ku}_{\mathsf{kgc},1})$ to $\mathcal{A}_{\ell^\star}$. Here, $\mathcal{B}_{\ell^\star}$ produces $\mathsf{ku}_{\mathsf{kgc},1}$ in the same way as in revoke & key update queries which we will explain later.

Let $Q_{\ell^\star}$ be the maximum number of secret key generation queries on level-$\ell^\star$ identities, and let $\mathtt{ID}_q \in \mathcal{I}^{\ell^\star}$ be an identity on which $\mathcal{A}_{\ell^\star}$ makes a q-th secret key generation query on level-$\ell^\star$ identities. Then, $\mathcal{B}_{\ell^\star}$ guesses[17] the number $Q^\star \in [Q_{\ell^\star}]$ such that $\mathtt{ID}_{Q^\star} = \mathtt{ID}^\star_{[\ell^\star]} \in \mathsf{prefix}^+(\mathtt{ID}^\star)$. If the guess is incorrect, $\mathcal{B}_{\ell^\star}$ outputs a random bit and aborts the game. The guess is correct with probability $1/Q_{\ell^\star} > 1/Q$. Note that we assume the guess is correct in the following.

$\mathcal{B}_{\ell^\star}$ answers $\mathcal{A}_{\ell^\star}$'s queries (including $\mathsf{ku}_{\mathsf{kgc},1}$) by interacting with $\mathcal{C}$ as follows:

Secret Key Generation Query: Upon the query, the challenger must perform three steps in the real security game, i.e., *user assignments*, *delegation key generations*, and *sub-secret key generations*. In the reduction, $\mathcal{B}_{\ell^\star}$, in turn, performs *modified user assignments*, *node division*, and *pseudo-MSK samplings* as follows:

1 (**Modified User Assignment**). Upon $\mathcal{A}_{\ell^\star}$'s query on a level-ℓ identity $\mathtt{ID}'$ for $\ell \neq \ell^\star$, $\mathcal{B}_{\ell^\star}$ runs $\eta_{\mathtt{ID}'} \leftarrow \mathsf{CS.Assign}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}')}, \mathtt{ID}')$ and assigns the user to leaf $\eta_{\mathtt{ID}'} \in \mathcal{L}_{\mathsf{pa}(\mathtt{ID}')}$ (as in the real scheme). In addition, if $\ell = \ell^\star - 1$, $\mathcal{B}_{\ell^\star}$ further samples a uniformly random leaf $\eta^\star_{\mathtt{ID}'} \leftarrow_R \mathcal{L}_{\mathtt{ID}'}$ in the level-$(\ell^\star - 1)$ user's binary tree $\mathcal{BT}_{\mathtt{ID}'}$ and updates $\mathcal{AL}_{\mathtt{ID}'} \leftarrow \{\eta^\star_{\mathtt{ID}'}\}$, even though there is there is no user assigned to the leaf. Then, $\mathcal{B}_{\ell^\star}$ modifies the way to assign level-$\ell^\star$ users $\mathtt{ID}_q$ to a leaf node in $\mathcal{L}_{\mathsf{pa}(\mathtt{ID}_q)}$ as follows:

$\mathtt{ID}_q$ **for** $q \in [Q^\star - 1]$: $\mathcal{B}_{\ell^\star}$ executes $\eta_{\mathtt{ID}_q} \leftarrow \mathsf{CS.Assign}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}_q)}, \mathtt{ID}_q)$ and assigns $\mathtt{ID}_q$ to a leaf $\eta_{\mathtt{ID}_q}$ and updates $\mathcal{AL}'_{\mathsf{pa}(\mathtt{ID}_q)} \leftarrow \mathcal{AL}_{\mathsf{pa}(\mathtt{ID}_q)} \cup \{\eta_{\mathtt{ID}_q}\}$. Although the procedure appears to be the same as that in the real scheme, $\mathtt{ID}_q$ is never assigned to leaf $\eta^\star_{\mathsf{pa}(\mathtt{ID}_q)}$ to which still has no assigned users.

[17] The guess is not required to prove selective security: thus, the reduction loss differs by a factor $O(Q)$.

$\mathtt{ID}_{Q^\star}$: $\mathcal{B}_{\ell^\star}$ assigns $\mathtt{ID}_{Q^\star}$ to the pre-sampled leaf $\eta_{\mathtt{ID}_{Q^\star}} = \eta^\star_{\mathsf{pa}(\mathtt{ID}_{Q^\star})} \in \mathcal{AL}_{\mathsf{pa}(\mathtt{ID}_{Q^\star})} \subset \mathcal{BT}_{\mathsf{pa}(\mathtt{ID}_{Q^\star})}$ and does not update $\mathcal{AL}_{\mathsf{pa}(\mathtt{ID}_{Q^\star})}$. Then, $\mathcal{B}_{\ell^\star}$ updates $\mathcal{AL}_{\mathtt{ID}'} \leftarrow \mathcal{AL}_{\mathtt{ID}'} \setminus \{\eta^\star_{\mathtt{ID}'}\}$ for all level-$(\ell^\star - 1)$ activated users $\mathtt{ID}'$ except $\mathsf{pa}(\mathtt{ID}_{Q^\star})$.

$\mathtt{ID}_q$ **for** $q \in [Q^\star + 1, Q_{\ell^\star}]$: $\mathcal{B}_{\ell^\star}$ executes $\eta_{\mathtt{ID}_q} \leftarrow \mathsf{CS.Assign}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}_q)}, \mathtt{ID}_q)$, assigns $\mathtt{ID}_q$ to a leaf $\eta_{\mathtt{ID}_q}$, and updates $\mathcal{AL}_{\mathsf{pa}(\mathtt{ID}_q)} \leftarrow \mathcal{AL}_{\mathsf{pa}(\mathtt{ID}_q)} \cup \{\eta_{\mathtt{ID}_q}\}$. In this case, $\mathtt{ID}_q$ may be assigned to leaf $\eta^\star_{\mathsf{pa}(\mathtt{ID}_q)}$ when $\mathsf{pa}(\mathtt{ID}_q) \neq \mathsf{pa}(\mathtt{ID}_{Q^\star})$.

Note that the modified user assignments are uniformly random from $\mathcal{A}_{\ell^\star}$'s view as in the real scheme.

2 (**Node Division**). $\mathcal{B}_{\ell^\star}$ divides all activated nodes managed in the reduction into two sets $\mathcal{SK}_{\ell^\star}$ and $\mathcal{KU}_{\ell^\star}$, as discussed in Section 2.4, both of which are initially empty sets. Upon $\mathcal{A}_{\ell^\star}$'s query, $\mathcal{B}_{\ell^\star}$ updates the sets as follows:

$$\mathcal{SK}_{\ell^\star} := \left\{ \theta : \begin{array}{c} \left(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})} \quad \text{for} \quad |\mathtt{ID}| \leq \ell^\star - 1\right) \vee \\ \left(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})} \setminus \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta^\star_{\mathsf{pa}(\mathtt{ID})}) \quad \text{for} \quad |\mathtt{ID}| = \ell^\star\right) \end{array} \right\},$$

$$\mathcal{KU}_{\ell^\star} := \mathcal{AN} \setminus \mathcal{SK}_{\ell^\star} = \left\{ \theta : \begin{array}{c} \left(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})} \quad \text{for} \quad |\mathtt{ID}| \geq \ell^\star + 1\right) \vee \\ \left(\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta^\star_{\mathsf{pa}(\mathtt{ID})}) \quad \text{for} \quad |\mathtt{ID}| = \ell^\star\right) \end{array} \right\}.$$

As a special case for $\ell^\star = L + 1$, $\mathcal{KU}_{L+1}$ is always an empty set, and $\mathcal{SK}_{L+1}$ denotes a set of all activated nodes managed in the reduction.

3 (**Pseudo-MSK Sampling**). $\mathcal{B}_{\ell^\star}$ executes

- $\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta} \leftarrow \mathsf{H.SampMSK}(\mathsf{H.MPK})$

and stores it in $\mathtt{MSKList}$.

If $\mathsf{pa}(\mathtt{ID}) = \mathsf{kgc}$, $\mathcal{B}_{\ell^\star}$ *implicitly* sets

$$\mathsf{delk}_{\mathsf{kgc},\theta} := \begin{cases} \mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathsf{kgc},\theta} & \text{for } \theta \in \mathcal{SK}_{\ell^\star}, \\ \widehat{\mathsf{H.MSK}}_{\mathsf{kgc},\theta} & \text{for } \theta \in \mathcal{KU}_{\ell^\star}. \end{cases}$$

Otherwise, $\mathcal{B}_{\ell^\star}$ does not create $\mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta}$ in this step.

Note that $\mathsf{delk}_{\mathsf{kgc},\theta}$ is sampled by executing the $\mathsf{H.SampMSK}$ algorithm in the real scheme. Here, $\mathsf{delk}_{\mathsf{kgc},\theta}$ follows the same distribution as in the real scheme due to the *pseudo-MSK indistinguishability* in Definition 2.

Finally, $\mathcal{B}_{\ell^\star}$ returns $\mathsf{ku}_{\mathtt{ID},\mathsf{T_{cu}}}$ to $\mathcal{A}_{\ell^\star}$, where $\mathcal{B}_{\ell^\star}$ creates $\mathsf{ku}_{\mathtt{ID},\mathsf{T_{cu}}}$ in the same manner as the revoke & key update query, which is explained later.

Secret Key Reveal Query: In the real security game, upon a query $\mathtt{ID}$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ finds $(\mathtt{ID}, \mathsf{sk}_{\mathtt{ID}}) \in \mathtt{SKList}$ and gives $\mathsf{sk}_{\mathtt{ID}}$ to $\mathcal{A}_{\ell^\star}$. In the reduction, $\mathcal{B}_{\ell^\star}$ performs *sub-secret key generations* and *delegation key generations* as follows:

1 (**Sub-secret Key Generation**). $\mathcal{B}_{\ell^\star}$ creates sub-secret keys $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})}$. For all nodes $\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta_{\mathtt{ID}})$, $\mathcal{B}_{\ell^\star}$ creates $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{SK}_{\ell^\star}}$ and $(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{KU}_{\ell^\star}}$ in distinct manners:

$(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta \in \mathcal{SK}_{\ell^\star}}$: $\mathcal{B}_{\ell^\star}$ finds $\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}$ from $\mathtt{MSKList}$ and makes an HIBE secret key reveal query on $(*, \mathtt{ID})$ to $\mathcal{C}$ and receives $\mathsf{H.sk}_{(*,\mathtt{ID})}$. Then, $\mathcal{B}_{\ell^\star}$ executes

- $\mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}, (*, \mathtt{ID}))$,
- $\mathsf{H.sk}_{(*,\mathtt{ID})}[\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}] \leftarrow$
  $\mathsf{H.EvalMSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(*,\mathtt{ID})}, \mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}], \mathtt{div})$.

Finally, $\mathcal{B}_{\ell^\star}$ sets $\mathsf{sk}_{\mathtt{ID},\theta} := \mathsf{H.sk}_{(*,\mathtt{ID})}[\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}]$ and stores it in $\mathtt{SKList}$.

Note that $\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}$ and $\mathsf{sk}_{\mathtt{ID},\theta}$ follow the same distribution as in the real scheme due to *pseudo-MSK indistinguishability* and *evaluation invariance* in Definition 2, respectively.

$(\mathsf{sk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$: $\mathcal{B}_{\ell^\star}$ finds $\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}$ from $\mathtt{MSKList}$ and executes

– $\mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}, (*,\mathtt{ID}))$.

Finally, $\mathcal{B}_{\ell^\star}$ sets $\mathsf{sk}_{\mathtt{ID},\theta} := \mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}]$ and stores it in $\mathtt{SKList}$.

Here, $\mathsf{sk}_{\mathtt{ID},\theta}$ follows the same distribution as in the real scheme due to the correctness of HIBE.

2 (**Delegation Key Generation**). $\mathcal{B}_{\ell^\star}$ creates delegation keys $(\mathsf{delk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{AN}_{\mathtt{ID}}}$. Here, for all nodes $\theta \in \mathcal{AN}_{\mathtt{ID}}$ such that $\mathsf{delk}_{\mathtt{ID},\theta}$ do not exist, $\mathcal{B}_{\ell^\star}$ creates $(\mathsf{delk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ and $(\mathsf{delk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$ in distinct manners:

$(\mathsf{delk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$: $\mathcal{B}_{\ell^\star}$ finds $\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}$ from $\mathtt{MSKList}$ and makes an HIBE secret key reveal query on $(*,\mathtt{ID})$ to $\mathcal{C}$ and receives $\mathsf{H.sk}_{(*,\mathtt{ID})}$. Then, $\mathcal{B}_{\ell^\star}$ executes

– $\mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}, (*,\mathtt{ID}))$,

– $\mathsf{H.sk}_{(*,\mathtt{ID})}[\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}] \leftarrow$
$\mathsf{H.EvalMSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(*,\mathtt{ID})}, \mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}], \mathtt{div})$.

Finally, $\mathcal{B}_{\ell^\star}$ sets $\mathsf{delk}_{\mathtt{ID},\theta} := \mathsf{H.sk}_{(*,\mathtt{ID})}[\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}]$.

Here, $\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}$ and $\mathsf{delk}_{\mathtt{ID},\theta}$ follow the same distribution as in the real scheme due to the *pseudo-MSK indistinguishability* and *evaluation invariance* in Definition 2, respectively.

$(\mathsf{delk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$: $\mathcal{B}_{\ell^\star}$ finds $\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}$ from $\mathtt{MSKList}$ and executes

– $\mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}, (*,\mathtt{ID}))$.

Finally, $\mathcal{B}_{\ell^\star}$ sets $\mathsf{delk}_{\mathtt{ID},\theta} := \mathsf{H.sk}_{(*,\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}]$.

Note that $\mathsf{delk}_{\mathtt{ID},\theta}$ follows the same distribution as in the real scheme due to the correctness of HIBE.

Finally, $\mathcal{B}_{\ell^\star}$ returns $\mathsf{sk}_{\mathtt{ID}} = ((\theta, \mathsf{sk}_{\mathtt{ID},\theta})_{\theta\in\mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})},\eta_{\mathtt{ID}})}, \mathcal{BT}_{\mathtt{ID}}, (\theta, \mathsf{delk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{AN}_{\mathtt{ID}}})$ to $\mathcal{A}_{\ell^\star}$.

Revoke & Key Update Query: Upon a query RL from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ checks if all conditions of the query are satisfied simultaneously. If the output is not $\perp$, $\mathcal{B}_{\ell^\star}$ updates each revocation list $\mathsf{RL}_{\mathtt{ID},\mathrm{T}}$ in the same manner as the real security game. Then, the challenger performs three steps in the real security game, i.e., *finding* $\mathcal{N}_{\mathtt{ID},\mathrm{T}}$, *delegation key generations*, and *sub-key update generations*. In the reduction, $\mathcal{B}_{\ell^\star}$ performs *finding* $\mathcal{N}_{\mathtt{ID},\mathrm{T}}$, *pseudo-MSK samplings*, and *sub-key update generations* as follows:

1 (**Finding** $\mathcal{N}_{\mathtt{ID},\mathrm{T}}$). $\mathcal{B}_{\ell^\star}$ executes

· $\mathcal{N}_{\mathtt{ID},\mathrm{T}} \leftarrow \mathsf{CS.Cover}(\mathcal{BT}_{\mathtt{ID}}, \mathcal{RL}_{\mathtt{ID},\mathrm{T}})$

as the real scheme.

2 (**Pseudo-MSK Sampling**). For each node $\theta \in \mathcal{N}_{\mathtt{ID},\mathrm{T}} \setminus \mathcal{AN}_{\mathtt{ID}}$, $\mathcal{B}_{\ell^\star}$ performs the pseudo-MSK sampling in the same manner as for secret key generation queries.

3 (**Sub-key Update Generation**). $\mathcal{B}_{\ell^\star}$ creates $(\mathsf{ku}_{\mathtt{ID},\mathrm{T}_{\mathsf{cu}},\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ and $(\mathsf{ku}_{\mathtt{ID},\mathrm{T}_{\mathsf{cu}},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$ in distinct manners:

$(\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}},\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$: $\mathcal{B}_{\ell^\star}$ finds $\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}$ from $\mathtt{MSKList}$ and runs

- $\mathsf{H.sk}_{(\mathtt{T}_{\mathtt{cu}},\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}, (\mathtt{T}_{\mathtt{cu}}, \mathtt{ID}))$.

Finally, $\mathcal{B}_{\ell^\star}$ sets $\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}},\theta} := \mathsf{H.sk}_{(\mathtt{T}_{\mathtt{cu}},\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}]$.

Note that $\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}$ and $\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}},\theta}$ follow the same distribution as those in the real scheme due to the *pseudo-MSK indistinguishability* in Definition 2 and the correctness of HIBE, respectively.

$(\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$: $\mathcal{B}_{\ell^\star}$ finds $\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}$ from $\mathtt{MSKList}$ and makes an HIBE secret key reveal query on $(\mathtt{T}_{\mathtt{cu}}, \mathtt{ID})$ to $\mathcal{C}$ and receives $\mathsf{H.sk}_{(\mathtt{T}_{\mathtt{cu}},\mathtt{ID})}$. Then, $\mathcal{B}_{\ell^\star}$ executes

- $\mathsf{H.sk}_{(\mathtt{T}_{\mathtt{cu}},\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}] \leftarrow \mathsf{H.GenSK}(\mathsf{H.MPK}, \widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}, (\mathtt{T}_{\mathtt{cu}}, \mathtt{ID}))$,
- $\mathsf{H.sk}_{(\mathtt{T}_{\mathtt{cu}},\mathtt{ID})}[\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}] \leftarrow$
  $\mathsf{H.EvalMSK}(\mathsf{H.MPK}, \mathsf{H.sk}_{(\mathtt{T}_{\mathtt{cu}},\mathtt{ID})}, \mathsf{H.sk}_{(\mathtt{T}_{\mathtt{cu}},\mathtt{ID})}[\widehat{\mathsf{H.MSK}}_{\mathsf{pa}(\mathtt{ID}),\theta}], \mathtt{div})$.

Finally, $\mathcal{B}_{\ell^\star}$ sets $\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}},\theta} := \mathsf{H.sk}_{(\mathtt{T}_{\mathtt{cu}},\mathtt{ID})}[\mathsf{H.MSK}/\widehat{\mathsf{H.MSK}}_{\mathtt{ID},\theta}]$.

Note that $\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}},\theta}$ follows the same distribution as that in the real scheme due the *evaluation invariance* in Definition 2.

Finally, $\mathcal{B}_{\ell^\star}$ returns the generated $\{\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}}}\}_{(\mathtt{ID},*)\in\mathtt{SKList}\setminus\mathsf{RL}}$ to $\mathcal{A}_{\ell^\star}$.

Decryption Key Reveal Query: Upon query $(\mathtt{ID}, \mathtt{T})$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ checks if all conditions of the query are satisfied simultaneously. If the output is not $\perp$, $\mathcal{B}_{\ell^\star}$ makes an HIBE decryption key reveal query on $(\mathtt{T}, \mathtt{ID})$ to $\mathcal{C}$ and receives $\mathsf{H.dk}_{(\mathtt{T},\mathtt{ID})}$. Finally, $\mathcal{B}_{\ell^\star}$ sets $\mathsf{dk}_{\mathtt{ID},\mathtt{T}} := \mathsf{H.dk}_{(\mathtt{T},\mathtt{ID})}$ and returns it to $\mathcal{A}_{\ell^\star}$.

Note that $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ follows the same distribution as that in the real scheme due to the correctness of HIBE.

Challenge Query: Upon a query $(\mathtt{ID}^\star, \mathtt{T}^\star, \mathsf{M}_0^\star, \mathsf{M}_1^\star)$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ check if all conditions of the query are satisfied simultaneously. If the output is not $\perp$, $\mathcal{B}_{\ell^\star}$ makes an HIBE challenge query on $((\mathtt{T}^\star, \mathtt{ID}^\star), \mathsf{M}_0^\star, \mathsf{M}_1^\star)$ to $\mathcal{C}$ and receives HIBE challenge ciphertext $\mathsf{H.ct}_{(\mathtt{T}^\star,\mathtt{ID}^\star)}$. Then, $\mathcal{B}_{\ell^\star}$ sends $\mathsf{H.ct}^\star := \mathsf{H.ct}_{(\mathtt{T}^\star,\mathtt{ID}^\star)}$ to $\mathcal{A}_{\ell^\star}$ as an RHIBE challenge ciphertext.

Note that $\mathsf{H.ct}^\star$ is created in the same manner as the real scheme.

After $\mathcal{B}_{\ell^\star}$ receives a bit b' from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ sends $\beta' \leftarrow b'$ to $\mathcal{C}$ as its own guess.

The above completes the description of $\mathcal{B}_{\ell^\star}$. Note that $\mathcal{B}_{\ell^\star}$ can make all of the above queries to $\mathcal{C}$ without making HIBE secret key queries on $\mathtt{ID} \in \mathsf{W.prefix}^+((\mathtt{T}^\star, \mathtt{ID}^\star))$ and HIBE decryption key reveal queries on $(\mathtt{T}^\star, \mathtt{ID}^\star)$. We observe that $\mathcal{B}_{\ell^\star}$ makes HIBE secret key reveal queries and decryption key reveal queries to $\mathcal{C}$ in the following cases.

- HIBE secret key reveal queries on $(*, \mathtt{ID})$ for $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{delk}_{\mathtt{ID},\theta})_{\theta\in\mathcal{SK}_{\ell^\star}}$ to answer RHIBE secret key reveal queries.
- HIBE secret key reveal queries on $(\mathtt{T}_{\mathtt{cu}}, \mathtt{ID})$ for $(\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}},\theta})_{\theta\in\mathcal{KU}_{\ell^\star}}$ to answer revoke & key update queries.
- HIBE decryption key reveal queries on $(\mathtt{T}, \mathtt{ID})$ for $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ to answer RHIBE decryption key reveal queries.

Next, we observe that the node division during secret key generation queries satisfies the following properties.

- All $(\mathsf{sk}_{\mathtt{ID},\theta}, \mathsf{delk}_{\mathsf{pa}(\mathtt{ID}),\theta})_{\theta \in \mathcal{SK}_{\ell^\star}}$ received by $\mathcal{A}_{\ell^\star}$ satisfy $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$. Thus, $(*, \mathtt{ID}) \notin \mathsf{W.prefix}^+((\mathtt{T}^\star, \mathtt{ID}^\star))$ hold. We explain why this holds as follows.
  - From the definition of the Type-$\ell^\star$ strategy, $\mathtt{ID} \in \mathcal{I}^{\leq \ell^\star - 1}$ on which $\mathcal{A}_{\ell^\star}$ makes secret key reveal queries satisfy $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$ because $\mathcal{A}_{\ell^\star}$ does not make the queries on $\mathtt{ID}^\star_{[\ell]}$ for $\ell \in [\ell^\star - 1]$.
  - From the definition of $\mathcal{SK}_{\ell^\star}$ and the assumption that $\mathtt{ID}_{Q^\star} = \mathtt{ID}^\star_{[\ell^\star]}$, all $\mathsf{sk}_{\mathtt{ID},\theta}$ of $\mathtt{ID} \in \mathcal{I}^{\ell^\star}$ on which $\mathcal{A}_{\ell^\star}$ makes secret key reveal queries satisfy $\theta \in \mathcal{SK}_{\ell^\star}$ only when $\mathtt{ID} \neq \mathtt{ID}^\star_{[\ell^\star]}$ because $\mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell^\star]})}, \eta_{\mathtt{ID}^\star_{[\ell^\star]}}) \cap \mathcal{SK}_{\ell^\star} = \emptyset$.
  - From the definition of $\mathcal{SK}_{\ell^\star}$, all $\mathsf{sk}_{\mathtt{ID},\theta}$ of $\mathtt{ID} \in \mathcal{I}^{\geq \ell^\star + 1}$ and $\mathsf{delk}_{\mathtt{ID},\theta}$ of $\mathtt{ID} \in \mathcal{I}^{\geq \ell^\star}$ on which $\mathcal{A}_{\ell^\star}$ makes secret key reveal queries satisfy $\theta \notin \mathcal{SK}_{\ell^\star}$ because all associated nodes belong to $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}')}$ for some $|\mathtt{ID}'| \geq \ell^\star + 1$.
- All $(\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}},\theta})_{\theta \in \mathcal{KU}_{\ell^\star}}$ received by $\mathcal{A}_{\ell^\star}$ satisfy $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star) \vee \mathtt{T} \neq \mathtt{T}^\star$. Thus, $(\mathtt{T}_{\mathtt{cu}}, \mathtt{ID}) \notin \mathsf{W.prefix}^+((\mathtt{T}^\star, \mathtt{ID}^\star))$ hold. We explain why this holds as follows.
  - From the definition of $\mathcal{KU}_{\ell^\star}$, all $\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}},\theta}$ of $\mathtt{ID} \in \mathcal{I}^{\leq \ell^\star - 1}$ received by $\mathcal{A}_{\ell^\star}$ satisfy $\theta \notin \mathcal{KU}_{\ell^\star}$ because the associated nodes belong to $\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}')}$ for some $|\mathtt{ID}'| \leq \ell^\star - 1$.
  - All $\mathsf{ku}_{\mathtt{ID},\mathtt{T}_{\mathtt{cu}},\theta}$ of $\mathtt{ID} \in \mathcal{I}^{\geq \ell^\star}$ received by $\mathcal{A}_{\ell^\star}$ satisfy $\theta \in \mathcal{KU}_{\ell^\star}$ only when $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star) \vee \mathtt{T} \neq \mathtt{T}^\star$. In particular, if $\mathtt{ID} \in \mathsf{prefix}^+(\mathtt{ID}^\star) \cap \mathcal{I}^{\geq \ell^\star}$, $\mathtt{ID}^\star_{[\ell^\star]} \in \mathsf{prefix}^+(\mathtt{ID})$. From the definitions of the Type-$\ell^\star$ strategy and the security of RHIBE, $\mathcal{A}_{\ell^\star}$ makes a secret key reveal query on $\mathtt{ID}^\star_{[\ell^\star]}$; therefore, all $\mathtt{ID}$ such that $\mathtt{ID}^\star_{[\ell^\star]} \in \mathsf{prefix}^+(\mathtt{ID})$ are revoked by $\mathtt{T}^\star$. Then, from the definition of $\mathcal{KU}_{\ell^\star}$ and the assumption that $\mathtt{ID}_{Q^\star} = \mathtt{ID}^\star_{[\ell^\star]}$, $\mathtt{T} \neq \mathtt{T}^\star$ holds.
- From the definition of RHIBE, $\mathsf{dk}_{\mathtt{ID},\mathtt{T}}$ received by $\mathcal{A}_{\ell^\star}$ satisfy $(\mathtt{ID}, \mathtt{T}) \neq (\mathtt{ID}^\star, \mathtt{T}^\star)$.

As observed previously, $\mathcal{B}_{\ell^\star}$ perfectly simulates the adaptive security game against $\mathcal{A}_{\ell^\star}$ with probability $1/Q$. Here, the probability that β' is a correct guess is the same as that of b'; thus, $\mathcal{B}_{\ell^\star}$'s advantage is $\mathsf{Adv}^{\mathtt{RHIBE}}_{\Pi,L,\mathcal{A}_{\ell^\star}}(\lambda) \leq O(Q) \cdot \mathsf{Adv}^{\mathtt{HIBE}}_{\Pi,L+1,\mathcal{B}_{\ell^\star}}(\lambda)$. Therefore, $\mathcal{B}$'s advantage against $\mathcal{A}$ for a general attack strategy is $\mathsf{Adv}^{\mathtt{RHIBE}}_{\Pi,L,\mathcal{A}}(\lambda) \leq \sum_{\ell^\star \in [L+1]} O(Q) \cdot \mathsf{Adv}^{\mathtt{HIBE}}_{\Pi,L+1,\mathcal{B}_{\ell^\star}}(\lambda) \leq O(LQ) \cdot \mathsf{Adv}^{\mathtt{HIBE}}_{\Pi,L+1,\mathcal{B}}(\lambda)$. □

7 Conclusion

In this paper, we have proposed the first adaptively secure RHIBE schemes in the standard model under the standard k-linear assumption. The achievement is significant because all known RHIBE schemes in the standard model [ESY16, KMT19, WZH+19, LP18, RLPL15, SE15b] only achieve selective security. In addition, the security of most known RHIBE schemes over bilinear groups [LP18, RLPL15, SE15b] are based on q-type assumptions. We obtain the proposed scheme by developing a generic construction of RHIBE from HIBE with mild requirements that are satisfied by several state-of-the-art pairing-based schemes [CG17, CW14, GCTC16]. Moreover, the proposed scheme is attractive even in a non-hierarchical case. Although the security of the currently most efficient adaptively secure RIBE schemes [GW19, WES17] is based on a non-standard assumption, the proposed RIBE scheme instantiated by [CG17] is secure under the standard k-linear assumption and achieves similar efficiency as we summarized in Table 2. It has to be an interesting open problem for constructing more efficient adaptively secure RHIBE schemes.

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference*, volume 6223 of *Lecture Notes in Computer Science*, pages 98–115. Springer, 2010.

[ABC$^+$11] Michel Abdalla, James Birkett, Dario Catalano, Alexander W. Dent, John Malone-Lee, Gregory Neven, Jacob C. N. Schuldt, and Nigel P. Smart. Wildcarded identity-based encryption. *J. Cryptology*, 24(1):42–82, 2011.

[AKN07] Michel Abdalla, Eike Kiltz, and Gregory Neven. Generalized key delegation for hierarchical identity-based encryption. In Joachim Biskup and Javier López, editors, *Computer Security - ESORICS 2007, 12th European Symposium On Research In Computer Security, Proceedings*, volume 4734 of *Lecture Notes in Computer Science*, pages 139–154. Springer, 2007.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2005.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[BGK08] Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. Identity-based encryption with efficient revocation. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008*, pages 417–426. ACM, 2008.

[CCKS18] Donghoon Chang, Amit Kumar Chauhan, Sandeep Kumar, and Somitra Kumar Sanadhya. Revocable identity-based encryption from codes with rank metric. In Nigel P. Smart, editor, *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018*, volume 10808 of *Lecture Notes in Computer Science*, pages 435–451. Springer, 2018.

[CG17] Jie Chen and Junqing Gong. ABE with tag made easy - concise framework and new instantiations in prime-order groups. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security. Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 35–65. Springer, 2017.

[CGW15] Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 9057 of *Lecture Notes in Computer Science*, pages 595–624. Springer, 2015.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012.

[CLL$^+$12] Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Khoa Nguyen. Revocable identity-based encryption from lattices. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *Information Security and Privacy - 17th Australasian Conference, ACISP 2012*, volume 7372 of *Lecture Notes in Computer Science*, pages 390–403. Springer, 2012.

[CW14] Jie Chen and Hoeteck Wee. Dual system groups and its applications - compact HIBE and more. *IACR Cryptology ePrint Archive*, 2014:265, 2014.

[DG17] Nico Döttling and Sanjam Garg. From selective IBE to full IBE and selective HIBE. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography - 15th International Conference, TCC 2017*, volume 10677 of *Lecture Notes in Computer Science*, pages 372–408. Springer, 2017.

[EHK$^+$17] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Luis Villar. An algebraic framework for Diffie-Hellman assumptions. *J. Cryptology*, 30(1):242–288, 2017.

[ESY16] Keita Emura, Jae Hong Seo, and Taek-Young Youn. Semi-generic transformation of revocable hierarchical identity-based encryption and its DBDH instantiation. *IEICE Transactions*, 99-A(1):83–91, 2016.

[GCTC16] Junqing Gong, Zhenfu Cao, Shaohua Tang, and Jie Chen. Extended dual system group and shorter unbounded hierarchical identity based encryption. *Des. Codes Cryptography*, 80(3):525–559, 2016.

[GW19] Aijun Ge and Puwen Wei. Identity-based broadcast encryption with efficient revocation. In Dongdai Lin and Kazue Sako, editors, *Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Proceedings, Part I*, volume 11442 of *Lecture Notes in Computer Science*, pages 405–435. Springer, 2019.

[HLCL18] Ziyuan Hu, Shengli Liu, Kefei Chen, and Joseph K. Liu. Revocable identity-based encryption from the computational Diffie-Hellman problem. In Willy Susilo and Guomin Yang, editors, *Information Security and Privacy - 23rd Australasian Conference, ACISP 2018, Proceedings*, volume 10946 of *Lecture Notes in Computer Science*, pages 265–283. Springer, 2018.

[ISW17] Yuu Ishida, Junji Shikata, and Yohei Watanabe. CCA-secure revocable identity-based encryption schemes with decryption key exposure resistance. *IJACT*, 3(3):288–311, 2017.

[JR17] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. *J. Cryptology*, 30(4):1116–1156, 2017.

[KMT19] Shuichi Katsumata, Takahiro Matsuda, and Atsushi Takayasu. Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance. In Dongdai Lin and Kazue Sako, editors, *Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Proceedings, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 441–471. Springer, 2019.

[Lee16] Kwangsu Lee. Revocable hierarchical identity-based encryption with adaptive security. *IACR Cryptology ePrint Archive*, 2016:749, 2016.

[Lee19] Kwangsu Lee. A generic construction for revocable identity-based encryption with subset difference methods. *IACR Cryptology ePrint Archive*, 2019:798, 2019.

[Lew12] Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 318–335. Springer, 2012.

[LLP17] Kwangsu Lee, Dong Hoon Lee, and Jong Hwan Park. Efficient revocable identity-based encryption via subset difference methods. *Des. Codes Cryptography*, 85(1):39–76, 2017.

[LP18] Kwangsu Lee and Seunghwan Park. Revocable hierarchical identity-based encryption with shorter private keys and update keys. *Des. Codes Cryptography*, 86(10):2407–2440, 2018.

[LP19] Roman Langrehr and Jiaxin Pan. Tightly secure hierarchical identity-based encryption. In Dongdai Lin and Kazue Sako, editors, *Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Proceedings, Part I*, volume 11442 of *Lecture Notes in Computer Science*, pages 436–465. Springer, 2019.

[LP20] Roman Langrehr and Jiaxin Pan. Hierarchical identity-based encryption with tight multi-challenge security. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Proceedings, Part I*, volume 12110 of *Lecture Notes in Computer Science*, pages 153–183. Springer, 2020.

[LV09] Benoît Libert and Damien Vergnaud. Adaptive-ID secure revocable identity-based encryption. In Marc Fischlin, editor, *Topics in Cryptology - CT-RSA 2009, The Cryptographers' Track at the RSA Conference 2009. Proceedings*, volume 5473 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2009.

[LW10] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 455–479. Springer, 2010.

[LW11] Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 547–567. Springer, 2011.

[ML19a] Xuecheng Ma and Dongdai Lin. Generic constructions of revocable identity-based encryption. In Zhe Liu and Moti Yung, editors, *Information Security and Cryptology - 15th International Conference, Inscrypt 2019*, volume 12020 of *Lecture Notes in Computer Science*, pages 381–396. Springer, 2019.

[ML19b] Xuecheng Ma and Dongdai Lin. Generic constructions of ribe via subset difference method. *IACR Cryptology ePrint Archive*, 2019:1376, 2019.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference. Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.

[OT15] Tatsuaki Okamoto and Katsuyuki Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. *Des. Codes Cryptography*, 77(2-3):725–771, 2015.

[RLPL15] Geumsook Ryu, Kwangsu Lee, Seunghwan Park, and Dong Hoon Lee. Unbounded hierarchical identity-based encryption with efficient revocation. In Howon Kim and Dooho Choi, editors, *Information Security Applications - 16th International Workshop, WISA 2015*, volume 9503 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 2015.

[RS14] Somindu C. Ramanna and Palash Sarkar. Efficient (anonymous) compact HIBE from standard assumptions. In Sherman S. M. Chow, Joseph K. Liu, Lucas Chi Kwong Hui, and Siu-Ming Yiu, editors, *Provable Security - 8th International Conference, ProvSec 2014. Proceedings*, volume 8782 of *Lecture Notes in Computer Science*, pages 243–258. Springer, 2014.

[SE13a] Jae Hong Seo and Keita Emura. Efficient delegation of key generation and revocation functionalities in identity-based encryption. In Ed Dawson, editor, *Topics in Cryptology - CT-RSA 2013 - The Cryptographers' Track at the RSA Conference 2013*, volume 7779 of *Lecture Notes in Computer Science*, pages 343–358. Springer, 2013.

[SE13b] Jae Hong Seo and Keita Emura. Revocable identity-based encryption revisited: Security model and construction. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography. Proceedings*, volume 7778 of *Lecture Notes in Computer Science*, pages 216–234. Springer, 2013.

[SE15a] Jae Hong Seo and Keita Emura. Adaptive-id secure revocable hierarchical identity-based encryption. In Keisuke Tanaka and Yuji Suga, editors, *Advances in Information and Computer Security - 10th International Workshop on Security, IWSEC 2015*, volume 9241 of *Lecture Notes in Computer Science*, pages 21–38. Springer, 2015.

[SE15b] Jae Hong Seo and Keita Emura. Revocable hierarchical identity-based encryption: History-free update, security against insiders, and short ciphertexts. In Kaisa Nyberg, editor, *Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015*, volume 9048 of *Lecture Notes in Computer Science*, pages 106–123. Springer, 2015.

[Sha07] Hovav Shacham. The BBG HIBE has limited delegation. *IACR Cryptology ePrint Archive*, 2007:201, 2007.

[SZSM17] Limin Shen, Futai Zhang, Yinxia Sun, and Jianfeng Ma. An efficient revocable ID-based encryption scheme in the standard model. *IJES*, 9(2):168–176, 2017.

[TW17a] Atsushi Takayasu and Yohei Watanabe. Lattice-based revocable identity-based encryption with bounded decryption key exposure resistance. In Josef Pieprzyk and Suriadi Suriadi, editors, *Information Security and Privacy - 22nd Australasian Conference, ACISP 2017*, volume 10342 of *Lecture Notes in Computer Science*, pages 184–204. Springer, 2017.

[TW17b] Atsushi Takayasu and Yohei Watanabe. Revocable identity-based encryption with bounded decryption key exposure resistance: Lattice-based construction and more. *IACR Cryptology ePrint Archive*, 2017:323, 2017.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009.

[WES17] Yohei Watanabe, Keita Emura, and Jae Hong Seo. New revocable IBE in prime-order groups: Adaptively secure, decryption key exposure resistant, and with short public parameters. In Helena Handschuh, editor, *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017. Proceedings*, volume 10159 of *Lecture Notes in Computer Science*, pages 432–449. Springer, 2017.

[WLJW16] Changji Wang, Yuan Li, Shengyi Jiang, and Jiayuan Wu. An efficient adaptive-id secure revocable hierarchical identity-based encryption scheme. In Meikang Qiu, editor, *Smart Computing and Communication - First International Conference, SmartCom 2016*, volume 10135 of *Lecture Notes in Computer Science*, pages 506–515. Springer, 2016.

[WLXZ14] Changji Wang, Yuan Li, Xiaonan Xia, and Kangjia Zheng. An efficient and provable secure revocable identity-based encryption scheme. *PLoS ONE*, 9(9):e106925, 2014.

[WZH$^+$19] Shixiong Wang, Juanyang Zhang, Jingnan He, Huaxiong Wang, and Chao Li. Simplified revocable hierarchical identity-based encryption from lattices. In Yi Mu, Robert H. Deng, and Xinyi Huang, editors, *Cryptology and Network Security - 18th International Conference, CANS 2019, Fuzhou, China, October 25-27, 2019, Proceedings*, volume 11829 of *Lecture Notes in Computer Science*, pages 99–119. Springer, 2019.

[XWW$^+$16] Qianqian Xing, Baosheng Wang, Xiaofeng Wang, Peixin Chen, Bo Yu, Yong Tang, and Xianming Gao. Unbounded revocable hierarchical identity-based encryption with adaptive-id security. In Jinjun Chen and Laurence T. Yang, editors, *18th IEEE International Conference on High Performance Computing and Communications; 14th*

IEEE International Conference on Smart City; 2nd IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2016, pages 430–437. IEEE, 2016.

[XWWT18] Qianqian Xing, Baosheng Wang, Xiaofeng Wang, and Jing Tao. Unbounded and revocable hierarchical identity-based encryption with adaptive security, decryption key exposure resistant, and short public parameters. *PloS one*, 13(4):e0195204, 2018.

[Zha12] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 758–775. Springer, 2012.

A Graphical Overview of Node Division

Here, we present graphical overviews of SE node division and our node division. As defined in Section 2, $\mathtt{ID}^\star$ is the target identity, and $\ell^\star$ is the minimum level of of $\mathsf{sk}_{\mathtt{ID}^\star_{[\ell^\star]}}$ received by the RHIBE adversary.

A.1 Seo-Emura Node Division

SE node division is defined as three mutually exclusive subsets, i.e., $\mathcal{SE}^{(1)}_{\ell^\star}, \mathcal{SE}^{(2)}_{\ell^\star}$, and $\mathcal{SE}^{(3)}_{\ell^\star}$.

$$\mathcal{SE}^{(1)}_{\ell^\star} := \left\{\theta : \left(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell]})} \quad \text{for } \ell \in [\ell^\star - 1]\right) \vee \left(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell^\star]})} \setminus \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell^\star]})}, \eta_{\mathtt{ID}^\star_{[\ell^\star]}})\right)\right\},$$

$$\mathcal{SE}^{(2)}_{\ell^\star} := \left\{\theta : \left(\theta \in \mathcal{BT}_{\mathtt{ID}^\star_{[\ell]}} \quad \text{for } \ell \in [\ell^\star, |\mathtt{ID}^\star|]\right) \vee \left(\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID}^\star_{[\ell^\star]})}, \eta_{\mathtt{ID}^\star_{[\ell^\star]}})\right)\right\},$$

$$\mathcal{SE}^{(3)}_{\ell^\star} := \mathcal{AN} \setminus (\mathcal{SE}^{(1)}_{\ell^\star} \cup \mathcal{SE}^{(2)}_{\ell^\star}) = \mathcal{AN} \setminus \left(\bigcup_{\ell=0}^{|\mathtt{ID}^\star|} \mathcal{BT}_{\mathtt{ID}^\star_{[\ell]}}\right).$$

Figure 2 shows the graphical overview of SE node division.

A.2 Our Node Division

Our node division is defined as two mutually exclusive subsets $\mathcal{SK}_{\ell^\star}$ and $\mathcal{KU}_{\ell^\star}$.

$$\mathcal{SK}_{\ell^\star} := \left\{\theta : \begin{array}{c} \left(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})} \quad \text{for} \quad |\mathtt{ID}| \leq \ell^\star - 1\right) \vee \\ \left(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})} \setminus \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta^\star_{\mathsf{pa}(\mathtt{ID})}) \quad \text{for} \quad |\mathtt{ID}| = \ell^\star\right) \end{array}\right\},$$

$$\mathcal{KU}_{\ell^\star} := \mathcal{AN} \setminus \mathcal{SK}_{\ell^\star} = \left\{\theta : \begin{array}{c} \left(\theta \in \mathcal{BT}_{\mathsf{pa}(\mathtt{ID})} \quad \text{for} \quad |\mathtt{ID}| \geq \ell^\star + 1\right) \vee \\ \left(\theta \in \mathsf{Path}(\mathcal{BT}_{\mathsf{pa}(\mathtt{ID})}, \eta^\star_{\mathsf{pa}(\mathtt{ID})}) \quad \text{for} \quad |\mathtt{ID}| = \ell^\star\right) \end{array}\right\}.$$

Figure 3 shows a graphical overview of our node division.

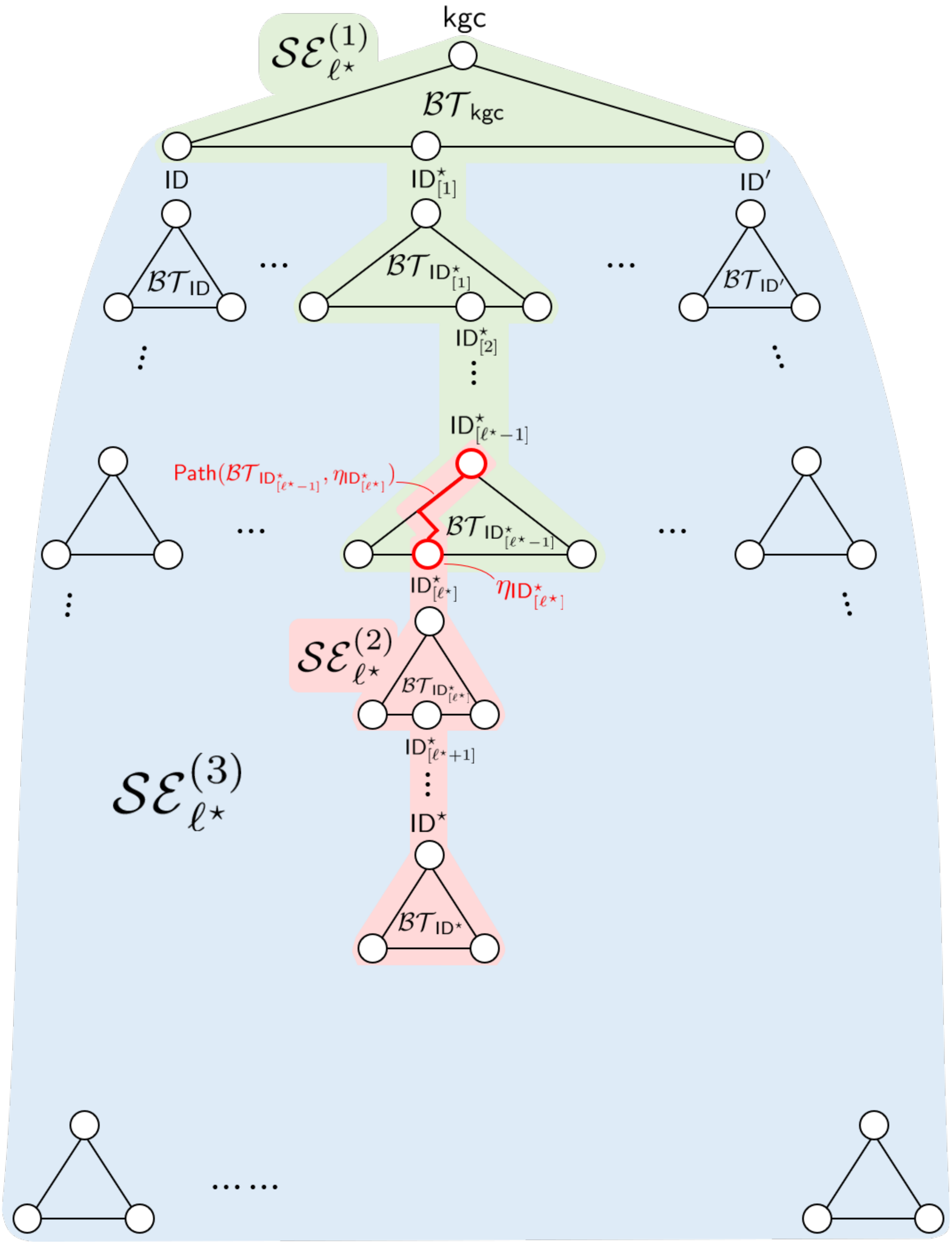
kgc
$\mathcal{SE}^{(1)}_{\ell^\star}$
$\mathcal{BT}_{\mathsf{kgc}}$
ID
$\mathsf{ID}^\star_{[1]}$
ID′
$\mathcal{BT}_{\mathsf{ID}}$
$\mathcal{BT}_{\mathsf{ID}^\star_{[1]}}$
$\mathcal{BT}_{\mathsf{ID}'}$
$\mathsf{ID}^\star_{[2]}$
$\mathsf{ID}^\star_{[\ell^\star-1]}$
$\mathsf{Path}(\mathcal{BT}_{\mathsf{ID}^\star_{[\ell^\star-1]}}, \eta_{\mathsf{ID}^\star_{[\ell^\star]}})$
$\mathcal{BT}_{\mathsf{ID}^\star_{[\ell^\star-1]}}$
$\mathsf{ID}^\star_{[\ell^\star]}$
$\eta_{\mathsf{ID}^\star_{[\ell^\star]}}$
$\mathcal{SE}^{(2)}_{\ell^\star}$
$\mathcal{BT}_{\mathsf{ID}^\star_{[\ell^\star]}}$
$\mathsf{ID}^\star_{[\ell^\star+1]}$
$\mathsf{ID}^\star$
$\mathcal{SE}^{(3)}_{\ell^\star}$
$\mathcal{BT}_{\mathsf{ID}^\star}$

Figure 2: SE node division

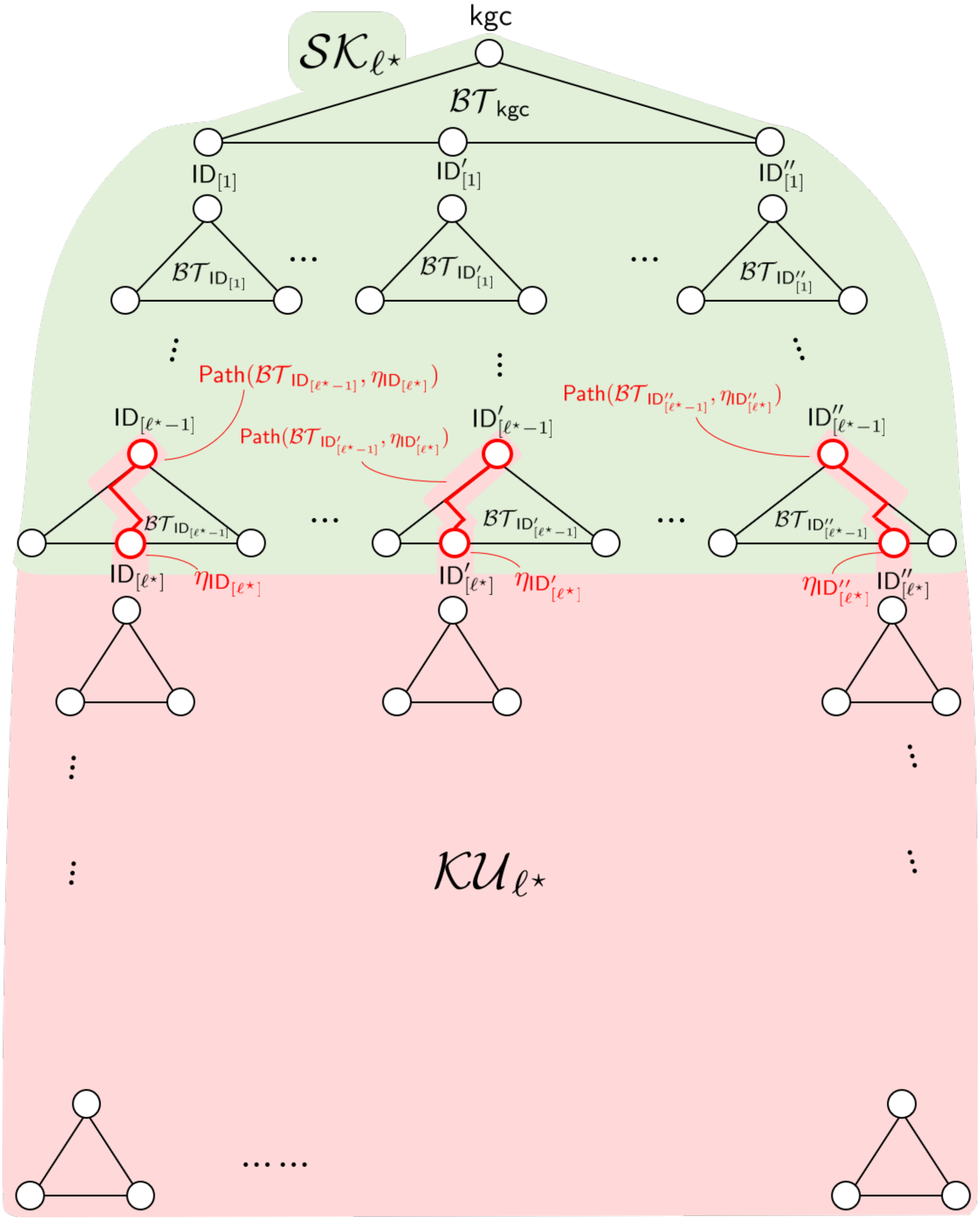
$\mathcal{SK}_{\ell^\star}$
kgc
$\mathcal{BT}_{\mathsf{kgc}}$
$\mathsf{ID}_{[1]}$
$\mathsf{ID}'_{[1]}$
$\mathsf{ID}''_{[1]}$
$\mathcal{BT}_{\mathsf{ID}_{[1]}}$
$\mathcal{BT}_{\mathsf{ID}'_{[1]}}$
$\mathcal{BT}_{\mathsf{ID}''_{[1]}}$
$\mathsf{Path}(\mathcal{BT}_{\mathsf{ID}_{[\ell^\star-1]}}, \eta_{\mathsf{ID}_{[\ell^\star]}})$
$\mathsf{Path}(\mathcal{BT}_{\mathsf{ID}'_{[\ell^\star-1]}}, \eta_{\mathsf{ID}'_{[\ell^\star]}})$
$\mathsf{Path}(\mathcal{BT}_{\mathsf{ID}''_{[\ell^\star-1]}}, \eta_{\mathsf{ID}''_{[\ell^\star]}})$
$\mathsf{ID}_{[\ell^\star-1]}$
$\mathsf{ID}'_{[\ell^\star-1]}$
$\mathsf{ID}''_{[\ell^\star-1]}$
$\mathcal{BT}_{\mathsf{ID}_{[\ell^\star-1]}}$
$\mathcal{BT}_{\mathsf{ID}'_{[\ell^\star-1]}}$
$\mathcal{BT}_{\mathsf{ID}''_{[\ell^\star-1]}}$
$\mathsf{ID}_{[\ell^\star]}$
$\eta_{\mathsf{ID}_{[\ell^\star]}}$
$\mathsf{ID}'_{[\ell^\star]}$
$\eta_{\mathsf{ID}'_{[\ell^\star]}}$
$\eta_{\mathsf{ID}''_{[\ell^\star]}}$
$\mathsf{ID}''_{[\ell^\star]}$
$\mathcal{KU}_{\ell^\star}$

Figure 3: Our node division