Efficient Identity-Based Encryption with Hierarchical Key-Insulation from HIBE

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

September 10, 2020

Abstract

Hierarchical key-insulated identity-based encryption (HKIBE) is identity-based encryption (IBE) that allows users to update their secret keys to achieve (hierarchical) key-exposure resilience, which is an important notion in practice. However, existing HKIBE constructions have limitations in efficiency: sizes of ciphertexts and secret keys depend on the hierarchical depth.

In this paper, we first triumph over the barrier by proposing simple but effective design methodologies to construct efficient HKIBE schemes. First, we show a generic construction from any hierarchical IBE (HIBE) scheme that satisfies a special requirement, called *MSK evaluatability* introduced by Emura et al. (ePrint, 2020). It provides several new and efficient instantiations since most pairing-based HIBE schemes satisfy the requirement. It is worth noting that it preserves all parameters' sizes of the underlying HIBE scheme, and hence we obtain several efficient HKIBE schemes under the k-linear assumption in the standard model. Since MSK evaluatability is dedicated to pairing-based HIBE schemes, the first construction restricts pairing-based instantiations. To realize efficient instantiation from various assumptions, we next propose a generic construction of an HKIBE scheme from any *plain* HIBE scheme. It is based on Hanaoka et al.'s HKIBE scheme (Asiacrypt 2005), and does not need any special properties. Therefore, we obtain new efficient instantiations from various assumptions other than pairing-oriented ones. Though the sizes of secret keys and ciphertexts are larger than those of the first construction, it is more efficient than Hanaoka et al.'s scheme in the sense of the sizes of master public/secret keys.

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

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

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

Contents

1 Introduction

1.1 Background

Identity-based encryption (IBE) [BF01] allows us to use arbitrary strings (e.g., user names, e-mail addresses) as users' public keys. After earlier seminal works [BB04, Wat05], considerable research related to IBE has been conducted from various perspectives such as efficiency improvements [JR13, Lew12, Wat09], weakening assumptions [DG17b], post-quantum constructions [ABB10a, ABB10b, BLSV18, CHKP12], and additional security properties [BGK08, BLSV18, BW06, BWY11, CDRW10, HHSI05]. Similar results have been obtained in the context of *hierarchical IBE* (HIBE)[GS02, HL02], which is one of the important extensions of IBE; e.g., efficiency improvements [CW14, GCTC16, Lew12, LP19, LP20, LW10, LW11, Wat09], weakening assumptions [DG17a], post-quantum constructions [ABB10a, ABB10b, CHKP12], and additional security properties [BW06, LRW11, SE13a].

According to Cisco's report [Cis14], tens of billions of IoT devices are expected to be deployed over the next few years. Therefore, one of the key challenges is how to make communications over IoT devices fast and reliable. Recently, IBE is expected to be used in the IoT environments (e.g., [AKA+19, KHA+19]) since devices' identities (serial numbers, MAC addresses, etc.) can be set as their public keys.[1] Therefore, IoT devices can make reliable and fast communication without PKI (i.e., without verifying public-key certificates). Another practical security requirement for robust IoT systems is *key-exposure resilience*. Secure IoT systems using IBE should still be available and guarantee a certain security level even if some devices in the system are corrupted, and their secret keys are exposed. Particularly in the IoT setting, it is difficult to manually revoke and re-setup corrupted IoT devices since it seems hard to detect when and which devices leak their secret keys. Therefore, the key-exposure resilience is important in practice; it guarantees that even if some devices (partially) leak their secret keys, the devices are still available in some sense. Thus, we focus on the problem is *how to achieve the key-exposure resilience (as efficient as possible) in the IBE setting*.

One of promising approaches to address the above problem is the *key-updating approach*. This paper considers the following *key-insulation mechanism* [DKXY02, HHSI05]. We prepare two kinds of secret keys depending on their roles: *helper keys*, stored on physically-secure devices, and *decryption keys*, which are stored on weak devices that may be tampered. Ciphertexts can be decrypted by decryption keys, which are periodically and non-interactively updated by helper keys. This approach is suitable for the above IoT scenario (and, of course, the more standard usage scenario) since (a) decryption keys are updated in a non-interactive way, and (b) decryption keys can be renewed and continue to be used regardless of whether the system owner knows which decryption keys are leaked. IBE with the key-insulation mechanism is called *key-insulated IBE* (KIBE) [HHSI05], and the security which should be achieved in this approach is:

(1) even if many decryption keys are exposed, KIBE can guarantee the security of non-exposed decryption keys;

(2) even if the helper key is exposed, no information on any decryption keys is leaked as long as no decryption keys are exposed.

The key-insulation structure can be extended to a hierarchical one, and IBE with the hierarchical

[1] Attribute-based encryption (ABE) [SW05, GPSW06] provides more flexible access control than IBE and its variants, such as wildcarded IBE [ABC+11] and wicked IBE [AKN07], though it is much less efficient. The IBE variants are flexible enough to apply for various IoT environments [AKA+19, KHA+19].

key-insulated property is called *hierarchical KIBE* (HKIBE) [HHSI05].[2] In HKIBE, helper keys are separated into multiple levels. Helper keys can update lower-level helper keys, and the lowest-level helper keys update decryption keys. Thus, the impact of key leakage can be significantly reduced by storing helper keys at different levels in different devices.

Although HKIBE seems to provide practical applications as above, an efficiency issue in HKIBE constructions remains unsolved. Hanaoka et al. [HHSI05] showed a generic construction from any HIBE scheme. It can be instantiated from various assumptions, however essentially sacrifices sizes of ciphertexts and decryption keys; it requires at least $O(L)$ HIBE ciphertexts and $O(L)$ HIBE secret keys for the resulting ciphertexts and decryption keys, respectively, where L is the maximum depth of hierarchical key-insulation. Therefore, even if the underlying HIBE scheme achieves compact ciphertexts and/or secret keys, those of the resultant HKIBE scheme cannot be compact. Although Hanaoka et al. [HHSI05] also showed a concrete HKIBE scheme from computational bilinear Diffie-Hellman (CBDH) assumption, which is more efficient than the generic construction, it relies on the random oracle and do not have compact parameters, in the sense that sizes of ciphertexts and decryption keys are not constant. The work of [SW18, WS16] proposed adaptively secure HKIBE schemes with compact ciphertexts and decryption keys from pairings; however, unfortunately, we found a flaw in the security proofs (which we communicated to the authors).[3] Thus, there are no secure HKIBE constructions that achieve compact ciphertexts and decryption keys.

1.2 Our Contributions

In this paper, we successfully make significant progress in constructing efficient HKIBE schemes. Specifically, we show two generic constructions of HKIBE schemes.

Generic Construction from HIBE with MSK Evaluatability. We take note of the similarities in security games in HKIBE and revocable HIBE (RHIBE) [BGK08, SE13a, SE15]; unlike standard (H)IBE, an adversary is allowed to get (a part of) a secret key of a challenge identity in both games. Based on the observation, we take a similar approach to the recent RHIBE construction [ETW20], and propose our first construction from any HIBE scheme that satisfies *MSK evaluatability*, which is the special algebraic property introduced in [ETW20]. Although the property restricts an applicable class of HIBE schemes to our construction, most pairing-based HIBE schemes, including most-efficient-ever ones [CG17, CW14, GCTC16], meet it. Our generic construction provides several concrete HKIBE schemes with new features as follows.

- The first HKIBE schemes with compact ciphertexts and decryption keys from [CG17, CW14] under the standard k-linear assumption. Note that there are no known schemes with similar efficiency even when we ignore the adaptive security, standard assumptions, and the standard model.[4]
- The first HKIBE scheme with compact master public keys in the standard model from [GCTC16] under the k-linear assumption.

Generic Construction from Any HIBE. Our second construction aims to get rid of the special property required in our first construction, and is a generic construction from any *plain* HIBE schemes. While this construction is based on [HHSI05], it achieves compact master keys[5] and does

[2] One may think up HIBE with the hierarchical key-insulated property. In this paper, we do not consider such an HIBE scheme since it must be quite complicated, and there has been actually no such work.

[3] We give the overview of the flaw in Appendix A.

[4] To be precise, an instantiation from [RS14], which is a special case of [CG17], is the same as Shikata and Watanabe's scheme [SW18]. It means that their scheme turns out to be secure, and we successfully fix the bug in their security proof.

[5] We refer to a pair of a master public and master secret keys as *master keys* for simplicity.

Table 1: A comparison between Hanaoka et al.'s generic construction and ours. "Generic HHSI05" means the generic construction shown in [HHSI05]. Each parameter of all HKIBE schemes consists of the same ingredient: a master public key pp, master secret key mk, and ciphertext $\mathsf{ct}_{\mathtt{id},\mathtt{t}}$ consist of master public keys, master secret keys, and ciphertexts of the underlying HIBE scheme, respectively, and a level-ℓ helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ and decryption key $\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})}$ consist of HIBE secret keys. Therefore, we compare the number of the ingredients that constitute each parameter. ROM and Std. stand for the random oracle model and the standard model, respectively, and L and ℓ denote the maximum hierarchical size and a hierarchical level, respectively. let α be the ciphertext overhead, which mainly includes an one-time signature and its verification key, caused by the multiple-encryption technique [DK05].

Construction	$\lvert\mathsf{pp}\rvert$	$\lvert\mathsf{mk}\rvert$	$\lvert\mathsf{ct}_{\mathtt{id},\mathtt{t}}\rvert$	$\lvert\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}\rvert$	$\lvert\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})}\rvert$	Security	Model	Building Block	Reduction Loss
Generic HHSI05 [HHSI05]	$O(L)$	$O(L)$	$O(L)$	$O(L-\ell)$	$O(L)$	CCA	ROM	CPA-secure HIBE	$O(Q)$
	$O(L)$	$O(L)$	$O(L)+\alpha$	$O(L-\ell)$	$O(L)$	CCA	Std.	CPA-secure HIBE and OTS	$O(Q)$
First Construction (§ 4)	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	CPA	Std.	CPA-secure HIBE w/ MSK eval.	$O(QL)$
	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	CCA	Std.	CCA-secure HIBE w/ MSK eval.	$O(QL)$
Second Construction (§ 5)	$O(1)$	$O(1)$	$O(L)$	$O(L-\ell)$	$O(L)$	CPA	Std.	CPA-secure HIBE	$O(L)$
	$O(1)$	$O(1)$	$O(L)+\alpha$	$O(L-\ell)$	$O(L)$	CCA	Std.	CPA-secure HIBE and OTS	$O(L)$

not require random oracles. We get the following concrete HKIBE schemes with new features from the second construction.

- The first (almost) tightly and adaptively secure HKIBE scheme with compact master keys from the k-linear assumption in the standard model from [LP19, LP20].
- The first selectively secure HKIBE scheme with compact master keys from the various assumptions in the standard model: the learning with errors [ABB10a, CHKP12]; learning from parity with noise [BLSV18]; computational Diffie-Hellman without pairing; and factoring Blum integers [DG17b].

Achieving CCA Security. Although we basically consider CPA-secure HKIBE schemes, we can easily extend them to CCA-secure schemes as follows. The first construction can be lifted to a CCA-secure scheme by just replacing the underlying CPA-secure HIBE scheme with a CCA-secure one. Note that since there is a well-known transformation [BCHK07] from CPA-secure HIBE schemes to CCA-secure ones that preserve almost the same efficiency, the CCA-secure version of our first construction achieves similar efficiency to the CPA-secure construction. We obtain a CCA-secure version of our second construction by applying the multiple-encryption technique [DK05], which is a well-known technique to achieve CCA security without random oracles. Note that as observed in the HHSI05 paper [HHSI05], it is also applicable to their scheme.

Efficiency Comparison. We compare our constructions with previous schemes. Table 1 provides efficiency comparisons between Hanaoka et al.'s generic construction [HHSI05] and our constructions. Our first construction preserves all parameter sizes of the underlying HIBE scheme. Our second construction has similar efficiency to the HHSI05 scheme but achieves constant-size master keys. Table 2 shows concrete efficiency among existing schemes and instantiations of our first

Table 2: A comparison among previous CCA-secure instantiations and the CCA-secure version of our first construction. "Concrete HHSI05" means the direct construction shown in [HHSI05]. We compare the number of group elements that constitute each parameter in this table. Note that we do not instantiate the underlying OTS scheme in all instantiations except for Concrete HHSI05, and the ciphertext overhead (i.e., the OTS elements) is denoted by α as in Table 1.

Scheme	$\lvert\mathsf{pp}\rvert$	$\lvert\mathsf{mk}\rvert$	$\lvert\mathsf{ct}_{\mathtt{id},\mathtt{t}}\rvert$	$\lvert\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}\rvert$	$\lvert\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})}\rvert$	Assumption
Concrete HHSI05 [HHSI05] (in ROM)	$O(1)$	$O(L)$	$O(L)$	$O((L-\ell)^2)$	$O(L^2)$	CBDH
Generic HHSI05 [HHSI05] +[CG17, CW14] w/ OTS	$O(L^2)$	$O(L)$	$O(L)+\alpha$	$O(\ell(L-\ell))$	$O(L)$	SXDH & OTS
Generic HHSI05 [HHSI05] +[GCTC16] w/ OTS	$O(L)$	$O(1)$	$O(L^2)+\alpha$	$O((L-\ell)^2)$	$O(L^2)$	SXDH & OTS
SW18 [SW18] w/ OTS (flawed)	$O(L)$	$O(1)$	$O(1)+\alpha$	$O(\ell)$	$O(1)$	SXDH & OTS
First Scheme (§ 4) +[CG17, CW14] w/ OTS	$O(L)$	$O(1)$	$O(1)+\alpha$	$O(\ell)$	$O(1)$	SXDH & OTS
First Scheme (§ 4) +[GCTC16] w/ OTS	$O(1)$	$O(1)$	$O(L)+\alpha$	$O(L-\ell)$	$O(L)$	SXDH & OTS

construction, which is more efficient than our second construction. The state-of-the-art pairing-based HIBE schemes [CG17, CW14, GCTC16] provide efficient HKIBE schemes. In particular, the instantiation of the first construction from [CG17, CW14] is CPA-secure under the k-linear assumption and achieves the same efficiency as the SW18 scheme [SW18] when setting $k = 1$, i.e., the symmetric external Diffie-Hellman (SXDH) assumption. We again would like to emphasize that the security proof in [SW18] was flawed. Furthermore, the first scheme can be easily extended to CCA-security by replacing the underlying CPA-secure HIBE scheme with CCA-secure one. Note that, as we noted above, we know the transformation [BCHK07] for HIBE that lifts CPA security to CCA security without sacrificing efficiency.

1.3 Related Work

The notion of key-insulated cryptography was first introduced by Dodis et al. [DKXY02]. Specifically, they formalized two kinds of key-insulated security notions: the one is weak security, which only satisfies the condition (1) described earlier; the other is strong security, which satisfies both (1) and (2). Bellare and Palacio [BP06] showed that weakly secure key-insulated public-key encryption is equivalent to (a restricted form of) IBE. Thus far, the key-insulated security have been considered in the IBE setting (with additional properties) [WLCM06, WLC$^+$08]. The key-insulation structure was extended to the hierarchical one by Hanaoka et al. [HHSI05], where the security captures the strong security, and they proposed an adaptively secure HKIBE scheme both with and without random oracles. Watanabe and Shikata [WS16] proposed an adaptively secure HKIBE scheme with compact ciphertexts and decryption keys. Later, the same authors [SW18] found out a bug in the security proof in [WS16] and fixed it and the corresponding construction. However, it contains another bug in their security proof, and our proposal fixes it as mentioned earlier.

Another key-updating approach is *forward security* [CHK07], which guarantees that even if the secret key is leaked, no information of previously-encrypted plaintexts is leaked by updating the secret key by themselves. However, it is inapplicable to the IoT scenario since it only prevents the leakage of data previously encrypted before the key leakage, and the exposed secret keys will not be able to be used.

R(H)IBE [BGK08, SE13a] is (H)IBE with efficient revocation functionality, and has a similar key-updating procedure and security notion to HKIBE. Each user needs to periodically update their decryption key, and the update is successful unless the user is revoked. In the security game, an adversary is allowed to get some decryption keys associated with a challenge identity. A lot of constructions have been proposed in the context of RIBE [BGK08, GW19, ISW17, Lee19, LLP17, ML19, SE13b, WES17] and RHIBE [ESY16, ETW20, KMT19, LP18, RLPL15, SE13a, SE15, WZH+19] thus far.

Organization. In Section 2, we briefly review hierarchical time-period map functions, which make us consistently deal with several layers of time periods in HKIBE, and HIBE with MSK evaluatability. We give the definition of HKIBE in Section 3, and show our two generic constructions in Sections 4 and 5, respectively.

2 Preliminaries

2.1 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 $[a, b] := \{a, a+1, \dots, b\}$ and $[a] := [1, a]$. As a special case, $[a, b] = \emptyset$ for $a > b$. For a finite set S, let $x \leftarrow_R S$ denote sampling x from S uniformly at random. For a κ_1-bit binary string $\mathtt{id}_1 \in \{0,1\}^{\kappa_1}$ and a κ_2-bit binary string $\mathtt{id}_2 \in \{0,1\}^{\kappa_2}$, let $\mathtt{id}_1 \| \mathtt{id}_2 \in \{0,1\}^{\kappa_1+\kappa_2}$ denote a $(\kappa_1 + \kappa_2)$-bit concatenation of $\mathtt{id}_1$ and $\mathtt{id}_2$.

2.2 Hierarchical Time-Period Map Functions

To properly deal with key-updating functionality, we consider (discrete) *time periods*, which are time spans during which a specific secret key is authorized for cryptographic operations such as decryption or in which the secret keys may remain in effect. Let $\mathcal{T}$ be a set of time periods. It is natural to consider that such a time period for key updates is related to *actual time*, i.e., clock time that we usually use in our daily lives. For instance, we can set a set of time periods $\mathcal{T}$ as days, say, $\mathcal{T} := \{\mathsf{2020\_Sep\_1}, \mathsf{2020\_Sep\_2}, \dots\}$. To connect time periods and actual time, we consider *time-period map functions* [HHSI05]. A time-period map function $\mathsf{T} : \mathcal{T}_{act} \to \mathcal{T}$ maps actual times to time periods, where $\mathcal{T}_{act}$ is a (possibly countably infinite) set of actual times.

Time-period map functions can be extended so that they have a certain hierarchical structure. Let $L := \mathsf{poly}(\lambda)$, and $\mathcal{T}_\ell$ for $\ell \in [0, L]$ be a finite set of time periods. We assume $|\mathcal{T}_L| \leq \cdots \leq |\mathcal{T}_1| \leq |\mathcal{T}_0|$ and $|\mathcal{T}_L| = 1$ (i.e., $\mathsf{T}_L(\mathtt{t}) = 0$ for any $\mathtt{t}$) for simplicity. The reason why we consider several layers of time periods is that in HKIBE, we consider several secret keys, called helper keys for $\mathcal{T}_L, \dots, \mathcal{T}_1$ and decryption keys for $\mathcal{T}_0$. More specifically, we consider different time intervals for the helper and decryption keys; the helper key at the highest level (i.e., $\mathcal{T}_L$) is never updated, and other helper keys are more frequently updated as the level decreases. The decryption key, which is related to $\mathcal{T}_0$, is most often updated. The hierarchical version of time-period map functions for the depth L captures this situation, and can be defined as a set of L time-period map functions $\mathsf{T}_L, \dots, \mathsf{T}_1, \mathsf{T}_0$ for distinct time-period sets $\mathcal{T}_L, \dots, \mathcal{T}_1, \mathcal{T}_0$. We use the hierarchical time-period map functions to mange several time periods consistently; one actual time $\mathtt{t} \in \mathcal{T}_{act}$ produces an $(L+1)$-dimensional *time-period vector* $(t_L, \dots, t_1, t_0) \in \mathcal{T}_L \times \cdots \times \mathcal{T}_1 \times \mathcal{T}_0$ via the functions $\mathsf{T}_L, \dots, \mathsf{T}_1, \mathsf{T}_0$. Let us give an example for readers: for $L = 3$ and $\mathtt{t} = \mathtt{2020\_Sep\_10\_23:59}$, we have $\mathsf{T}_3(\mathtt{t}) = \mathtt{0}$, $\mathsf{T}_2(\mathtt{t}) = \mathtt{2020}$, $\mathsf{T}_1(\mathtt{t}) = \mathtt{2020\_Sep}$, and $\mathsf{T}_0(\mathtt{t}) = \mathtt{2020\_Sep\_10}$. T_3 in this example indicates "no update", and T_2, T_1, and T_0 capture yearly, monthly, and daily updates, respectively. For notational convenience, we use a shortened form of time-period vectors for $\mathtt{t} \in \mathcal{T}_{act}$: $\mathsf{T}_{[L-1,\ell]}(\mathtt{t}) := (\mathsf{T}_{L-1}(\mathtt{t}), \ldots, \mathsf{T}_{\ell}(\mathtt{t}))$, where $\ell \in [0, L-1]$.[6] Note that the order of $[\cdot]$ of $\mathsf{T}_{[\cdot]}(\cdot)$ is reversed compared with the order of $[\cdot]$ defined in Section 2.1.

2.3 HIBE

Hierarchical Identity. Let an ℓ-dimensional identity vector $\mathtt{ID}_\ell := (\mathtt{id}_1, \ldots, \mathtt{id}_\ell)$ denote an identity at a level (or, a hierarchy depth) ℓ. In this paper, we may sometimes call $\mathtt{ID}_\ell = (\mathtt{id}_1, \cdots, \mathtt{id}_\ell)$ and each $\mathtt{id}_i$ a *hierarchical identity* and an *element identity*, respectively. Let $\mathcal{I}$ be an element-identity space which is determined only by the security parameter λ, and therefore, a hierarchical-identity space at level ℓ is $\mathcal{I}^\ell$.

We define several notations for $\mathtt{ID}_\ell = (\mathtt{id}_1, \cdots, \mathtt{id}_\ell)$ below. For a non-negative integer $k \leq \ell$, an k-dimensional prefix of $\mathtt{ID}_\ell$ is denoted by $\mathtt{ID}_{[k]} := (\mathtt{id}_1, \ldots, \mathtt{id}_k)$. We denote by $\mathsf{prefix}^+(\mathtt{ID}_\ell) := \{\mathtt{ID}_{[1]}, \mathtt{ID}_{[2]}, \ldots, \mathtt{ID}_{[\ell-1]}, \mathtt{ID}_\ell\}$ a set of all prefixes of $\mathtt{ID}_\ell$ and itself. We often omit the subscript from $\mathtt{ID}_\ell$ and simply describe $\mathtt{ID}$ for simplicity, and use $|\mathtt{ID}| := \ell$ to denote a hierarchical level of the hierarchical identity.

Syntax. An HIBE scheme Σ with the depth L consists of four algorithms $(\mathsf{Init}, \mathsf{Enc}, \mathsf{GenSK}, \mathsf{Dec})$.

- $\mathsf{Init}(1^\lambda, L) \to (\mathsf{MPK}, \mathsf{MSK})$: given the security parameter λ and the maximum hierarchical depth L, it outputs a master-key pair $(\mathsf{MPK}, \mathsf{MSK})$.
- $\mathsf{Enc}(\mathsf{MPK}, \mathtt{ID}, \mathsf{M}) \to \mathsf{C}_{\mathtt{ID}}$: given MPK, user's identity $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$, and a plaintext M, it outputs a ciphertext $\mathsf{C}_{\mathtt{ID}}$.
- $\mathsf{GenSK}(\mathsf{MPK}, \mathsf{SK}_{\mathtt{ID}'}, \mathtt{ID}) \to \mathsf{SK}_{\mathtt{ID}}$: given MPK, a user's secret key $\mathsf{SK}_{\mathtt{ID}'}$, and an identity $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$ s.t. $\mathtt{ID}$'s parent is $\mathtt{ID}'$, it outputs a secret key $\mathsf{SK}_{\mathtt{ID}}$. The second input $\mathsf{SK}_{\mathtt{ID}'}$ can be replaced by MSK. For notational convenience, we regard $\mathsf{SK}_{\mathtt{ID}_0}$ as the master secret key (MSK) MSK.
- $\mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{\mathtt{ID}}, \mathsf{C}_{\mathtt{ID}}) \to \mathsf{M}$: given MPK, a secret key $\mathsf{SK}_{\mathtt{ID}}$, and a ciphertext $\mathsf{C}_{\mathtt{ID}}$, it outputs the decryption result M.

Correctness. We require that for all security parameters $\lambda \in \mathbb{N}$, hierarchy levels $L \in \mathbb{N}$, $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Init}(1^\lambda, L)$, identities $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$, and plaintexts M, it holds $\mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{\mathtt{ID}}, \mathsf{Enc}(\mathsf{MPK}, \mathtt{ID}, \mathsf{M})) = \mathsf{M}$ with overwhelming probability, where $\mathsf{SK}_{\mathtt{ID}} \leftarrow \mathsf{GenSK}(\mathsf{MPK}, \mathsf{MSK}, \mathtt{ID})$. Moreover, given $\mathsf{SK}_{\mathtt{ID}}$ for any identity $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$, $\mathsf{GenSK}(\mathsf{MPK}, \mathsf{MSK}, \mathtt{ID})$ and $\mathsf{GenSK}(\mathsf{MPK}, \mathsf{SK}_{\mathtt{ID}'}, \mathtt{ID})$ s.t. $\mathtt{ID}' \in \mathsf{prefix}^+(\mathtt{ID})$ are identically distributed.

Adaptive Security. Intuitively, HIBE requires that it is hard for an adversary who adaptively obtains polynomially many secret keys $\mathsf{SK}_{\mathtt{ID}}$ such that $\mathtt{ID} \notin \mathsf{prefix}^+(\mathtt{ID}^\star)$ to extract secret information from $\mathsf{C}_{\mathtt{ID}^\star}$.

More formally, let Σ be an HIBE scheme, and we consider a game between an adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. The game is parameterized by the security parameter λ and the maximum hierarchical depth L. The game proceeds as follows: $\mathcal{C}$ first runs $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Init}(1^\lambda, L)$ and gives MPK to $\mathcal{A}$. $\mathcal{A}$ may adaptively make the following *secret-key reveal query*: upon a query $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$ from $\mathcal{A}$, $\mathcal{C}$ returns $\mathsf{SK}_{\mathtt{ID}} \leftarrow \mathsf{GenSK}(\mathsf{MPK}, \mathsf{MSK}, \mathtt{ID})$ to $\mathcal{A}$. $\mathcal{A}$ is also allowed to make the following *challenge query* only once: upon a query $(\mathtt{ID}^\star, \mathsf{M}_0^\star, \mathsf{M}_1^\star)$ from $\mathcal{A}$ such that $|\mathsf{M}_0^\star| = |\mathsf{M}_1^\star|$, $\mathcal{C}$

[6] We here omit $t_L \in \mathcal{T}_L$ for simplicity since $|\mathcal{T}_L| = 1$.

returns $\mathsf{C}^\star_{\mathtt{ID}^\star} \leftarrow \mathsf{Enc}(\mathsf{MPK}, \mathtt{ID}^\star, \mathsf{M}^\star_b)$ to $\mathcal{A}$, where $b \leftarrow_R \{0,1\}$. Note that $\mathcal{A}$ is not allowed to make the secret-key reveal query on $\mathtt{ID}^\star$ and its prefix in this game. At some point, $\mathcal{A}$ outputs $b' \in \{0,1\}$ as its guess for b and terminates. In this game, $\mathcal{A}$'s adaptive security advantage is defined by $\mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L,\mathcal{A}}(\lambda) \coloneqq 2 \cdot |\Pr[b' = b] - 1/2|$.

Definition 1 (CPA security for HIBE)**.** *We say that an HIBE scheme Σ with depth L satisfies* adaptive-identity CPA security *(or* adaptive security *for brevity), if the advantage* $\mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L,\mathcal{A}}(\lambda)$ *is negligible for all PPT adversaries $\mathcal{A}$.*

The selective-identity CPA security (selective security for short) is analogously defined except that the challenge identity $\mathtt{ID}^\star$ is submitted to $\mathcal{C}$ at the beginning of the game, instead of the challenge query. Furthermore, CCA security is also defined by allowing $\mathcal{A}$ to submit the following decryption query: upon a query $(\mathtt{ID}, \mathsf{C}_{\mathtt{ID}})$ $(\neq (\mathtt{ID}^\star, \mathsf{C}^\star_{\mathtt{ID}^\star}))$ from $\mathcal{A}$, $\mathcal{C}$ returns $\mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{\mathtt{ID}}, \mathsf{C}_{\mathtt{ID}})$ to $\mathcal{A}$.

MSK Evaluatability [ETW20]. We require that an HIBE scheme used in our first construction satisfies the *MSK evaluatability*, which is a special algebraic property introduced in [ETW20]. In the following, we use a notation $\mathsf{SK}_{\mathtt{ID}}[\mathsf{MSK}]$, instead of $\mathsf{SK}_{\mathtt{ID}}$, to explicitly describe the MSK-part of $\mathsf{SK}_{\mathtt{ID}}$, i.e., which element of $\mathcal{MSK}$ is used to compute $\mathsf{SK}_{\mathtt{ID}}$.

Intuitively, MSK evaluatability has the following two properties.

(1) Anyone can sample a random element $\widehat{\mathsf{MSK}} \in \mathcal{MSK}$, called a *pseudo-MSK*, where $\mathcal{MSK}$ is a space of possible master secret keys. We describe the sampling procedure as a pseudo-MSK sampling algorithm $\mathsf{SampMSK}$. Furthermore, anyone create secret keys $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}]$ for any $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$ under a pseudo-MSK $\widehat{\mathsf{MSK}}$. This pseudo-MSK $\widehat{\mathsf{MSK}}$ is, of course, different from the true MSK MSK with overwhelming probability.[7]

(2) Suppose that $\mathcal{MSK}$ has some algebraic structure and allows one to compute $\widehat{\mathsf{MSK}}_1 \cdot \widehat{\mathsf{MSK}}_2$ and $\widehat{\mathsf{MSK}}_1/\widehat{\mathsf{MSK}}_2$ for any $\widehat{\mathsf{MSK}}_1, \widehat{\mathsf{MSK}}_2 \in \mathcal{MSK}$. Note that $\widehat{\mathsf{MSK}}_1$ and $\widehat{\mathsf{MSK}}_2$ might be the true MSK. Let $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_1]$ and $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_2]$ be HIBE secret keys for the same identity $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$ but under $\widehat{\mathsf{MSK}}_1$ and $\widehat{\mathsf{MSK}}_2$, respectively. Then, there exists an efficient algorithm $\mathsf{EvalMSK}$ which merges the two secret keys into one secret key $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_1 \cdot \widehat{\mathsf{MSK}}_2]$ (resp., $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_1/\widehat{\mathsf{MSK}}_2]$) with a label $\mathtt{mul}$ (resp., $\mathtt{div}$).

Formally, MSK evaluatability is defined as follows.

Definition 2 (MSK Evaluatability [ETW20])**.** *Let Σ be an HIBE scheme. We say that Σ supports MSK evaluatability if there exist algorithms* $\mathsf{SampMSK}$ *and* $\mathsf{EvalMSK}$*:*

- $\mathsf{SampMSK}(\mathsf{MPK}) \to \widehat{\mathsf{MSK}}$*: This is the* pseudo-MSK sampling *algorithm that, given* MPK*, outputs a pseudo-MSK* $\widehat{\mathsf{MSK}} \in \mathcal{MSK}$*.*
- $\mathsf{EvalMSK}(\mathsf{MPK}, \mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_1], \mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_2], \mathsf{lab}) \to \mathsf{SK}_{\mathtt{ID}}[f_{\mathsf{lab}}(\widehat{\mathsf{MSK}}_1, \widehat{\mathsf{MSK}}_2)]$*: This is the* MSK evaluation *algorithm that, given two secret keys* $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_1]$, $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_2]$ *for the same* $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$ *under* $\widehat{\mathsf{MSK}}_1, \widehat{\mathsf{MSK}}_2 \in \mathcal{MSK}$*, and a label* $\mathsf{lab} \in \{\mathtt{mul}, \mathtt{div}\}$*, it outputs a secret key* $\mathsf{SK}_{\mathtt{ID}}[f_{\mathsf{lab}}(\widehat{\mathsf{MSK}}_1, \widehat{\mathsf{MSK}}_2)]$*, where* $f_{\mathtt{mul}}(\widehat{\mathsf{MSK}}_1, \widehat{\mathsf{MSK}}_2) = \widehat{\mathsf{MSK}}_1 \cdot \widehat{\mathsf{MSK}}_2$ *and* $f_{\mathtt{div}}(\widehat{\mathsf{MSK}}_1, \widehat{\mathsf{MSK}}_2) = \widehat{\mathsf{MSK}}_1/\widehat{\mathsf{MSK}}_2$*.*

Moreover, the following two requirements are satisfied:

[7] Otherwise, MSK evaluatability immediately breaks the security of HIBE.

- ▷ **Pseudo-MSK Indistinguishability:** *For any lab* $\in \{\mathtt{mul}, \mathtt{div}\}$ *and any* $\widehat{\mathsf{MSK}} \in \mathcal{MSK}$*, given* MPK *and* $\widehat{\mathsf{MSK}}$*, the two distributions* $\mathsf{SampMSK}(\mathsf{MPK})$ *and* $f_{\mathsf{lab}}(\widehat{\mathsf{MSK}}, \mathsf{SampMSK}(\mathsf{MPK}))$ *are identically distributed.*
- ▷ **Evaluation Correctness:** *For any* $\mathsf{lab} \in \{\mathtt{mul}, \mathtt{div}\}$*, any* $\widehat{\mathsf{MSK}}_1, \widehat{\mathsf{MSK}}_2 \in \mathcal{MSK}$*, and any* $\mathtt{ID} \in \mathcal{I}^{|\mathtt{ID}|}$*, given* MPK *and* $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_1], \mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_2]$*, the two distributions* $\mathsf{GenSK}(\mathsf{MPK}, f_{\mathsf{lab}}(\widehat{\mathsf{MSK}}_1, \widehat{\mathsf{MSK}}_2), \mathtt{ID})$ *and* $\mathsf{EvalMSK}(\mathsf{MPK}, \mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_1], \mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_2], \mathsf{lab})$ *are identically distributed.*

Note that most pairing-based HIBE schemes can satisfy MSK evaluatability. For example, as noted in [ETW20], several state-of-the-art pairing-based HIBE schemes [CG17, CW14, GCTC16] has this property. Let us give an intuition with the following abstract example. Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be cyclic groups (group operations in all are written in multiplicative forms) of prime-order p, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a non-degenerate bilinear map. We use the implicit notation [EHK+17]: for $a \in \mathbb{Z}_p$ and generators $g_i \in \mathbb{G}_i$ ($i \in \{1, 2, T\}$), $[a]_i := g_i^a \in \mathbb{G}_i$, and for a vector $\mathbf{a} := (a_1, \ldots, a_d) \in \mathbb{Z}_p^k$, $[\mathbf{a}]_i := ([a_1]_i, \ldots, [a_d]_i) \in \mathbb{G}_i$. In several pairing-based HIBE schemes based on the k-linear assumption (e.g., [CG17, CW14]), the MSK is in the form of $[\mathbf{k}]_2 \in \mathbb{G}_2^{k+1}$ and the secret key $\mathsf{SK}_{\mathtt{ID}}[\mathsf{MSK}]$ contains $[\mathbf{k}]_2 \cdot \mathsf{F}(\mathtt{ID})^r$, where $\mathsf{F} : \mathcal{I} \to \mathbb{G}_2^{k+1}$ is a certain public function and $r \in \mathbb{Z}_p$ is a randomness. It is obvious that since anyone can compute a pseudo-MSK $\widehat{\mathsf{MSK}} := [\widehat{\mathbf{k}}]_2$ for uniformly sampled $\widehat{\mathbf{k}} \in \mathbb{Z}_p^{k+1}$, there exists the $\mathsf{SampMSK}$ algorithm. Moreover, it clearly satisfies pseudo-MSK indistinguishability since even given $[\mathbf{k}]_2$, $[\mathbf{k}]_2 \cdot [\widehat{\mathbf{k}}]_2 = [\mathbf{k} + \widehat{\mathbf{k}}]_2$ (or $[\mathbf{k}]_2 / [\widehat{\mathbf{k}}]_2 = [\mathbf{k} - \widehat{\mathbf{k}}]_2$) and $[\widehat{\mathbf{k}}]_2$ are identically distributed. Furthermore, it is easy to confirm that it also provides $\mathsf{EvalMSK}$: for any $\widehat{\mathsf{MSK}}_1 := [\widehat{\mathbf{k}}_1]_2, \widehat{\mathsf{MSK}}_2 := [\widehat{\mathbf{k}}_2]_2 \in \mathbb{G}_2^{k+1}$, the corresponding component of $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_1 \cdot \widehat{\mathsf{MSK}}_2]$ can be computed as $([\widehat{\mathbf{k}}_1]_2 \cdot \mathsf{F}(\mathtt{ID})^{r_1}) \cdot ([\widehat{\mathbf{k}}_2]_2 \cdot \mathsf{F}(\mathtt{ID})^{r_1}) = [\widehat{\mathbf{k}}_1 + \widehat{\mathbf{k}}_2]_2 \cdot \mathsf{F}(\mathtt{ID})^{r_1 + r_2}$ (other components can be computed in a similar way). It is clear that the component $[\widehat{\mathbf{k}}_1 + \widehat{\mathbf{k}}_2]_2 \cdot \mathsf{F}(\mathtt{ID})^{r_1 + r_2}$ is identically distributed to a secret key directly computed by GenSK with $\widehat{\mathsf{MSK}}_1 \cdot \widehat{\mathsf{MSK}}_2$.[8] Hence, it satisfies evaluation correctness. We omit the case of the division since it is straightforward.

On the other hand, it seems difficult for HIBE schemes over pairing-free groups [DG17a] and lattice-based HIBE schemes [ABB10a, ABB10b, CHKP12] to satisfy MSK evaluatability since they do not have such a simple algebraic structure.

3 HKIBE

We review a definition of HKIBE based on [HHSI05, WS16, SW18] which present the most strict security model. Please keep in mind that an identity $\mathtt{id} \in \mathcal{I}$ in HKIBE is always a (non-hierarchical) one-dimensional vector.

3.1 Model

There are two types of keys, i.e., *helper keys* and *decryption keys*, and they depend on an identity $\mathtt{id}$ and each of the hierarchical time periods $\mathcal{T}_L, \ldots, \mathcal{T}_0$. Every user $\mathtt{id}$ has a level-ℓ *helper key* $\mathsf{hk}^{(\ell)}_{\mathtt{id}, \mathsf{T}_\ell(\mathtt{t})}$ for $\ell = 1, 2, \ldots, L$ and a *decryption key* $\mathsf{dk}_{\mathtt{id}, \mathsf{T}_0(\mathtt{t})}$. The upper level-$(\ell + 1)$ helper key $\mathsf{hk}^{(\ell+1)}_{\mathtt{id}, \mathsf{T}_{\ell+1}(\mathtt{t})}$ can derive a level-ℓ key update $\mathsf{ku}^{(\ell)}_{\mathtt{id}, \mathsf{T}_\ell(\mathtt{t})}$ for updating the lower level-ℓ helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id}, \mathsf{T}_\ell(\mathtt{t}')}$ to be $\mathsf{hk}^{(\ell)}_{\mathtt{id}, \mathsf{T}_\ell(\mathtt{t})}$. Similarly, the decryption key $\mathsf{dk}_{\mathtt{id}, \mathsf{T}_0(\mathtt{t})}$ is updated by using a key update derived from a

[8] To be precise, the component $[\widehat{\mathbf{k}}_1 + \widehat{\mathbf{k}}_2]_2 \cdot \mathsf{F}(\mathtt{ID})^{r_1 + r_2}$ should be re-randomized to satisfy evaluation correctness since it requires that given $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_1]$ and $\mathsf{SK}_{\mathtt{ID}}[\widehat{\mathsf{MSK}}_2]$, the two distributions are identical.

level-1 helper key. A ciphertext $\mathsf{ct}_{\mathtt{id},\mathtt{t}}$ of HKIBE depends on a receiver's identity $\mathtt{id} \in \mathcal{I}$ and actual time $\mathtt{t} \in \mathcal{T}_{act}$, and can be decrypted by a decryption key $\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t}')}$ if $\mathsf{T}_0(\mathtt{t}) = \mathsf{T}_0(\mathtt{t}')$.

Specifically, HKIBE consists of six algorithms $(\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{GenHK}, \mathsf{KeyUp}, \mathsf{Upd}, \mathsf{Decrypt})$ and proceeds as follows. First of all, the key generation center (KGC) runs Setup to generate a master-key pair $(\mathsf{pp}, \mathsf{mk})$. Upon a request from a user $\mathtt{id}$, the KGC runs GenHK to get a set of initial helper keys $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}$ as a secret key for $\mathtt{id}$. Suppose that each helper key is stored in a different (physically-secure) device. The level-0 helper key $\mathsf{hk}^{(0)}_{\mathtt{id},t_0}$ is used as a decryption key, and we often write it as $\mathsf{dk}_{\mathtt{id},t_0}$. A plaintext M is encrypted by $\mathsf{Encrypt}$ with not only an identity $\mathtt{id}$ but (current) time $\mathtt{t}$. The resulting ciphertext, which is denoted by $\mathsf{ct}_{\mathtt{id},\mathtt{t}}$, can be decrypted by $\mathsf{Decrypt}$ with $\mathtt{id}$'s decryption key $\mathsf{dk}_{\mathtt{id},t_0}$ $(= \mathsf{hk}^{(0)}_{\mathtt{id},t_0})$ if and only if $t_0 = \mathsf{T}_0(\mathtt{t})$. Here, we describe how to update helper and decryption keys as follows. Suppose that the user $\mathtt{id}$ has $(\mathsf{hk}^{(\ell)}_{\mathtt{id},t'_\ell})_{\ell\in[0,L]}$ and wants to update it for $\mathtt{t}$. The level-L helper key $\mathsf{hk}^{(L)}_{\mathtt{id},0}$ is never updated, and therefore, $\mathsf{hk}^{(L)}_{\mathtt{id},\mathsf{T}_L(\mathtt{t})} = \mathsf{hk}^{(L)}_{\mathtt{id},0}$ for any $\mathtt{t} \in \mathcal{T}_{act}$. For every $\ell = L-1, \ldots, 0$, the user $\mathtt{id}$ first runs KeyUp to generate $\mathtt{id}$'s level-ℓ *key update* $\mathsf{ku}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ by running KeyUp with $\mathsf{hk}^{(\ell+1)}_{\mathtt{id},\mathsf{T}_{\ell+1}(\mathtt{t})}$. The user then runs Upd with the key update $\mathsf{ku}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ to update $\mathtt{id}$'s level-ℓ helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id},t'_\ell}$ to $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$. At the end of this updating procedure, the user obtains a decryption key $\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})}$ $(= \mathsf{hk}^{(0)}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})})$.

Syntax. An HKIBE scheme Π consists of the six algorithms $(\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{GenHK}, \mathsf{KeyUp}, \mathsf{Upd}, \mathsf{Decrypt})$ defined as follows:

- $\mathsf{Setup}(1^\lambda, L) \to (\mathsf{pp}, \mathsf{mk})$: This is the *setup* algorithm that, given the security parameter λ and the maximum depth of the hierarchy $L \in \mathbb{N}$, it outputs a master-key pair $(\mathsf{pp}, \mathsf{mk})$.
- $\mathsf{Encrypt}(\mathsf{pp}, \mathtt{id}, \mathtt{t}, \mathsf{M}) \to \mathsf{ct}_{\mathtt{id},\mathtt{t}}$: This is the *encryption* algorithm that, given pp, an element identity $\mathtt{id} \in \mathcal{I}$, current time $\mathtt{t} \in \mathcal{T}_{act}$, and a plaintext $\mathsf{M} \in \mathcal{M}$, it outputs a ciphertext $\mathsf{ct}_{\mathtt{id},\mathtt{t}}$.
- $\mathsf{GenHK}(\mathsf{pp}, \mathsf{mk}, \mathtt{id}) \to (\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}$: This is the *helper-key generation* algorithm that, given pp, mk, and an element identity $\mathtt{id} \in \mathcal{I}$, it outputs a set of initial helper keys $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}$. The level-0 helper key is also called a decryption key and set as $\mathsf{dk}_{\mathtt{id},0} \coloneqq \mathsf{hk}^{(0)}_{\mathtt{id},0}$.
- $\mathsf{KeyUp}(\mathsf{pp}, \mathtt{t}, \mathsf{hk}^{(\ell+1)}_{\mathtt{id},t_{\ell+1}}) \to \mathsf{ku}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ or $\perp$: This is the *key update information generation* algorithm that, given pp, actual time $\mathtt{t} \in \mathcal{T}_{act}$, and an $\mathtt{id}$'s level-$(\ell+1)$ helper key $\mathsf{hk}^{(\ell+1)}_{\mathtt{id},t_{\ell+1}}$ at a time period $t_{\ell+1} \in \mathcal{T}_{\ell+1}$, it outputs an $\mathtt{id}$'s level-ℓ key update $\mathsf{ku}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ at a time period $\mathsf{T}_\ell(\mathtt{t})$ if $t_{\ell+1} = \mathsf{T}_{\ell+1}(\mathtt{t})$. Otherwise, it outputs $\perp$.
- $\mathsf{Upd}(\mathsf{pp}, \mathsf{hk}^{(\ell)}_{\mathtt{id},\tau_\ell}, \mathsf{ku}^{(\ell)}_{\mathtt{id},t_\ell}) \to \mathsf{hk}^{(\ell)}_{\mathtt{id},\tau_\ell}$: This is the *helper key update* algorithm that, given pp, an $\mathtt{id}$'s level-ℓ helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id},\tau_\ell}$ at a time period $\tau_\ell \in \mathcal{T}_\ell$, and an $\mathtt{id}$'s level-ℓ key update $\mathsf{ku}^{(\ell)}_{\mathtt{id},t_\ell}$ at a time period $t_\ell \in \mathcal{T}_\ell$, it outputs an updated helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id},t_\ell}$ at a time period t_ℓ.
- $\mathsf{Decrypt}(\mathsf{pp}, \mathsf{dk}_{\mathtt{id},t_0}, \mathsf{ct}_{\mathtt{id},\mathtt{t}}) \to \mathsf{M}$ or $\perp$: This is the *decryption* algorithm that, given pp, an $\mathtt{id}$'s decryption key $\mathsf{dk}_{\mathtt{id},t_0}$ at a time period $t_0 \in \mathcal{T}_0$, and a ciphertext $\mathsf{ct}_{\mathtt{id},\mathtt{t}}$, it outputs M or $\perp$ which indicates decryption failure.

Remark 1 (Update Frequency)**.** *For simplicity, we assume that the lower-level helper key is more frequently updated than the upper-level helper key. Namely, several level-ℓ helper keys $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(t^{(1)})}, \ldots, \mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(t^{(m)})}$ are updated by the same level-$(\ell+1)$ helper key $\mathsf{hk}^{(\ell+1)}_{\mathtt{id},t}$, where $t^{(1)}, \ldots, t^{(m)} \in \mathcal{T}_{act}$ and $t = \mathsf{T}_{\ell+1}(t^{(1)}) = \cdots = \mathsf{T}_{\ell+1}(t^{(m)})$. This assumption of use frequency captures actual situations: the upper level of helper keys is, the more rarely they should be used, i.e., the more isolated they should be from the Internet.*

Correctness. We require a ciphertext $\mathsf{ct}_{\mathtt{id},\mathtt{t}}$ associated with $(\mathtt{id},\mathtt{t})$ to be properly decrypted by a decryption key $\mathsf{dk}_{\mathtt{id},t_0}$ for the same $\mathtt{id}$ and $t_0 = \mathsf{T}_0(\mathtt{t})$ if $\mathsf{dk}_{\mathtt{id},t_0}$ is correctly generated from any *updating path.*

More formally, for all security parameter 1^λ, all hierarchical depth $L \in \mathbb{N}$, all $(\mathsf{pp},\mathsf{mk}) \leftarrow \mathsf{Setup}(1^\lambda, L)$, all $\mathsf{M} \in \mathcal{M}$, all $\mathtt{id} \in \mathcal{I}$, and all sequence $(\mathtt{t}_1,\dots,\mathtt{t}_n) \in \mathcal{T}_{act}^n$ for arbitrary number $n = \mathsf{poly}(\lambda)$, we consider the following experiment:

- $\mathsf{ct}_{\mathtt{id},\mathtt{t}_n} \leftarrow \mathsf{Encrypt}(\mathsf{pp},\mathtt{id},\mathtt{t}_n,\mathsf{M})$.
- $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]} \leftarrow \mathsf{GenHK}(\mathsf{pp},\mathsf{mk},\mathtt{id})$.
- Let $\mathtt{t}_0 := 0$ for simplicity. For all $j = 1,2,\dots,n$, execute the following procedures for $\ell = L-1, L-2,\dots,0$:
  - $\mathsf{ku}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t}_j)} \leftarrow \mathsf{KeyUp}(\mathsf{pp},\mathtt{t}_j,\mathsf{hk}^{(\ell+1)}_{\mathtt{id},\mathsf{T}_{\ell+1}(\mathtt{t}_j)})$.
  - $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t}_j)} \leftarrow \mathsf{Upd}(\mathsf{pp},\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t}_{j-1})},\mathsf{ku}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t}_j)})$.
- $\mathsf{M}' \leftarrow \mathsf{Decrypt}(\mathsf{pp},\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t}_n)},\mathsf{ct}_{\mathtt{id},\mathtt{t}_n})$.

Definition 3 (Correctness)**.** *We say that an HKIBE scheme* Π *with depth* L *satisfies* correctness*, if the probability* $\mathsf{M}' = \mathsf{M}$ *in the above experiment holds with overwhelming probability.*

3.2 Security

Let Π be an HKIBE scheme. We consider the adaptive-identity CPA security for HKIBE (the adaptive security for short), which is defined via a game between an adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. The game is parameterized by the security parameter λ and the maximum hierarchical depth $L \in \mathbb{N}$. Intuitively, $\mathcal{A}$ is able to receive all helper keys as long as they are insufficient for deriving a decryption key $\mathsf{dk}_{\mathtt{id}^\star,\mathtt{t}^\star}$ for the target tuple $(\mathtt{id}^\star,\mathtt{t}^\star)$. The game proceeds as follows:

$\mathcal{C}$ first runs $(\mathsf{pp},\mathsf{mk}) \leftarrow \mathsf{Setup}(1^\lambda, L)$ and gives pp to $\mathcal{A}$. $\mathcal{C}$ prepares $\mathtt{HKList}$ and stores all identity/initial helper keys $(\mathtt{id},(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]})$ generated during the game in $\mathtt{HKList}$ while we will not explicitly mention this procedure.

$\mathcal{A}$ may adaptively make the following four types of queries to $\mathcal{C}$:

Helper-Key Generation Query: Upon a query $\mathtt{id} \in \mathcal{I}$ from $\mathcal{A}$, $\mathcal{C}$ checks if $(\mathtt{id},*) \notin \mathtt{HKList}$, and returns $\perp$ to $\mathcal{A}$ if this is *not* the case. Otherwise, $\mathcal{C}$ executes $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]} \leftarrow \mathsf{GenHK}(\mathsf{pp},\mathsf{mk},\mathtt{id})$ and returns nothing to $\mathcal{A}$.

We require that all identities $\mathtt{id}$ appearing in the following queries (except the challenge query) are "activated", in the sense that $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}$ is generated via this query and hence $(\mathtt{id},(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}) \in \mathtt{HKList}$.

Initial Helper-Key Reveal Query: Until the challenge query, upon a query $\mathtt{id} \in \mathcal{I}$ from $\mathcal{A}$, $\mathcal{C}$ finds $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}$ from $\mathtt{HKList}$ and returns $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}$ to $\mathcal{A}$. After the challenge query, $\mathcal{C}$ checks whether $\mathtt{id} \neq \mathtt{id}^\star$ and returns $\perp$ if this is not the case. Otherwise, $\mathcal{C}$ returns $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}$ to $\mathcal{A}$ in the same way.

Key-Insulation Query: Until the challenge query, upon a query $(\mathtt{id},\mathtt{t},\ell) \in \mathcal{I} \times \mathcal{T}_{act} \times [0,L]$ from $\mathcal{A}$, $\mathcal{C}$ finds $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}$ from $\mathtt{HKList}$ and runs

- $\mathsf{ku}^{(i)}_{\mathtt{id},\mathsf{T}_i(\mathtt{t})} \leftarrow \mathsf{KeyUp}(\mathsf{pp},\mathtt{t},\mathsf{hk}^{(i+1)}_{\mathtt{id},\mathsf{T}_{i+1}(\mathtt{t})})$,
- $\mathsf{hk}^{(i)}_{\mathtt{id},\mathsf{T}_i(\mathtt{t})} \leftarrow \mathsf{Upd}(\mathsf{pp},\mathsf{hk}^{(i)}_{\mathtt{id},0},\mathsf{ku}^{(i)}_{\mathtt{id},\mathsf{T}_i(\mathtt{t})})$,

for $i = L-1, \dots, \ell$ to obtain $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$. Then, $\mathcal{C}$ returns $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ to $\mathcal{A}$. After the challenge query, when $\mathtt{id} = \mathtt{id}^\star$, $\mathcal{C}$ checks whether there exists the following *special hierarchical level* $\ell^\star$ after answering the query:

(i) $\mathsf{hk}^{(\ell^\star)}_{\mathtt{id}^\star,t_{\ell^\star}}$ of any $t_{\ell^\star} \in \mathcal{T}_{\ell^\star}$ are not revealed to $\mathcal{A}$. Namely, no level-$\ell^\star$ helper keys have not been revealed to $\mathcal{A}$ ever.

(ii) For all $i \in [0, \ell^\star - 1]$ and all $\mathtt{t} \in \mathcal{T}_{act}$ such that $\mathsf{T}_i(\mathtt{t}) = \mathsf{T}_i(\mathtt{t}^\star)$ holds, $\mathsf{hk}^{(i)}_{\mathtt{id}^\star,\mathsf{T}_i(\mathtt{t})}$ are not revealed to $\mathcal{A}$.

If this is not the case, $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ returns $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ to $\mathcal{A}$ in the same way.

Challenge Query: $\mathcal{A}$ is allowed to make this query only once. Upon a query $(\mathtt{id}^\star, \mathtt{t}^\star, \mathsf{M}^\star_0, \mathsf{M}^\star_1)$ from $\mathcal{A}$ such that $|\mathsf{M}^\star_0| = |\mathsf{M}^\star_1|$, $\mathcal{C}$ checks whether the following conditions simultaneously hold:

– $\mathcal{A}$ does not make the initial helper-key reveal query on $\mathtt{id}^\star$.

– There is a special hierarchical level $\ell^\star$ as explained in the key-insulation query.

If the conditions are not simultaneously satisfied, $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ picks a bit $b \in \{0,1\}$ uniformly at random, runs $\mathsf{ct}^\star_{\mathtt{id}^\star,\mathtt{t}^\star} \leftarrow \mathsf{Encrypt}(\mathsf{pp}, \mathtt{id}^\star, \mathtt{t}^\star, \mathsf{M}^\star_b)$, and returns the challenge ciphertext $\mathsf{ct}^\star_{\mathtt{id}^\star,\mathtt{t}^\star}$ to $\mathcal{A}$.

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

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

Definition 4 ([HHSI05]). *We say that an HKIBE scheme* Π *with depth* L *satisfies* adaptive-identity CPA security *(or* adaptive security *for brevity), if the advantage* $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$ *is negligible for all PPT adversaries* $\mathcal{A}$*.*

Why We Need the Restrictions. We briefly explain the restrictions (i) and (ii) appeared in key-insulation query, i.e., why we need the special hierarchical level $\ell^\star$. To define as strong security as possible while preventing trivial attacks, we should allow $\mathcal{A}$ to make as many queries as possible unless $\mathcal{A}$ can trivially create $\mathsf{dk}_{\mathtt{id}^\star,\mathsf{T}_0(\mathtt{t}^\star)}$. As for the restriction (i), if at least one level-i helper key for $\mathtt{id}^\star$ is leaked at every level $i \in [0, L]$, it also means that $\mathcal{A}$ can create all decryption keys including $\mathsf{dk}_{\mathtt{id}^\star,\mathsf{T}_0(\mathtt{t}^\star)}$. As for the restriction (ii), suppose that for some $i \in [0, \ell^\star - 1]$, $\mathcal{A}$ gets $(\mathsf{hk}^{(j)}_{\mathtt{id}^\star,\mathsf{T}_j(\mathtt{t}_j)})_{j\in[0,i-1]}$ such that $\mathsf{T}_j(\mathtt{t}_j) \neq \mathsf{T}_j(\mathtt{t}^\star)$ for $j \in [0, i-1]$ via key-insulation queries. Then, one helper key $\mathsf{hk}^{(i)}_{\mathtt{id}^\star,\mathsf{T}_i(\mathtt{t})}$ such that $\mathsf{T}_i(\mathtt{t}) = \mathsf{T}_i(\mathtt{t}^\star)$ is enough for $\mathcal{A}$ to compute a decryption key $\mathsf{dk}_{\mathtt{id}^\star,\mathsf{T}_0(\mathtt{t}^\star)}$ even if $\mathcal{A}$ has no level-$\ell^\star$ helper key $\mathsf{hk}^{(\ell^\star)}_{\mathtt{id}^\star,\mathsf{T}_{\ell^\star}(\mathtt{t})}$ for any $\mathtt{t} \in \mathcal{T}_{act}$. Hence, we need the restriction about the special level $\ell^\star$ to the key-insulation query for $\mathtt{id}^\star$. For the same reason, we, of course, disallow $\mathcal{A}$ to make the initial helper-key reveal query for $\mathtt{id}^\star$.

Selective Security and CCA Security. The selective-identity CPA security is analogously defined. The only exception is that $\mathcal{A}$ should send a challenge identity and time $(\mathtt{id}^\star, \mathtt{t}^\star)$ to $\mathcal{C}$ before receiving a master public key pp. Moreover, CCA security is also defined by allowing $\mathcal{A}$ to submit the following decryption query: upon a query $(\mathtt{id}, \mathtt{t}, \mathsf{ct}_{\mathtt{id},\mathtt{t}})$ $(\neq (\mathtt{id}^\star, \mathtt{t}^\star, \mathsf{ct}^\star_{\mathtt{id}^\star,\mathtt{t}^\star}))$ from $\mathcal{A}$, $\mathcal{C}$ returns $\mathsf{Decrypt}(\mathsf{pp}, \mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})}, \mathsf{ct}_{\mathtt{id},\mathtt{t}})$ to $\mathcal{A}$.

Remark 2 (Weak vs. Strong Security). *The security defined above is referred to as the* strong *security [DKXY02, HHSI05], which is the standard security requirement in key-insulated cryptography in the sense that* $\mathcal{A}$ *is allowed to get helper keys at a higher level than the special level* $\ell^\star$*. In the weak security definition, the special level* $\ell^\star$ *turns to the threshold level* $\ell^\star$*. Namely,* $\mathcal{A}$ *cannot get any helper keys at level* $\ell \in [\ell^\star, L]$*. As we claimed earlier, Bellare and Palacio's work [BP06] implies that any HIBE scheme can be transformed into an HKIBE scheme with weak security.*

Level	Elements	1st component		2nd component
L	$\mathsf{hk}^{(L)}_{\mathtt{id},\mathsf{T}_L(\mathtt{t})}$	$\mathsf{SK}_{\mathtt{id}}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,L-1]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]$		—
		$\downarrow$ *delegated from* $\mathtt{id}$ *to* $(\mathtt{id},\mathsf{T}_{L-1}(t))$		
$L-1$	$\mathsf{hk}^{(L-1)}_{\mathtt{id},\mathsf{T}_{L-1}(\mathtt{t})}$	$\mathsf{SK}_{(\mathtt{id},\mathsf{T}_{L-1}(\mathtt{t}))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,L-2]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]$	$\xleftarrow{\mathsf{EvalMSK}}$	$\widehat{\mathsf{MSK}}_{\mathtt{id},L-1}$
		$\downarrow$ *delegated from* $(\mathtt{id},\mathsf{T}_{L-1}(t))$ *to* $(\mathtt{id},\mathsf{T}_{[L-1,L-2]}(t))$		
$\dots$	$\vdots$	$\vdots$		$\vdots$
		$\downarrow$ *delegated from* $(\mathtt{id},\mathsf{T}_{[L-1,2]}(t))$ *to* $(\mathtt{id},\mathsf{T}_{[L-1,1]}(t))$		
1	$\mathsf{hk}^{(1)}_{\mathtt{id},\mathsf{T}_1(\mathtt{t})}$	$\mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,1]}(\mathtt{t}))}\left[\frac{\mathsf{MSK}}{\widehat{\mathsf{MSK}}_{\mathtt{id},0}}\right]$	$\xleftarrow{\mathsf{EvalMSK}}$	$\widehat{\mathsf{MSK}}_{\mathtt{id},1}$
		$\downarrow$ *delegated from* $(\mathtt{id},\mathsf{T}_{[L-1,1]}(t))$ *to* $(\mathtt{id},\mathsf{T}_{[L-1,0]}(t))$		
0	$\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})}$	$\mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))}[\mathsf{MSK}]$	$\xleftarrow{\mathsf{EvalMSK}}$	$\widehat{\mathsf{MSK}}_{\mathtt{id},0}$
CT	$\mathsf{ct}_{\mathtt{id},\mathtt{t}}$	$\mathsf{C}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))}$		

Figure 1: The intuition of our first construction.

Remark 3 (A Variant of Security Definition)**.** *One may consider a stronger variant of our security definition so that $\mathcal{A}$ can designate a derivation path of helper keys with a key-insulation query on* $(\mathtt{id}, t, \ell)$*; that is, $\mathcal{A}$ is allowed to designate how the helper key* $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(t)}$ *is derived. Since such a strong notion makes formalization complicated, we do not consider it for simplicity. Nevertheless, our constructions satisfy such a strong definition.*

4 Generic Construction from HIBE with MSK Evaluatability

We propose a generic construction of an HKIBE scheme with key-insulation depth L from an HIBE scheme with identity depth $L+1$ supporting *MSK evaluatability*. We assume that $\mathcal{I}, \mathcal{T}_0, \dots, \mathcal{T}_L \subseteq \mathcal{I}_{\text{HIBE}}$ holds, where $\mathcal{I}_{\text{HIBE}}$ is an element identity space of HIBE.

4.1 Construction Idea

The basic idea is quite simple: to encrypt a message M with an identity $\mathtt{id}$ and time $\mathtt{t}$, run the HIBE encryption algorithm Enc with M and a hierarchical identity $(\mathtt{id}, \mathsf{T}_{[L-1,0]}(\mathtt{t}))$ (i.e., $\mathsf{ct}_{\mathtt{id},\mathtt{t}} \coloneqq \mathsf{C}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))}$). Therefore, to make the decryption procedure consistent, we set a decryption key $\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})} \in \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))}$. However, if we set each helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})} \coloneqq \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}$ similarly, the resultant construction is the same as Bellare and Palacio's transformation [BP06] mentioned in Remark 2; it does not achieve strong security. Our construction's core spirit is that we use L pseudo-MSKs $\widehat{\mathsf{MSK}}_{\mathtt{id},0}, \dots, \widehat{\mathsf{MSK}}_{\mathtt{id},L-1}$ to mask the highest-level helper key $\mathsf{hk}^{(L)}_{\mathtt{id},\mathsf{T}_L(\mathtt{t})}$ and gradually remove them with $\mathsf{EvalMSK}$ as the hierarchy of key-insulation levels is lowered.[9] More specifically, when executing GenHK with any $\mathtt{id}$, we mask the true MSK MSK with all the pseudo-MSKs $\widehat{\mathsf{MSK}}_{\mathtt{id},0}, \dots, \widehat{\mathsf{MSK}}_{\mathtt{id},L-1}$ (in the fraction form) and set the highest-level helper key $\mathsf{hk}^{(L)}_{\mathtt{id},0}$ as

[9] Note that HKIBE is IBE with hierarchical key insulation, not HIBE with key insulation.

$\mathsf{SK}_{\mathtt{id}}[\mathsf{MSK}/\prod_{i=0}^{L-1}\widehat{\mathsf{MSK}}_{\mathtt{id},i}]$. Besides, for every $\ell \in [0, L-1]$, an level-ℓ helper key $\mathsf{hk}_{\mathtt{id},t_\ell}^{(\ell)}$ contains the pseudo-MSK $\widehat{\mathsf{MSK}}_{\mathtt{id},\ell}$ and is updated in the following two steps.

(1) Run GenSK with its higher-level helper key $\mathsf{hk}_{\mathtt{id},t_{\ell+1}}^{(\ell+1)} = \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell+1]}(\mathtt{t}))}[\mathsf{MSK}/\prod_{i=0}^{\ell}\widehat{\mathsf{MSK}}_{\mathtt{id},i}]$ to obtain $\mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1\ell]}(\mathtt{t}))}[\mathsf{MSK}/\prod_{i=0}^{\ell}\widehat{\mathsf{MSK}}_{\mathtt{id},i}]$.

(2) Run $\mathsf{EvalMSK}$ with the obtained key $\mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}[\mathsf{MSK}/\prod_{i=0}^{\ell}\widehat{\mathsf{MSK}}_{\mathtt{id},i}]$, the level-$\ell$ pseudo-MSK $\widehat{\mathsf{MSK}}_{\mathtt{id},\ell}$, and $\mathsf{lab} = \mathtt{mul}$ to get $\mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}[\mathsf{MSK}/\prod_{i=0}^{\ell-1}\widehat{\mathsf{MSK}}_{\mathtt{id},i}]$, which is set as an updated level-ℓ helper key $\mathsf{hk}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}^{(\ell)}$.

In the end, the mask is entirely removed at the lowest level, i.e., $\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})} \in \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))}[\mathsf{MSK}]$. We illustrate the idea in Figure 1. As can be seen above, all the masks cannot be removed unless an adversary gets secret keys at all levels; the adversary is not allowed to do so due to the security definition (i.e., there exists a special level $\ell^\star$ that the adversary cannot access).

4.2 Construction

Our HKIBE scheme $\Pi = (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{GenHK}, \mathsf{KeyUp}, \mathsf{Upd}, \mathsf{Decrypt})$ from an HIBE scheme $\Sigma = (\mathsf{Init}, \mathsf{Enc}, \mathsf{GenSK}, \mathsf{Dec}, \mathsf{SampMSK}, \mathsf{EvalMSK})$ is as follows.

- $\mathsf{Setup}(1^\lambda, L) \to (\mathsf{pp}, \mathsf{mk})$: Run $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Init}(1^\lambda, L+1)$, then output $\mathsf{pp} := \mathsf{MPK}$ and $\mathsf{mk} := \mathsf{MSK}$.
- $\mathsf{Encrypt}(\mathsf{pp}, \mathtt{id}, \mathtt{t}, \mathsf{M}) \to \mathsf{ct}_{\mathtt{id},\mathtt{t}}$: Parse $\mathsf{pp} = \mathsf{MPK}$. Run
  · $\mathsf{C}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))} \leftarrow \mathsf{Enc}(\mathsf{MPK}, (\mathtt{id}, \mathsf{T}_{[L-1,0]}(\mathtt{t})), \mathsf{M})$
  and output $\mathsf{ct}_{\mathtt{id},\mathtt{t}} := \mathsf{C}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))}$.
- $\mathsf{GenHK}(\mathsf{pp}, \mathsf{mk}, \mathtt{id}) \to (\mathsf{hk}_{\mathtt{id},0}^{(\ell)})_{\ell\in[0,L]}$: Parse $\mathsf{pp} = \mathsf{MPK}$ and $\mathsf{mk} = \mathsf{MSK}$. First, compute $\widehat{\mathsf{MSK}}_{\mathtt{id},\ell} \leftarrow \mathsf{SampMSK}(\mathsf{MPK})$ for $\ell \in [0, L-1]$. Then, for $\ell \in [0, L]$, run

$$\begin{cases} \mathsf{SK}_{\mathtt{id}}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,L-1]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right] \leftarrow \mathsf{GenSK}\left(\mathsf{MPK}, \frac{\mathsf{MSK}}{\prod_{i\in[0,L-1]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}, \mathtt{id}\right) & \text{if } \ell = L, \\ \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(0))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right] \leftarrow \mathsf{GenSK}\left(\mathsf{MPK}, \frac{\mathsf{MSK}}{\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}, (\mathtt{id}, \mathsf{T}_{[L-1,\ell]}(0))\right) & \text{if } \ell \in [L-1], \\ \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(0))}[\mathsf{MSK}] \leftarrow \mathsf{GenSK}(\mathsf{MPK}, \mathsf{MSK}, (\mathtt{id}, \mathsf{T}_{[L-1,0]}(0))) & \text{if } \ell = 0. \end{cases}$$

  Note that without loss of generality, we assume $0 \in \mathcal{T}_{act}$ to describe initial helper keys simply. Output $(\mathsf{hk}_{\mathtt{id},0}^{(\ell)})_{\ell\in[0,L]}$, where
  - $\mathsf{hk}_{\mathtt{id},0}^{(L)} := \mathsf{SK}_{\mathtt{id}}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,L-1]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]$,
  - $\mathsf{hk}_{\mathtt{id},0}^{(\ell)} := \left(\widehat{\mathsf{MSK}}_{\mathtt{id},\ell}, \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(0))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]\right)$ for $\ell \in [L-1]$.
  - $\mathsf{hk}_{\mathtt{id},0}^{(0)}(= \mathsf{dk}_{\mathtt{id},0}) := \left(\widehat{\mathsf{MSK}}_{\mathtt{id},0}, \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(0))}\right)$.
- $\mathsf{KeyUp}(\mathsf{pp}, \mathtt{t}, \mathsf{hk}_{\mathtt{id},t_{\ell+1}}^{(\ell+1)}) \to \mathsf{ku}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}^{(\ell)}$ or $\perp$: If $t_{\ell+1} \neq \mathsf{T}_{\ell+1}(\mathtt{t})$, output $\perp$. Otherwise, parse
  ▷ $\mathsf{pp} = \mathsf{MPK}$

▷ $\mathsf{hk}^{(\ell+1)}_{\mathtt{id},t_{\ell+1}} = \left(\widehat{\mathsf{MSK}}_{\mathtt{id},\ell+1}, \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell+1]}(\mathtt{t}))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]\right)$.

Run

· $\mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]$

$$\leftarrow \mathsf{GenSK}\left(\mathsf{MPK}, \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell+1]}(\mathtt{t}))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right], (\mathtt{id}, \mathsf{T}_{[L-1,\ell]}(\mathtt{t}))\right),$$

and output $\mathsf{ku}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})} = \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]$.

- $\mathsf{Upd}(\mathsf{pp}, \mathsf{hk}^{(\ell)}_{\mathtt{id},\tau_\ell}, \mathsf{ku}^{(\ell)}_{\mathtt{id},t_\ell}) \to \mathsf{hk}^{(\ell)}_{\mathtt{id},t_\ell}$: Suppose $\tau_\ell = \mathsf{T}_\ell(\mathtt{t})$ and $t_\ell = \mathsf{T}_\ell(\mathtt{t}')$. Parse

  ▷ $\mathsf{pp} = \mathsf{MPK}$,

  ▷ $\mathsf{hk}^{(\ell)}_{\mathtt{id},\tau_\ell} = \left(\widehat{\mathsf{MSK}}_{\mathtt{id},\ell}, \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]\right)$,

  ▷ $\mathsf{ku}^{(\ell)}_{\mathtt{id},t_\ell} = \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}'))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]$.

  Run

  · $\mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}'))}\left[\widehat{\mathsf{MSK}}_{\mathtt{id},\ell}\right] \leftarrow \mathsf{GenSK}\left(\mathsf{MPK}, \widehat{\mathsf{MSK}}_{\mathtt{id},\ell}, (\mathtt{id}, \mathsf{T}_{[L-1,\ell]}(\mathtt{t}'))\right)$,

  · $\mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}'))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]$

  $$\leftarrow \mathsf{EvalMSK}\left(\mathsf{MPK}, \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}'))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right], \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}'))}\left[\widehat{\mathsf{MSK}}_{\mathtt{id},\ell}\right], \mathtt{mul}\right),$$

  Output $\mathsf{hk}^{(\ell)}_{\mathtt{id},t_\ell} := \left(\widehat{\mathsf{MSK}}_{\mathtt{id},\ell}, \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}'))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id},i}}\right]\right)$.

  As the special case for $\ell = 0$, $\mathsf{hk}^{(0)}_{\mathtt{id},t_0} := (\widehat{\mathsf{MSK}}_{\mathtt{id},0}, \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}'))})$.

- $\mathsf{Decrypt}(\mathsf{pp}, \mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})}, \mathsf{ct}_{\mathtt{id},\mathtt{t}}) \to \mathsf{M}$: Parse

  ▷ $\mathsf{pp} = \mathsf{MPK}$,

  ▷ $\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})} = \mathsf{hk}^{(0)}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})} = (\widehat{\mathsf{MSK}}_{\mathtt{id},0}, \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))})$,

  ▷ $\mathsf{ct}_{\mathtt{id},\mathtt{t}} = \mathsf{C}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))}$.

  Run and output

  – $\mathsf{M} \leftarrow \mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))}, \mathsf{C}_{(\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))})$.

Correctness. Thanks to the evaluation correctness of MSK evaluatability, decryption keys of our HKIBE scheme follow the same distributions as those of the underlying HIBE scheme; hence, the correctness of our HKIBE scheme readily follows from that of the underlying HIBE scheme.

4.3 Security

The security of the HKIBE scheme is reduced to from that of the underlying HIBE scheme supporting MSK evaluatability.

Theorem 1. *If the underlying HIBE scheme with hierarchical depth $L+1$ supporting MSK evaluatability satisfies adaptive security, then the above HKIBE scheme with hierarchical depth L also satisfies adaptive security. Specifically, if there exists an adversary $\mathcal{A}$ to break adaptive security of the above HKIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$, then there exists a reduction algorithm $\mathcal{B}$ to break adaptive security of the underlying HIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+1,\mathcal{B}}(\lambda) \geq \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)/\Theta(QL)$, where Q denotes the number of helper-key generation queries.*

Proof Overview. First of all, we divide $\mathcal{A}$'s attack strategy into $L+1$ types with respect to a special hierarchical level $\ell^\star \in [0, L]$ defined in the key-insulation query. Let $\mathcal{A}_{\ell^\star}$ be an adversary $\mathcal{A}$ that makes key-insulation queries so that there exists a special level $\ell^\star$. Since this covers all the possible strategies, the proof against a fixed $\mathcal{A}_{\ell^\star}$ is sufficient for a proof against $\mathcal{A}$ of a general strategy with $\Theta(L)$ reduction loss.

Now, we use $\mathcal{A}_{\ell^\star}$ as a building block and construct a reduction algorithm $\mathcal{B}_{\ell^\star}$ against the underlying HIBE scheme. The main observation is that $\mathcal{B}_{\ell^\star}$ can answer all $\mathcal{A}_{\ell^\star}$'s queries by making HIBE secret-key reveal queries for the corresponding identity, say, $(\mathtt{id}, \mathsf{T}_{[L-1,\ell]}(\mathtt{t}))$, as long as it holds

$$(\mathtt{id}, \mathsf{T}_{[L-1,\ell]}(\mathtt{t})) \notin \mathsf{prefix}^+((\mathtt{id}^\star, \mathsf{T}_{[L-1,0]}(\mathtt{t}^\star))) \tag{1}$$

even without the knowledge of the challenge tuple $(\mathtt{id}^\star, \mathtt{t}^\star)$. Obviously, $\mathcal{B}_{\ell^\star}$ answers all queries for $\mathtt{id}$ $(\neq \mathtt{id}^\star)$ by making HIBE secret-key reveal queries since such a case always satisfies the condition (1). Therefore, the challenge is how $\mathcal{B}_{\ell^\star}$ answers $\mathcal{A}_{\ell^\star}$'s queries for $\mathtt{id}^\star$, which might not meet the condition (1). Roughly speaking, we look at the MSK-part of $(\mathsf{hk}^{(\ell)}_{\mathtt{id}^\star,0})_{\ell \in [0,L]}$ differently: In the construction, for every $\ell \in [0, L]$, the MSK-part of $\mathsf{hk}^{(\ell)}_{\mathtt{id}^\star,0}$ is $\mathsf{MSK}/\prod_{i \in [0,\ell-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}^\star,i}$, where $\widehat{\mathsf{MSK}}_{\mathtt{id}^\star,0}, \ldots, \widehat{\mathsf{MSK}}_{\mathtt{id}^\star,L-1}$ are pseudo-MSKs for $\mathtt{id}^\star$. In this proof, $\mathcal{B}_{\ell^\star}$ samples a level-ℓ pseudo-MSK $\widehat{\mathsf{MSK}}_{\mathtt{id}^\star,\ell}$ for $\ell \in [0, L] \setminus \{\ell^\star\}$. Note that $\widehat{\mathsf{MSK}}_{\mathtt{id}^\star,L}$ is picked instead of $\widehat{\mathsf{MSK}}_{\mathtt{id}^\star,\ell^\star}$. Then, $\mathcal{B}_{\ell^\star}$ (implicitly) sets $\widehat{\mathsf{MSK}}_{\mathtt{id}^\star,\ell^\star} := \mathsf{MSK}/\prod_{i \in [0,L] \setminus \{\ell^\star\}} \widehat{\mathsf{MSK}}_{\mathtt{id}^\star,i}$. All the above pseudo-MSKs are properly distributed. A key-insulation query $(\mathtt{id}^\star, \mathtt{t}, \ell)$ that contradicts the condition (1) satisfies $\mathsf{T}_\ell(\mathtt{t}) = \mathsf{T}_\ell(\mathtt{t}^\star)$ and $\ell \in [\ell^\star + 1, L]$ since a query that satisfies $\mathsf{T}_\ell(\mathtt{t}) = \mathsf{T}_\ell(\mathtt{t}^\star)$ is not allowed for $\ell \in [0, \ell^\star - 1]$ by definition. Indeed, $\mathcal{B}_{\ell^\star}$ can answer such a query since the corresponding helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id}^\star,\mathsf{T}_\ell(\mathtt{t})}$ can be computed with only the above pseudo-MSKs by $\mathcal{B}_{\ell^\star}$ itself. In particular, the distribution of the helper keys is identically distributed as that of helper keys created as in the construction thanks to *pseudo-MSK indistinguishability* in Def. 2. Note that $\mathcal{A}_{\ell^\star}$ does not make any key-insulation query for $\ell^\star$ by definition. On the other hand, a key-insulation query $(\mathtt{id}^\star, \mathtt{t}, \ell)$ for $\ell \in [0, \ell^\star - 1]$ such that $\mathsf{T}_\ell(\mathtt{t}) \neq \mathsf{T}_\ell(\mathtt{t}^\star)$ satisfies the condition (1). Therefore, $\mathcal{B}_{\ell^\star}$ makes an HIBE secret-key reveal query on $(\mathtt{id}^\star, \mathsf{T}_{[L-1,\ell]}(\mathtt{t}))$ to get $\mathsf{SK}_{(\mathtt{id}^\star,\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}[\mathsf{MSK}]$, and runs $\mathsf{EvalMSK}$ to return $\mathsf{hk}^{(\ell)}_{\mathtt{id}^\star,\mathsf{T}\ell}$ to $\mathcal{A}_{\ell^\star}$. The output is properly distributed thanks to the *evaluation correctness* in Def. 2.

The above simulation can be done only when $\mathcal{B}_{\ell^\star}$ knows $\mathtt{id}^\star$ (i.e., selective security). Nonetheless, it is applicable even to adaptive security by *guessing when the target identity* $\mathtt{id}^\star$ *is queried at the beginning of the game*: let $\mathtt{id}_q$ be an identity on which $\mathcal{A}_{\ell^\star}$ makes q-th helper-key generation query, and $\mathcal{B}_{\ell^\star}$ first guesses the number $Q^\star \in [Q]$ such that $\mathtt{id}_{Q^\star} = \mathtt{id}^\star$ with $\Theta(Q)$ reduction loss.[10] Then, $\mathcal{B}_{\ell^\star}$ sets the pseudo-MSKs for $\mathtt{id}_{Q^\star}$ as above, instead of $\mathtt{id}^\star$, although it will turn out $\mathtt{id}_{Q^\star} = \mathtt{id}^\star$ after the challenge query.

Theorem 1. We formally describe the proof as follows. As above, let $\mathcal{B}_{\ell^\star}$ be a reduction algorithm against the underlying HIBE scheme. We show how to construct $\mathcal{B}_{\ell^\star}$ by using $\mathcal{A}_{\ell^\star}$ as follows. First of all, please keep in mind that $\mathcal{B}_{\ell^\star}$ will set

$$(\mathtt{id}^\star, \mathsf{T}_{[L-1,0]}(\mathtt{t}^\star))$$

[10] Strictly speaking, we have to consider the case of $Q^\star = 0$, which means the adversary never makes a helper-key generation query (and the corresponding queries) on $\mathtt{id}^\star$. Since we can consider such an adversary as $\mathcal{A}_{L+1}$ and give a proof in a similar way, we omit the proof.

as the challenge identity in the HIBE security game. Therefore, $\mathcal{B}_{\ell^\star}$ does not make HIBE secret-key reveal queries on the challenge identity itself and its prefix identities during the game. At first, $\mathcal{B}_{\ell^\star}$ is given an HIBE's master public key MPK from an HIBE challenger $\mathcal{C}$. Then, $\mathcal{B}_{\ell^\star}$ initializes $\mathtt{HKList} = \emptyset$ and sends $\mathsf{pp} := \mathsf{MPK}$ to an HKIBE adversary $\mathcal{A}_{\ell^\star}$.

Let $\mathtt{id}_q$ be an identity on which $\mathcal{A}_{\ell^\star}$ makes a q-th helper-key generation query. Then, $\mathcal{B}_{\ell^\star}$ guesses the number $Q^\star \in [Q]$ such that $\mathtt{id}_{Q^\star} = \mathtt{id}^\star$. If the guess is incorrect, $\mathcal{B}_{\ell^\star}$ outputs a random bit and aborts the game. The guess is correct with probability $1/Q$. In the following, we assume that the guess is correct.

$\mathcal{B}_{\ell^\star}$ answers $\mathcal{A}_{\ell^\star}$'s queries by interacting with $\mathcal{C}$ as follows:

Helper-Key Generation Query: Upon a query $\mathtt{id}_q$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ checks if $(\mathtt{id}, \cdot) \notin \mathtt{HKList}$ holds, and returns $\perp$ to $\mathcal{A}_{\ell^\star}$ if this is *not* the case. Otherwise, $\mathcal{B}_{\ell^\star}$ proceeds as follows:

Case for $q \neq Q^\star$: $\mathcal{B}_{\ell^\star}$ makes secret-key reveal queries on $((\mathtt{id}_q, \mathsf{T}_{[L-1,\ell]}(0)))_{\ell\in[0,L]}$ and receives $(\mathsf{SK}_{(\mathtt{id}_q,\mathsf{T}_{[L-1,\ell]}(0))})_{\ell\in[0,L]}$.[11] $\mathcal{B}_{\ell^\star}$ runs

- $\widehat{\mathsf{MSK}}_{\mathtt{id}_q,\ell} \leftarrow \mathsf{SampMSK}(\mathsf{MPK})$ for $\ell \in [0, L-1]$

Then, for $\ell \in [1, L]$, $\mathcal{B}_{\ell^\star}$ executes

- $\mathsf{SK}_{(\mathtt{id}_q,\mathsf{T}_{[L-1,\ell]}(0))}\Big[\prod_{i\in[0,\ell-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}_q,i}\Big] \leftarrow \mathsf{GenSK}(\mathsf{MPK}, \prod_{i\in[0,\ell-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}_q,i}, (\mathtt{id}_q, \mathsf{T}_{[L-1,\ell]}(0)))$,
- $\mathsf{SK}_{(\mathtt{id}_q,\mathsf{T}_{[L-1,\ell]}(0))}\Big[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}_q,i}}\Big]$ $\leftarrow \mathsf{EvalMSK}(\mathsf{MPK}, \mathsf{SK}_{(\mathtt{id}_q,\mathsf{T}_{[L-1,\ell]}(0))}, \mathsf{SK}_{(\mathtt{id}_q,\mathsf{T}_{[L-1,\ell]}(0))}\Big[\prod_{i\in[0,\ell-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}_q,i}\Big], \mathtt{div})$,

and stores an initial helper key $(\mathtt{id}_q, (\mathsf{hk}^{(\ell)}_{\mathtt{id}_q,0})_{\ell\in[0,L]})$ in $\mathtt{HKList}$, where

- $\mathsf{hk}^{(L)}_{\mathtt{id}_q,0} := \mathsf{SK}_{\mathtt{id}_q}\Big[\frac{\mathsf{MSK}}{\prod_{i\in[0,L-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}_q,i}}\Big]$,
- $\mathsf{hk}^{(\ell)}_{\mathtt{id}_q,0} := \Big(\widehat{\mathsf{MSK}}_{\mathtt{id}_q,\ell}, \mathsf{SK}_{(\mathtt{id}_q,\mathsf{T}_{[L-1,\ell]}(0))}\Big[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}_q,i}}\Big]\Big)$ for $\ell \in [0, L-1]$.

Observe that the helper keys created by $\mathcal{B}_{\ell^\star}$ are properly distributed as follows: For every $\ell \in [1, L]$, $\mathsf{SK}_{(\mathtt{id}_q,\mathsf{T}_{[L-1,\ell]}(0))}\Big[\mathsf{MSK}/\prod_{i\in[0,L-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}_q,i}\Big]$, which is the component of the level-ℓ helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id}_q,0}$, is created by running

- $\mathsf{SK}_{(\mathtt{id}_q,\mathsf{T}_{[L-1,\ell]}(0))}\Big[\frac{\mathsf{MSK}}{\prod_{i\in[0,L-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}_q,i}}\Big] \leftarrow \mathsf{GenSK}\Big(\mathsf{MPK}, \frac{\mathsf{MSK}}{\prod_{i\in[0,L-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}_q,i}}, (\mathtt{id}_q, \mathsf{T}_{[L-1,\ell]}(0))\Big)$,

in the construction while $\mathcal{B}_{\ell^\star}$ runs $\mathsf{EvalMSK}$ algorithm in the reduction. Thanks to the *evaluation correctness* in Def. 2, the both of $\mathsf{SK}_{(\mathtt{id}_q,\mathsf{T}_{[L-1,\ell]}(0))}[\mathsf{MSK}/\prod_{i\in[0,L-1]} \widehat{\mathsf{MSK}}_{\mathtt{id}_q,i}]$ follow the same distribution. Note that the case for $\ell = 0$ (i.e., $\mathsf{SK}_{(\mathtt{id}_q,\mathsf{T}_{[L-1,0]}(0))}$) is the same procedure as in the construction.

Case for $q = Q^\star$: $\mathcal{B}_{\ell^\star}$ runs

- $\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell} \leftarrow \mathsf{SampMSK}(\mathsf{MPK})$ for $\ell \in [0, L] \setminus \{\ell^\star\}$,
- $\mathsf{SK}_{(\mathtt{id}_{Q^\star},\mathsf{T}_{[L-1,\ell]}(0))}\Big[\prod_{i\in[\ell,L]} \widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}\Big] \leftarrow \mathsf{GenSK}\Big(\mathsf{MPK}, \prod_{i\in[\ell,L]} \widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}, (\mathtt{id}_{Q^\star}, \mathsf{T}_{[L-1,\ell]}(0))\Big)$ for $\ell \in [\ell^\star + 1, L]$,

[11] In the following, we use $(\mathtt{id}, \mathsf{T}_{[L-1,L]}(\mathtt{t}))$ as the alternative expression of $\mathtt{id}$ for compact notation. Similarly, we suppose $\prod_{i\in[0,-1]} \widehat{\mathsf{MSK}}_{\mathtt{id},i} := 1$.

and stores *a part of* an initial helper key $(\mathtt{id}_{Q^\star}, (\mathsf{hk}^{(\ell)}_{\mathtt{id}_{Q^\star},0})_{\ell\in[\ell^\star+1,L]}, (\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell})_{\ell\in[0,\ell^\star]})$ in HKList, where

- $\mathsf{hk}^{(\ell)}_{\mathtt{id}_{Q^\star},0} := \Big(\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell}, \mathsf{SK}_{(\mathtt{id}_{Q^\star},\mathsf{T}_{[L-1,\ell]}(0))}\Big[\prod_{i\in[\ell,L]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}\Big]\Big)$ for $\ell\in[\ell^\star+1,L]$,

$\mathcal{B}_{\ell^\star}$ does not create the rest of the initial helper key $(\mathsf{hk}^{(\ell)}_{\mathtt{id}^\star,0})_{\ell\in[0,\ell^\star]}$ at this point, and on key-insulation query, helper keys for any $\mathtt{t}\in\mathcal{T}_{act}$ and $\ell\in[0,\ell^\star-1]$ will be computed directly from the pseudo-MSKs, not the corresponding initial helper keys.

We show that the level-ℓ helper keys for $\ell\in[\ell^\star+1,L]$ and the pseudo-MSKs created above and are properly distributed as follows. Since the above procedure is the same as that of the construction if $\ell^\star=L$, we consider the case for $\ell^\star\neq L$. In the case, $\mathcal{B}_{\ell^\star}$ implicitly sets

- $\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell} = \widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell}$ for $\ell\in[0,L]\setminus\{\ell^\star\}$,
- $\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell^\star} = \dfrac{\mathsf{MSK}}{\prod_{i\in[0,L]\setminus\{\ell^\star\}}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}}$.

Case for $\ell\in[0,\ell^\star-1]$: The level-ℓ pseudo-MSK $\mathsf{MSK}_{\mathtt{id}_{Q^\star},\ell} = \widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell}$ is created in the same way as the construction by running SampMSK algorithm.

Case for $\ell^\star$: The level-$\ell^\star$ pseudo-MSK $\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell^\star}$ is created by running SampMSK algorithm in the construction (i.e., $\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell^\star}$ is the output of SampMSK). In the reduction, we assume that the level-$\ell^\star$ pseudo-MSK is $\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell^\star} = \mathsf{MSK}/\prod_{i\in[0,L-1]\setminus\{\ell^\star\}}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}$ although $\mathcal{B}_{\ell^\star}$ does not compute it explicitly. Thanks to the *pseudo-MSK indistinguishability* in Def. 2, the level-$\ell^\star$ pseudo-MSK $\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell^\star}$ follows the same distribution as the construction.

Case for $\ell\in[\ell^\star+1,L]$: First of all, the main component $\mathsf{SK}_{\mathtt{id}_{Q^\star}}[\mathsf{MSK}/\prod_{i\in[0,L-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}]$ of level-L helper key $\mathsf{hk}^{(L)}_{\mathtt{id}_{Q^\star},0}$ is created by

- $\mathsf{SK}_{\mathtt{id}_{Q^\star}}\left[\dfrac{\mathsf{MSK}}{\prod_{i\in[0,L-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}}\right] \leftarrow \mathsf{GenSK}\left(\mathsf{MPK}, \dfrac{\mathsf{MSK}}{\prod_{i\in[0,L-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}}, \mathtt{id}_{Q^\star}\right)$

in the construction. The difference between the construction and the reduction is that the MSK-part $\mathsf{MSK}/\prod_{i\in[0,L-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}$ is replaced by the pseudo-MSK $\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},L}$. Observe that

$$\begin{aligned}\frac{\mathsf{MSK}}{\prod_{i\in[0,L-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}} &= \frac{\mathsf{MSK}}{\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell^\star}\cdot\prod_{i\in[0,L-1]\setminus\{\ell^\star\}}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}}\\ &= \frac{\mathsf{MSK}}{(\mathsf{MSK}/\prod_{i\in[0,L]\setminus\{\ell^\star\}}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i})\cdot\prod_{i\in[0,L-1]\setminus\{\ell^\star\}}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}}\\ &= \widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},L}.\end{aligned}$$

Therefore, thanks to the *evaluation correctness* in Def. 2, the level-L helper key follows the same distribution as in the construction. Similarly, for $\ell\in[\ell^\star+1,L-1]$ the main component of level-ℓ helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id}_{Q^\star},0}$ is $\mathsf{SK}_{\mathtt{id}_{Q^\star}}[\mathsf{MSK}/\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}]$ in the construction, and its MSK-part is replaced with $\prod_{i\in[\ell,L]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}$ in the reduction. It is easy to see that the level-ℓ helper key follows the same distribution as in the construction thanks to the *evaluation correctness* in Def. 2.

Initial Helper-Key Reveal Queries: Upon a query $\mathtt{id}_q$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ finds $(\mathsf{hk}^{(\ell)}_{\mathtt{id}_q,0})_{\ell\in[0,L]}$ from HKList and returns $(\mathsf{hk}^{(\ell)}_{\mathtt{id}_q,0})_{\ell\in[0,L]}$ to $\mathcal{A}_{\ell^\star}$. $\mathcal{B}_{\ell^\star}$ can answer all $\mathcal{A}_{\ell^\star}$'s queries since $\mathcal{A}_{\ell^\star}$ does not make the query on $\mathtt{id}^\star$ $(=\mathtt{id}_{Q^\star})$ due to the restriction in the query.

Key-Insulation Query: Upon a query $(\mathtt{id}_q, \mathtt{t}, \ell)$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ proceeds as follows:

Case for $\mathtt{id}_q \neq \mathtt{id}_{Q^\star}$: $\mathcal{B}_{\ell^\star}$ finds the initial helper keys $(\mathsf{hk}^{(\ell)}_{\mathtt{id}_q,0})_{\ell\in[0,L]}$ from $\mathtt{HKList}$ and creates $\mathsf{hk}^{(\ell)}_{\mathtt{id}_q,\mathsf{T}_\ell(\mathtt{t})}$ in the same way as the construction.

Case for $\mathtt{id}_q = \mathtt{id}_{Q^\star}$: Due to the Type-$\ell^\star$ strategy and the restriction on the special level $\ell^\star$, $\mathcal{A}_{\ell^\star}$ does not make any key-insulation queries on $\ell = \ell^\star$. $\mathcal{B}_{\ell^\star}$ finds *a part of* the initial helper key $((\mathsf{hk}^{(\ell)}_{\mathtt{id}_{Q^\star},0})_{\ell\in[\ell^\star+1,L]}, (\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell})_{\ell\in[0,\ell^\star]})$ from $\mathtt{HKList}$ and performs as follows:

Level-ℓ for $\ell \in [\ell^\star+1, L]$: $\mathcal{B}_{\ell^\star}$ creates $\mathsf{hk}^{(\ell)}_{\mathtt{id}_{Q^\star},\mathsf{T}_\ell(\mathtt{t})}$ in the same way as the construction.

Level-ℓ for $\ell \in [0, \ell^\star - 1]$: $\mathcal{B}_{\ell^\star}$ first makes an HIBE secret-key reveal query on $(\mathtt{id}_{Q^\star}, \mathsf{T}_{[L-1,\ell]}(\mathtt{t}))$ and receives $\mathsf{SK}_{(\mathtt{id}_{Q^\star},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}$ from $\mathcal{C}$. Then, $\mathcal{B}_{\ell^\star}$ uses $(\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i})_{i\in[0,\ell-1]}$ to run

- $\mathsf{SK}_{(\mathtt{id}_{Q^\star},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}\left[\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}\right]$
  $\leftarrow \mathsf{GenSK}(\mathsf{MPK}, \prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}, (\mathtt{id}_{Q^\star}, \mathsf{T}_{[L-1,\ell]}(\mathtt{t})))$,
- $\mathsf{SK}_{(\mathtt{id}_{Q^\star},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}}\right]$
  $\leftarrow \mathsf{EvalMSK}(\mathsf{MPK}, \mathsf{SK}_{(\mathtt{id}_{Q^\star},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}, \mathsf{SK}_{(\mathtt{id}_{Q^\star},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}\left[\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}\right],$ $\mathtt{div})$.

Finally, $\mathcal{B}_{\ell^\star}$ returns

$$\mathsf{hk}^{(\ell)}_{\mathtt{id}_{Q^\star},\mathtt{t}} = \left(\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},\ell}, \mathsf{SK}_{(\mathtt{id}_{Q^\star},\mathsf{T}_{[L-1,\ell]}(\mathtt{t}))}\left[\frac{\mathsf{MSK}}{\prod_{i\in[0,\ell-1]}\widehat{\mathsf{MSK}}_{\mathtt{id}_{Q^\star},i}}\right]\right)$$

to $\mathcal{A}_{\ell^\star}$. Observe that the level-ℓ helper key is properly distributed thanks to the *evaluation correctness* in Def. 2.

Challenge Query: Upon a query $(\mathtt{id}^\star, \mathtt{t}^\star, \mathsf{M}^\star_0, \mathsf{M}^\star_1)$ from $\mathcal{A}_{\ell^\star}$ such that $|\mathsf{M}^\star_0| = |\mathsf{M}^\star_1|$, $\mathcal{B}_{\ell^\star}$ makes a challenge query on $((\mathtt{id}^\star, \mathsf{T}_{[L-1,0]}(\mathtt{t}^\star)), \mathsf{M}^\star_0, \mathsf{M}^\star_1)$ and receives an HIBE challenge ciphertext $\mathsf{C}_{(\mathtt{id}^\star,\mathsf{T}_{[L-1,0]}(\mathtt{t}^\star))}$. Then, $\mathcal{B}_{\ell^\star}$ sends an HKIBE challenge ciphertext $\mathsf{ct}_{\mathtt{id}^\star,\mathtt{t}^\star} := \mathsf{C}_{(\mathtt{id}^\star,\mathsf{T}_{[L-1,0]}(\mathtt{t}^\star))}$ to $\mathcal{A}_{\ell^\star}$.

Observe that $\mathsf{C}_{(\mathtt{id}^\star,\mathsf{T}_{[L-1,0]}(\mathtt{t}^\star))}$ is created in the same way as the construction.

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

The above completes the description of $\mathcal{B}_{\ell^\star}$. Observe that $\mathcal{B}_{\ell^\star}$ can make all $\mathcal{A}$'s queries with $\mathcal{C}$. $\mathcal{B}_{\ell^\star}$ makes the HIBE challenge query on

$$(\mathtt{id}^\star, \mathsf{T}_{[L-1,0]}(\mathtt{t}^\star)),$$

while $\mathcal{B}_{\ell^\star}$ makes HIBE secret-key reveal queries on the following identity $\mathtt{id}_q$ ($q \in [Q]$):

- For all $q \in [Q] \setminus \{Q^\star\}$, we have $\mathtt{id}_q \notin \mathsf{prefix}^+((\mathtt{id}^\star, \mathsf{T}_{[L-1,0]}(\mathtt{t}^\star)))$. Therefore, $\mathcal{B}_{\ell^\star}$ can make the HIBE secret-key reveal queries on $(\mathtt{id}_q, \mathsf{T}_{[L-1,\ell]}(0))_{\ell\in[0,L]}$ for the helper-key generation query on $\mathtt{id}_q$ (if $\mathcal{B}_{\ell^\star}$'s guess of the number $Q^\star$ is correct).
- For $q = Q^\star$ (i.e., $\mathtt{id}_q = \mathtt{id}_{Q^\star} = \mathtt{id}^\star$), $\mathcal{A}_{\ell^\star}$ might make a key-insulation query on $(\mathtt{id}_{Q^\star}, \mathsf{T}_{[L-1,\ell]}(\mathtt{t}))$ for $\mathtt{t} \in \mathcal{T}_{act}$ and $\ell \in [0, \ell^\star - 1]$. $\mathcal{B}_{\ell^\star}$ can make the HIBE secret-key generation query on $(\mathtt{id}_{Q^\star}, \mathsf{T}_{[L-1,\ell]}(\mathtt{t}))$ since it holds $(\mathtt{id}_{Q^\star}, \mathsf{T}_{[L-1,\ell]}(\mathtt{t})) \notin \mathsf{prefix}^+((\mathtt{id}^\star, \mathsf{T}_{[L-1,0]}(\mathtt{t}^\star)))$ due to the restriction on the special level $\ell^\star$.[12]

[12] For $\ell \in [\ell^\star + 1, L]$, $\mathcal{B}_{\ell^\star}$ can respond to the key-insulation query by itself since $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ does not contain the true-MSK.

As we already observed, $\mathcal{B}_{\ell^\star}$ perfectly simulates the adaptive security game against $\mathcal{A}_{\ell^\star}$ with probability $1/Q$. Since the probability that β' is a correct guess is the same as that of b', $\mathcal{B}_{\ell^\star}$'s advantage is $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}_{\ell^\star}}(\lambda) = Q \cdot \mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+1,\mathcal{B}_{\ell^\star}}(\lambda)$.

Therefore, $\mathcal{B}$'s advantage against $\mathcal{A}$ of general attack strategy is $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L+1,\mathcal{A}}(\lambda) = \sum_{\ell^\star\in[0,L]} Q \cdot \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}_{\ell^\star}}(\lambda) \leq Q(L+1) \cdot \mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L,\mathcal{B}}(\lambda)$. □

If the underlying HIBE scheme is selectively secure, our HKIBE scheme then satisfies selective security. Similarly, if the underlying HIBE scheme is CCA-secure, our HKIBE scheme meets CCA security. We omit the proofs since they can be done in the same manner as Theorem 1.

Corollary 1. *If the underlying HIBE scheme with hierarchical depth $L+1$ supporting MSK evaluatability satisfies selective security, then the above HKIBE scheme with hierarchical depth L also satisfies selective security. Specifically, if there exists an adversary $\mathcal{A}$ to break selective security of our HKIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$, then there exists a reduction algorithm $\mathcal{B}$ to break selective security of the underlying HIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+1,\mathcal{B}}(\lambda) \geq \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)/\Theta(L)$.*

Corollary 2. *If the underlying HIBE scheme with hierarchical depth $L+1$ supporting MSK evaluatability satisfies (adaptive) CCA security, then the above HKIBE scheme with hierarchical depth L also satisfies (adaptive) CCA security. Specifically, if there exists an adversary $\mathcal{A}$ to break adaptive CCA security of our HKIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$, then there exists a reduction algorithm $\mathcal{B}$ to break adaptive CCA security of the underlying HIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+1,\mathcal{B}}(\lambda) \geq \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)/\Theta(QL)$.*

5 Generic Construction from Plain HIBE

We provide a generic construction of an HKIBE scheme with depth L from any plain HIBE scheme with depth $L+1$. This construction is based on Hanaoka et al.'s HKIBE scheme [HHSI05], and can be easily extended to an adaptive-identity CCA secure scheme (see Section 5.4). We suppose that the first $\lceil\log(L+1)\rceil$ bits of each identity of our HKIBE scheme is used for indicating the hierarchical level ℓ, and the rest expresses the identity (i.e., $\ell\|\mathtt{id} \in \mathcal{I}_{\text{HIBE}} \approx [0,L] \times \mathcal{I}$). For instance, if the bit-length of identities in the underlying HIBE scheme is 256 bits and $L = 250$ (i.e., $\lceil\log(L+1)\rceil = 8$), then the identity in our HKIBE scheme is 248 bits.

5.1 Construction Idea

The aim of our second construction is to get rid of MSK evaluatability from the first construction while keeping the security. The basic idea is *to employ an* $(L+1)$*-out-of-*$(L+1)$ *secret sharing scheme for plaintexts*, where as the first construction employs it for MSK-parts of helper keys. Specifically, this construction's core spirit is that a ciphertext $\mathsf{ct}_{\mathtt{id},\mathtt{t}}$ consists of $L+1$ HIBE ciphertexts $((\mathsf{C}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}))})_{\ell\in[L]}, \mathsf{C}_{0\|\mathtt{id}})$, where the first L ciphertexts are encryptions of uniformly random *pseudo-plaintexts* $(\mathsf{M}_\ell)_{\ell\in[L]}$ and the last ciphertext is an encryption of the plaintext masked with all pseudo-plaintexts $\mathsf{M} \bigoplus_{\ell\in[L]} \mathsf{M}_\ell$. The plaintext and all pseudo-plaintexts can be viewed as shares of an $(L+1)$-out-of-$(L+1)$ secret sharing scheme. We design the scheme so that each user $\mathtt{id}$ can decrypt a ciphertext $\mathsf{ct}_{\mathtt{id},\mathtt{t}}$ only if the user is able to decrypt all the $L+1$ HIBE ciphertexts. Indeed, the similar design concept is employed in the HHSI05 scheme [HHSI05]. However, our design requires only one HIBE scheme with the hierarchical depth $L+1$ while the HHSI05 scheme consists of $L+1$ HIBE schemes for the different depth $\ell \in [L+1]$. This design improvement makes master public/secret keys be constant sizes.

Level	Elements	Components				
		1st	2nd	$\cdots$	L-th	$(L+1)$-th
L	$\mathsf{hk}^{(L)}_{\mathtt{id},\mathsf{T}_L(\mathtt{t})}$	$\mathsf{SK}_{L\Vert\mathtt{id}}$	—	$\cdots$	—	—
		↓ *delegated from* $L\Vert\mathtt{id}$ *to* $(L\Vert\mathtt{id},\mathsf{T}_{L-1}(t))$				
$L-1$	$\mathsf{hk}^{(L-1)}_{\mathtt{id},\mathsf{T}_{L-1}(\mathtt{t})}$	$\mathsf{SK}_{(L\Vert\mathtt{id},\mathsf{T}_{L-1}(\mathtt{t}))}$	$\mathsf{SK}_{L-1\Vert\mathtt{id}}$	$\cdots$	—	—
		↓	↓ *delegated from* $L-1\Vert\mathtt{id}$ *to* $(L-1\Vert\mathtt{id},\mathsf{T}_{L-2}(t))$			
$\vdots$	$\vdots$	$\vdots$	$\vdots$			
		↓	↓	$\cdots$		
1	$\mathsf{hk}^{(1)}_{\mathtt{id},\mathsf{T}_1(\mathtt{t})}$	$\mathsf{SK}_{(L\Vert\mathtt{id},\mathsf{T}_{[L-1,1]}(\mathtt{t}))}$	$\mathsf{SK}_{(L-1\Vert\mathtt{id},\mathsf{T}_{[L-2,1]}(\mathtt{t}))}$	$\cdots$	$\mathsf{SK}_{1\Vert\mathtt{id}}$	—
		↓	↓	$\cdots$	↓	
0	$\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})}$	$\mathsf{SK}_{(L\Vert\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))}$	$\mathsf{SK}_{(L-1\Vert\mathtt{id},\mathsf{T}_{[L-2,0]}(\mathtt{t}))}$	$\cdots$	$\mathsf{SK}_{(1\Vert\mathtt{id},\mathsf{T}_0(\mathtt{t}))}$	$\mathsf{SK}_{0\Vert\mathtt{id}}$
CT	$\mathsf{ct}_{\mathtt{id},\mathtt{t}}$	$\mathsf{C}_{(L\Vert\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))}$	$\mathsf{C}_{(L-1\Vert\mathtt{id},\mathsf{T}_{[L-2,0]}(\mathtt{t}))}$	$\cdots$	$\mathsf{C}_{(1\Vert\mathtt{id},\mathsf{T}_0(\mathtt{t}))}$	$\mathsf{C}_{0\Vert\mathtt{id}}$
M	M	M_L	M_{L-1}	$\cdots$	M_1	$\mathsf{M}\bigoplus_{i=1}^{L}\mathsf{M}_i$

Figure 2: The intuition of our second construction.

We illustrate the overview in Figure 2. As mentioned above, we consider $L+1$ hierarchical identities for ciphertexts in the underlying HIBE scheme: $(L\|\mathtt{id},\mathsf{T}_{[L-1,0]}(\mathtt{t}))$, $(L-1\|\mathtt{id},\mathsf{T}_{[L-2,0]}(\mathtt{t}))$, $\ldots$, $(1\|\mathtt{id},\mathsf{T}_0(\mathtt{t}))$, and $0\|\mathtt{id}$. Obviously, each ciphertext $\mathsf{C}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}))}$ of $\mathsf{ct}_{\mathtt{id},\mathtt{t}}$ can be decrypted by $\mathsf{SK}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}))}$ of $\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})}$ for $\ell\in[0,L]$.[13] Each helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ includes $L-\ell$ HIBE secret keys $(\mathsf{SK}_{j\|\mathtt{id},\mathsf{T}_{[j-1,\ell]}(\mathtt{t})})_{j\in[\ell+1,L]}$, which are delegated from their upper-level helper keys $(\mathsf{SK}_{j\|\mathtt{id},\mathsf{T}_{[j-1,\ell+1]}(\mathtt{t})})_{j\in[\ell+1,L]}$, and a new HIBE secret key $\mathsf{SK}_{\ell\|\mathtt{id}}$. Since there exists a special level $\ell^\star$ such that no level-$\ell^\star$ helper keys are compromised, an adversary does not have $\mathsf{SK}_{\ell^\star\|\mathtt{id}}$. In addition to this, the adversary cannot obtain any secret keys which can derive $\mathsf{SK}_{(\ell^\star\|\mathtt{id},\mathsf{T}_{[\ell^\star,0]}(\mathtt{t}^\star))}$ due to the restriction in the security game, the adversary cannot decrypt the challenge ciphertext.

5.2 Construction

Our HKIBE scheme $\Pi=(\mathsf{Setup},\mathsf{Encrypt},\mathsf{GenHK},\mathsf{KeyUp},\mathsf{Upd},\mathsf{Decrypt})$ from a plain HIBE scheme $\Sigma=(\mathsf{Init},\mathsf{Enc},\mathsf{GenSK},\mathsf{Dec})$ is as follows.

- $\mathsf{Setup}(1^\lambda,L)\to(\mathsf{pp},\mathsf{mk})$: Run $(\mathsf{MPK},\mathsf{MSK})\leftarrow\mathsf{Init}(1^\lambda,L+1)$ and output $\mathsf{pp}:=\mathsf{MPK}$ and $\mathsf{mk}:=\mathsf{MSK}$.
- $\mathsf{Encrypt}(\mathsf{pp},\mathtt{id},\mathtt{t},\mathsf{M})\to\mathsf{ct}_{\mathtt{id},\mathtt{t}}$: Parse $\mathsf{pp}=\mathsf{MPK}$. Sample $\mathsf{M}_\ell\leftarrow_R\mathcal{M}$ for $\ell\in[L]$ and run
  · $\mathsf{C}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}))}\leftarrow\mathsf{Enc}(\mathsf{MPK},(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}),\mathsf{M}_\ell)$ for $\ell\in[0,L]$,

  then output $\mathsf{ct}_{\mathtt{id},\mathtt{t}}:=(\mathsf{C}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]})})_{\ell\in[0,L]}$.
- $\mathsf{GenHK}(\mathsf{pp},\mathsf{mk},\mathtt{id})\to(\mathsf{hk}^{(0)}_{\mathtt{id},0})_{\ell\in[0,L]}$: Parse $\mathsf{pp}=\mathsf{MPK}$ and $\mathsf{mk}=\mathsf{MSK}$. For $\ell\in[0,L]$, compute $\mathsf{hk}^{(\ell)}_{\mathtt{id},0}:=(\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(0))})_{i\in[\ell,L]}$ as follows: for $i\in[\ell,L]$, run
  · $\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,0]}(0))}\leftarrow\mathsf{GenSK}(\mathsf{MPK},\mathsf{SK}_{i\|\mathtt{id}},(i\|\mathtt{id},\mathsf{T}_{[i-1,0]}(0)))$.

[13] $(0\|\mathtt{id},\mathsf{T}_{[-1,0]}(\mathtt{t}))$ here means $0\|\mathtt{id}$. The rest of this paper follows from this notation for notational simplicity. Similarly, $(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,\ell]}(\mathtt{t}))$ means $\ell\|\mathtt{id}$ for any $\ell\in[L]$.

  Output $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}$.

- $\mathsf{KeyUp}(\mathsf{pp},\mathtt{t},\mathsf{hk}^{(\ell+1)}_{\mathtt{id},t_{\ell+1}}) \rightarrow \mathsf{ku}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ or $\perp$: Output $\perp$ if $t_{\ell+1} \neq \mathsf{T}_{\ell+1}(\mathtt{t})$. Otherwise, parse
  - ▷ $\mathsf{pp} = \mathsf{MPK}$,
  - ▷ $\mathsf{hk}^{(\ell+1)}_{\mathtt{id},t_{\ell+1}} = (\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell+1]}(\mathtt{t}))})_{i\in[\ell+1,L]}$.

  Run
  - · $\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(\mathtt{t}))} \leftarrow \mathsf{GenSK}(\mathsf{MPK},\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell+1]}(\mathtt{t}))},(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(\mathtt{t})))$ for $i \in [\ell+1, L]$

  and output $\mathsf{ku}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})} \coloneqq (\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(\mathtt{t}))})_{i\in[\ell+1,L]}$.
- $\mathsf{Upd}(\mathsf{pp},\mathsf{hk}^{(\ell)}_{\mathtt{id},\tau_\ell},\mathsf{ku}^{(\ell)}_{\mathtt{id},t_\ell}) \rightarrow \mathsf{hk}^{(\ell)}_{\mathtt{id},t_\ell}$: Suppose $\tau_\ell = \mathsf{T}_\ell(\mathtt{t})$ and $t_\ell = \mathsf{T}_\ell(\mathtt{t}')$. Parse
  - ▷ $\mathsf{hk}^{(\ell)}_{\mathtt{id},\tau_\ell} = (\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(\mathtt{t}))})_{i\in[\ell,L]}$,
  - ▷ $\mathsf{ku}^{(\ell)}_{\mathtt{id},t_\ell} = (\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(\mathtt{t}'))})_{i\in[\ell+1,L]}$.

  Output $\mathsf{hk}^{(\ell)}_{\mathtt{id},t_\ell} \coloneqq (\mathsf{SK}_{\ell\|\mathtt{id}},(\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(\mathtt{t}'))})_{i\in[\ell+1,L]}) = (\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(\mathtt{t}'))})_{i\in[\ell,L]}$.
- $\mathsf{Decrypt}(\mathsf{pp},\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})},\mathsf{ct}_{\mathtt{id},\mathtt{t}}) \rightarrow \mathsf{M}$: Parse
  - ▷ $\mathsf{pp} = \mathsf{MPK}$,
  - ▷ $\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})} = \mathsf{hk}^{(0)}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})} = (\mathsf{SK}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}))})_{\ell\in[0,L]}$,
  - ▷ $\mathsf{ct}_{\mathtt{id},\mathtt{t}} = (\mathsf{C}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}))})_{\ell\in[0,L]}$.

  For $\ell \in [L]$, run
  - · $\mathsf{M}_\ell \leftarrow \mathsf{Dec}(\mathsf{MPK},\mathsf{SK}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}))},\mathsf{C}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}))})$,

  Output
  - – $\mathsf{M} = \mathsf{Dec}(\mathsf{MPK},\mathsf{SK}_{0\|\mathtt{id}},\mathsf{C}_{0\|\mathtt{id}}) \bigoplus_{\ell\in[L]} \mathsf{M}_\ell$.

Correctness. Since ciphertexts and decryption keys of our HKIBE scheme consists of those the underlying HIBE scheme, the correctness of the HKIBE scheme readily follows from that of the underlying HIBE scheme.

5.3 Security

The security of the HKIBE scheme is reduced to from that of the underlying HIBE scheme.

Theorem 2. *If the underlying HIBE scheme with hierarchical depth $L+1$ satisfies the adaptive security, the above HKIBE scheme with hierarchical depth L also satisfies the adaptive security. Specifically, if there exists an adversary $\mathcal{A}$ to break the adaptive security of the above HKIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$, then there exists a reduction algorithm $\mathcal{B}$ to break the adaptive security of the underlying HIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+1,\mathcal{B}}(\lambda) \geq \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)/\Theta(L)$.*

Proof Overview. In the proof, we divide $\mathcal{A}$'s attack strategy into $L+1$ types and define $\mathcal{A}_{\ell^\star}$ for every $\ell^\star \in [0, L]$ as in the proof of Theorem 1 with $\Theta(L)$ reduction loss. We show the proof against $\mathcal{A}_{\ell^\star}$ for fixed $\ell^\star$.

We use $\mathcal{A}_{\ell^\star}$ as a building block and construct a reduction algorithm $\mathcal{B}_{\ell^\star}$ against the underlying (plain) HIBE scheme. The challenge ciphertext for $(\mathtt{id}^\star,\mathtt{t}^\star)$ includes $L+1$ HIBE ciphertexts $\mathsf{C}^\star_{(L\|\mathtt{id}^\star,\mathsf{T}_{[L-1,0]}(\mathtt{t}^\star))},\ldots,\mathsf{C}^\star_{0\|\mathtt{id}^\star}$, and one of them should be the HIBE challenge ciphertext to reduce to adaptive security of the underlying HIBE scheme. Since $\mathcal{A}_{\ell^\star}$ does not make any key-insulation queries for $(\mathtt{id}^\star,\mathtt{t},\ell^\star)$, $\mathcal{B}_{\ell^\star}$ submits

$$(\ell^\star\|\mathtt{id}^\star,\mathsf{T}_{[\ell^\star-1,0]}(\mathtt{t}^\star)),$$

as the HIBE challenge identity. Therefore, $\mathcal{B}_{\ell^\star}$ can answer all $\mathcal{A}_{\ell^\star}$'s key-insulation queries, say, $(\mathtt{id}, \mathtt{t}, \ell)$, by making HIBE secret-key reveal queries for the corresponding identities $((j\|\mathtt{id}, \mathsf{T}_{[j-1,\ell]}(\mathtt{t})))_{j\in[\ell,L]}$ as long as it holds

$$(j\|\mathtt{id}, \mathsf{T}_{[j-1,\ell]}(\mathtt{t})) \notin \mathsf{prefix}^+((\ell^\star\|\mathtt{id}^\star, \mathsf{T}_{[\ell^\star-1,0]}(\mathtt{t}^\star))) \text{ for all } j \in [\ell, L], \tag{2}$$

even without the knowledge of the challenge tuple $(\mathtt{id}^\star, \mathtt{t}^\star)$. Since $\mathcal{B}_{\ell^\star}$ can obtain all HIBE secret keys such that $(\ell, \mathtt{id}) \neq (\ell^\star, \mathtt{id}^\star)$ via HIBE secret-key generation queries, the condition (2) can be more specific:

$$(\ell^\star\|\mathtt{id}^\star, \mathsf{T}_{[\ell^\star-1,\ell]}(\mathtt{t})) \notin \mathsf{prefix}^+((\ell^\star\|\mathtt{id}^\star, \mathsf{T}_{[\ell^\star-1,0]}(\mathtt{t}^\star))). \tag{3}$$

Therefore, we should care only about the key-insulation query $(\mathtt{id}^\star, \mathtt{t}, \ell)$ that produces $(\ell^\star\|\mathtt{id}^\star, \mathsf{T}_{[\ell^\star-1,\ell]}(\mathtt{t}))$. In the construction, only level-ℓ helper keys $\mathsf{hk}^{(\ell)}_{\mathtt{id}^\star,\mathsf{T}_\ell(\mathtt{t})}$ for $\ell \in [0, \ell^\star]$ include HIBE secret keys for $(\ell^\star\|\mathtt{id}^\star, \mathsf{T}_{[\ell^\star-1,\ell]}(\mathtt{t}))$. All possible $\mathcal{A}_{\ell^\star}$'s queries that contradict the condition (3) are:

- the initial helper-key reveal query on $\mathtt{id}^\star$;
- the key-insulation query on $(\mathtt{id}^\star, \mathtt{t}, \ell^\star)$ for any $\mathtt{t} \in \mathcal{T}_{act}$;
- the key-insulation query on $(\mathtt{id}^\star, \mathtt{t}, \ell)$ for any $\mathtt{t} \in \mathcal{T}_{act}$ and any $\ell \in [0, \ell^\star - 1]$ such that $\mathsf{T}_{[\ell^\star-1,\ell]}(\mathtt{t}) = \mathsf{T}_{[\ell^\star-1,\ell]}(\mathtt{t}^\star)$.

However, all the above queries are not allowed in the security game (see Definition 4). Thus, the condition (3) always holds for all $\mathcal{A}_{\ell^\star}$'s queries.

The remaining challenge is how $\mathcal{B}_{\ell^\star}$ embeds the challenge plaintexts $(\mathsf{M}_0^\star, \mathsf{M}_1^\star)$ into HIBE challenge ciphertext on $(\ell^\star\|\mathtt{id}^\star, \mathsf{T}_{[\ell^\star-1,0]}(\mathtt{t}^\star))$. Roughly speaking, we look at the pseudo-plaintexts of the challenge ciphertext differently: In the construction, for every $\ell \in [L]$, the HIBE ciphertext on $(\ell\|\mathtt{id}^\star, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star))$ is an encryption of a level-ℓ pseudo-plaintext $\mathsf{M}_\ell \leftarrow_R \mathcal{M}$ and that on $0\|\mathtt{id}^\star$ is an encryption of $\mathsf{M} \bigoplus_{\ell\in[L]} \mathsf{M}_\ell$. In the reduction $\mathcal{B}_{\ell^\star}$ sets each plaintext as follows:

- For $\ell \in [0, L] \setminus \{\ell^\star\}$, $\mathcal{B}_{\ell^\star}$ randomly chooses a level-ℓ pseudo-plaintext $\widehat{\mathsf{M}}_\ell$ and sets an HIBE ciphertext on $(\ell\|\mathtt{id}^\star, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star))$ is an encryption of $\widehat{\mathsf{M}}_\ell$. Similarly, $\mathcal{B}_{\ell^\star}$ randomly chooses a level-0 pseudo-plaintext $\widehat{\mathsf{M}}_0$ and sets an HIBE ciphertext on $0\|\mathtt{id}^\star$ is an encryption of $\widehat{\mathsf{M}}_0$.
- $\mathcal{B}_{\ell^\star}$ (implicitly) sets HIBE challenge ciphertext on $(\ell^\star\|\mathtt{id}^\star, \mathsf{T}_{[\ell^\star-1,0]}(\mathtt{t}^\star))$ is an encryption of $\mathsf{M}_b^\star \bigoplus_{\ell\in[0,L]\setminus\{\ell^\star\}} \widehat{\mathsf{M}}_\ell$.

All the above pseudo-plaintexts are properly distributed.

Theorem 2. We formally describe the proof as follows. Let $\mathcal{B}_{\ell^\star}$ be a reduction algorithm against the underlying HIBE scheme. We show how to construct $\mathcal{B}_{\ell^\star}$ by using $\mathcal{A}_{\ell^\star}$ as follows. At first, $\mathcal{B}_{\ell^\star}$ is given an HIBE's master public key MPK from an HIBE challenger $\mathcal{C}$. Then, $\mathcal{B}_{\ell^\star}$ initializes $\mathtt{HKList} = \emptyset$ and sends $\mathsf{pp} := \mathsf{MPK}$ to an HKIBE adversary $\mathcal{A}_{\ell^\star}$.

$\mathcal{B}_{\ell^\star}$ answers $\mathcal{A}_{\ell^\star}$'s queries by interacting with $\mathcal{C}$ as follows:

Helper-Key Generation Query: Upon a query $\mathtt{id}$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ checks if $(\mathtt{id}, \cdot) \notin \mathtt{HKList}$, and returns $\perp$ to $\mathcal{A}_{\ell^\star}$ if this is *not* the case. Otherwise, $\mathcal{B}_{\ell^\star}$ makes an HIBE secret-key query on $\ell\|\mathtt{id}$ for $\ell \in [0, L]\setminus\{\ell^\star\}$ to $\mathcal{C}$, receives $(\mathsf{SK}_{\ell\|\mathtt{id}})_{\ell\in[0,L]\setminus\{\ell^\star\}}$, and stores *a part of* an initial helper key $(\mathtt{id}, (\mathsf{SK}_{\ell\|\mathtt{id}})_{\ell\in[0,L]\setminus\{\ell^\star\}})$ in $\mathtt{HKList}$. Clearly, the part of the helper keys $(\mathsf{SK}_{\ell\|\mathtt{id}})_{\ell\in[0,L]\setminus\{\ell^\star\}}$ are created in the same way as the construction.

Initial Helper-Key Reveal Queries: Upon a query $\mathtt{id}$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ finds $(\mathsf{SK}_{\ell\|\mathtt{id}})_{\ell\in[0,L]\setminus\{\ell^\star\}}$ from $\mathtt{HKList}$. $\mathcal{B}_{\ell^\star}$ retrieves $\mathsf{SK}_{\ell^\star\|\mathtt{id}}$ if $\mathtt{HKList}$ also contains it.[14] If not, $\mathcal{B}_{\ell^\star}$ makes an HIBE secret-key reveal query on $\ell^\star\|\mathtt{id}$ to get $\mathsf{SK}_{\ell^\star\|\mathtt{id}}$ and stores it in $\mathtt{HKList}$ together with $(\mathsf{SK}_{\ell\|\mathtt{id}})_{\ell\in[0,L]\setminus\{\ell^\star\}}$. Then, $\mathcal{B}_{\ell^\star}$ creates $(\mathsf{hk}^{(\ell)}_{\mathtt{id},0})_{\ell\in[0,L]}$ by using $(\mathsf{SK}_{\ell\|\mathtt{id}})_{\ell\in[0,L]}$ as in the construction, and returns it to $\mathcal{A}_{\ell^\star}$. It is obvious that $\mathsf{hk}^{(\ell^\star)}_{\mathtt{id},0}$ is created in the same way as the construction.

Key-Insulation Query: Upon a query $(\mathtt{id},\mathtt{t},\ell)$, $\mathcal{B}_{\ell^\star}$ makes all HIBE secret-key reveal queries for $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$, i.e., queries on $((i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(\mathtt{t})))_{i\in[\ell,L]}$ to get $(\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(\mathtt{t}))})_{i\in[\ell,L]}$. Finally, $\mathcal{B}_{\ell^\star}$ returns $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})} := (\mathsf{SK}_{(i\|\mathtt{id},\mathsf{T}_{[i-1,\ell]}(\mathtt{t}))})_{i\in[\ell,L]}$ to $\mathcal{A}_{\ell^\star}$. It is obvious that the helper key is properly distributed thanks to the *correctness* of the underlying HIBE scheme.

Challenge Query: Upon a query $(\mathtt{id}^\star,\mathtt{t}^\star,\mathsf{M}^\star_0,\mathsf{M}^\star_1)$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ samples $\widehat{\mathsf{M}}_\ell \leftarrow_R \mathcal{M}$ for $\ell\in[0,L]\setminus\{\ell^\star\}$ and makes an HIBE challenge query on $((\ell^\star\|\mathtt{id}^\star,\mathsf{T}_{[\ell^\star-1,0]}(\mathtt{t}^\star)),$ $\mathsf{M}^\star_0\bigoplus_{\ell\in[0,L]\setminus\{\ell^\star\}}\widehat{\mathsf{M}}_\ell,\mathsf{M}^\star_1\bigoplus_{\ell\in[0,L]\setminus\{\ell^\star\}}\widehat{\mathsf{M}}_\ell)$, and receives an HIBE challenge ciphertext $\mathsf{C}^\star_{(\ell^\star\|\mathtt{id}^\star,\mathsf{T}_{[\ell^\star-1,0]}(\mathtt{t}^\star))}$. Then, $\mathcal{B}_{\ell^\star}$ runs

– $\mathsf{C}^\star_{(\ell\|\mathtt{id}^\star,\mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star))}\leftarrow\mathsf{Enc}(\mathsf{MPK},(\ell\|\mathtt{id}^\star,\mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star)),\widehat{\mathsf{M}}_\ell)$ for $\ell\in[0,L]\setminus\{\ell^\star\}$,

and returns $(\mathsf{C}^\star_{(\ell\|\mathtt{id}^\star,\mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star))})_{\ell\in[0,L]}$ to $\mathcal{A}_{\ell^\star}$ as an HKIBE challenge ciphertext.

Observe that the challenge ciphertext is properly distributed by implicitly setting

– $\mathsf{M}_\ell=\widehat{\mathsf{M}}_\ell$ for $\ell\in[0,L]\setminus\{\ell^\star\}$,

– $\mathsf{M}_{\ell^\star}=\mathsf{M}^\star_b\bigoplus_{\ell\in[0,L]\setminus\{\ell^\star\}}\widehat{\mathsf{M}}_\ell$,

where $b\leftarrow_R\{0,1\}$. Level-ℓ pseudo-plaintext $\mathsf{M}_\ell=\widehat{\mathsf{M}}_\ell$ for $\ell\in[L]\setminus\{\ell^\star\}$ is created in the same way as the construction. Since the level-0 pseudo-plaintext $\widehat{\mathsf{M}}_0$ is uniformly distributed over the message space of the underlying HIBE scheme, the distribution of $\mathsf{M}_{\ell^\star}$ is independent of $\mathsf{M}^\star_b$ and $(\widehat{\mathsf{M}}_\ell)_{\ell\in[L]\setminus\{\ell^\star\}}$, and uniformly random in the HIBE message space as in the construction. Therefore, the distribution of $(\mathsf{C}^\star_{(\ell\|\mathtt{id}^\star,\mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star))})_{\ell\in[L]}$is identical to that in the construction. $\mathsf{C}^\star_{0\|\mathtt{id}^\star}$ is an encryption of $\mathsf{M}^\star_b\bigoplus_{\ell\in[L]}\mathsf{M}_\ell$ in the construction while it is an encryption of $\widehat{\mathsf{M}}_0$ in the reduction. Since

$$\mathsf{M}^\star_b\bigoplus_{\ell\in[L]}\mathsf{M}_\ell=\mathsf{M}^\star_b\oplus\mathsf{M}_{\ell^\star}\bigoplus_{\ell\in[L]\setminus\{\ell^\star\}}\mathsf{M}_\ell=\mathsf{M}^\star_b\oplus\mathsf{M}^\star_b\bigoplus_{\ell\in[0,L]\setminus\{\ell^\star\}}\widehat{\mathsf{M}}_\ell\bigoplus_{\ell\in[L]\setminus\{\ell^\star\}}\widehat{\mathsf{M}}_\ell=\widehat{\mathsf{M}}_0$$

holds, the distribution of $\mathsf{C}^\star_{0\|\mathtt{id}^\star}$ is the same as that of the construction.

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

The above completes the description of $\mathcal{B}_{\ell^\star}$. Observe that $\mathcal{B}_{\ell^\star}$ can make all $\mathcal{A}$'s queries with $\mathcal{C}$. $\mathcal{B}_{\ell^\star}$ makes the HIBE challenge query on

$$(\ell^\star\|\mathtt{id}^\star,\mathsf{T}_{[\ell^\star-1,0]}(\mathtt{t}^\star)),$$

while $\mathcal{B}_{\ell^\star}$ can make HIBE secret-key reveal queries in any case for the following reasons.

- The initial helper-key reveal query on $\mathtt{id}$.

  Case for $\mathtt{id}\neq\mathtt{id}^\star$: It is obvious that $\mathcal{B}_{\ell^\star}$ can make HIBE secret-key reveal queries on $\ell\|\mathtt{id}$ for every $\ell\in[0,L]$.

[14] It depends on whether $\mathtt{id}$ has been used for key-insulation query.

Case for $\mathtt{id} = \mathtt{id}^\star$: This query is not allowed in the game.

- The key-insulation query on $(\mathtt{id}, \mathtt{t}, \ell)$.

  Case for $\mathtt{id} \neq \mathtt{id}^\star$: $\mathcal{B}_{\ell^\star}$ can make HIBE secret-key reveal queries to return $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$.

  Case for $\mathtt{id} = \mathtt{id}^\star$: We take look at the following three cases.

  Case for $\ell \in [\ell^\star + 1, L]$: In this case, $\mathsf{hk}^{(\ell)}_{\mathtt{id}^\star,\mathsf{T}_\ell(\mathtt{t})}$ does not include any HIBE secret key $\mathsf{SK}_{(\ell^\star \| \mathtt{id}^\star, \mathsf{T}_{[\ell^\star-1,\ell]}(\mathtt{t}))}$, which violates the condition (3), by the construction. Therefore, $\mathcal{B}_{\ell^\star}$ can make HIBE secret-key reveal queries on $\ell \| \mathtt{id}$ and $(i \| \mathtt{id}, \mathsf{T}_{[i-1,\ell]}(\mathtt{t}))$ for every $i \in [\ell + 1, L]$.

  Case for $\ell = \ell^\star$: This case never occurs due to the restriction on $\ell^\star$.

  Case for $\ell \in [0, \ell^\star - 1]$: Since it always holds $\mathsf{T}_\ell(\mathtt{t}) \neq \mathsf{T}_\ell(\mathtt{t}^\star)$ due to the restriction on $\ell^\star$, and it means that such a query always meets the condition (3). $\mathcal{B}_{\ell^\star}$ can make HIBE secret-key reveal queries on $\ell \| \mathtt{id}$ and $(i \| \mathtt{id}, \mathsf{T}_{[i-1,\ell]}(\mathtt{t}))$ for every $i \in [\ell+1, L]$.

As we already observed, $\mathcal{B}_{\ell^\star}$ perfectly simulates the adaptive security game against $\mathcal{A}_{\ell^\star}$. Since the probability that β' is a correct guess is the same as that of b', $\mathcal{B}_{\ell^\star}$'s advantage is $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}_{\ell^\star}}(\lambda) = \mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+1,\mathcal{B}_{\ell^\star}}(\lambda)$. Therefore, $\mathcal{B}$'s advantage against $\mathcal{A}$ of general attack strategy is $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda) = \sum_{\ell^\star \in [0,L]} \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}_{\ell^\star}}(\lambda) = (L+1) \cdot \mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+1,\mathcal{B}}(\lambda)$. □

As in the first construction, if the underlying HIBE scheme is selectively secure, our HKIBE scheme then satisfies selective security. We omit the proof since it can be done in the same manner as Theorem 2.

Corollary 3. *If the underlying HIBE scheme with hierarchical depth $L + 1$ satisfies selective security, then the above HKIBE scheme with hierarchical depth L also satisfies selective security. Specifically, if there exists an adversary $\mathcal{A}$ to break selective security of our HKIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$, then there exists a reduction algorithm $\mathcal{B}$ to break selective security of the underlying HIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+1,\mathcal{B}}(\lambda) \geq \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)/\Theta(L)$.*

5.4 Achieving CCA Security

Unlike the first construction, we cannot obtain a CCA-secure construction by just replacing the underlying CPA-secure HIBE scheme with CCA-secure ones. The reason is that the second construction require $L + 1$ HIBE ciphertexts for each HKIBE ciphertext, while the ciphertext of the first construction consists of only one HIBE ciphertext. In other words, there is the following trivial attack: an adversary $\mathcal{A}$ replaces $\mathsf{C}^\star_{0\|\mathtt{id}^\star}$ of the challenge ciphertext $\mathsf{ct}^\star_{\mathtt{id}^\star,\mathtt{t}^\star}$ with $\mathsf{Enc}(\mathsf{MPK}, 0\|\mathtt{id}^\star, 0^{|\mathsf{M}^\star_0|})$ (the modified challenge ciphertext is denoted by $\mathsf{ct}'_{\mathtt{id}^\star,\mathtt{t}^\star}$), makes a decryption query on $(\mathtt{id}^\star, \mathtt{t}^\star, \mathsf{ct}'_{\mathtt{id}^\star,\mathtt{t}^\star})$, and receives $\bigoplus_{\ell\in[L]} \mathsf{M}_\ell$. Similarly, $\mathcal{A}$ replaces $\mathsf{C}^\star_{(\ell\|\mathtt{id}^\star,\mathsf{T}_{[\ell-1,0]})}$ of $\mathsf{ct}^\star_{\mathtt{id}^\star,\mathtt{t}^\star}$ with $\mathsf{Enc}(\mathsf{MPK}, (\ell\|\mathtt{id}^\star, \mathsf{T}_{[\ell-1,0]}), 0^{|\mathsf{M}^\star_0|})$ (the modified challenge ciphertext is denoted by $\mathsf{ct}''_{\mathtt{id}^\star,\mathtt{t}^\star}$), makes a decryption query on $(\mathtt{id}^\star, \mathtt{t}^\star, \mathsf{ct}''_{\mathtt{id}^\star,\mathtt{t}^\star})$, and receives $\mathsf{M}^\star_b \bigoplus_{\ell\in[L]} \mathsf{M}_\ell$. Therefore, $\mathcal{A}$ can get $\mathsf{M}^\star_b$ and win the game with probability one.

As in [HHSI05], we adopt the well-known multiple encryption approach [DK05] to achieve CCA security.

One-Time Signature (OTS). An OTS scheme Γ consists of three algorithms $(\mathsf{SSetup}, \mathsf{Sign}, \mathsf{Vrfy})$.

- $\mathsf{SSetup}(1^\lambda) \to (\mathsf{sigk}, \mathsf{verk})$: given the security parameter λ, it outputs a key pair $(\mathsf{sigk}, \mathsf{verk})$.
- $\mathsf{Sign}(\mathsf{sigk}, \mathsf{M}) \to \sigma$: given the signing key sigk and a message M, it outputs a signature σ.

- $\mathsf{Vrfy}(\mathsf{verk}, \mathsf{M}, \sigma) \to \top$ or $\bot$: given the verification key verk, a message M, and its signature σ, it outputs $\top$, which indicates "acceptance", or $\bot$, which indicates "rejection".

We require that for all security parameters λ, $(\mathsf{sigk}, \mathsf{verk}) \leftarrow \mathsf{SSetup}(1^\lambda)$, and messages M, it holds $\mathsf{Vrfy}(\mathsf{verk}, \mathsf{M}, \mathsf{Sign}(\mathsf{sigk}, \mathsf{M})) = \top$ with overwhelming probability.

We define a security notion for OTS. Let Γ be an OTS scheme, and we consider a game between an adversary $\mathcal{F}$ and the challenger $\mathcal{C}$. The game is parameterized by the security parameter λ. The game proceeds as follows: $\mathcal{C}$ first runs $(\mathsf{sigk}, \mathsf{verk}) \leftarrow \mathsf{SSetup}(1^\lambda)$ and gives verk to $\mathcal{A}$. $\mathcal{F}$ is allowed to make the *signature generation query* only once: upon a query M from $\mathcal{F}$, $\mathcal{C}$ returns $\sigma \leftarrow \mathsf{Sign}(\mathsf{sigk}, \mathsf{M})$ to $\mathcal{A}$. $\mathcal{F}$ outputs $(\mathsf{M}^\star, \sigma^\star)$ and terminates. In this game, $\mathcal{F}$'s adaptive security advantage is defined by $\mathsf{Adv}^{\mathtt{OTS}}_{\Gamma,\mathcal{F}}(\lambda) := \Pr[\mathsf{Vrfy}(\mathsf{verk}, \mathsf{M}^\star, \sigma^\star) \to \top \wedge (\mathsf{M}^\star, \sigma^\star) \neq (\mathsf{M}, \sigma)]$.

Definition 5 (Strong Unforgeability)**.** *We say that an OTS scheme* Γ *satisfies* strong unforgeability*, if the advantage* $\mathsf{Adv}^{\mathtt{OTS}}_{\Gamma,\mathcal{F}}(\lambda)$ *is negligible for all PPT adversaries* $\mathcal{F}$*.*

Construction. First of all, we change the maximum hierarchy depth $L+1$ of the underlying HIBE scheme to $L+2$. We then modify the $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$ algorithms of the second construction as follows.

- $\mathsf{Encrypt}(\mathsf{pp}, \mathtt{id}, \mathtt{t}, \mathsf{M}) \to \mathsf{ct}_{\mathtt{id},\mathtt{t}}$: Parse $\mathsf{pp} = \mathsf{MPK}$. Generate $(\mathsf{sigk}, \mathsf{verk}) \leftarrow \mathsf{SSetup}(1^\lambda)$. Sample $\mathsf{M}_\ell \leftarrow_R \mathcal{M}$ for $\ell \in [L]$ and run
  - · $\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk})} \leftarrow \mathsf{Enc}(\mathsf{MPK}, (\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}, \mathsf{verk}), \mathsf{M}_\ell)$ for $\ell \in [L]$,
  - · $\mathsf{C}_{(0 \| \mathtt{id}, \mathsf{verk})} \leftarrow \mathsf{Enc}(\mathsf{MPK}, (0 \| \mathtt{id}, \mathsf{verk}), \mathsf{M} \bigoplus_{\ell \in [L]} \mathsf{M}_\ell)$,
  - · $\sigma \leftarrow \mathsf{Sign}(\mathsf{sigk}, (\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}, \mathsf{verk})})_{\ell \in [0,L]})$,

  then output $\mathsf{ct}_{\mathtt{id},\mathtt{t}} := ((\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}, \mathsf{verk})})_{\ell \in [0,L]}, \sigma, \mathsf{verk})$. Here, $(0 \| \mathtt{id}, \mathsf{T}_{[-1,0]}, \mathsf{verk})$ means $(0 \| \mathtt{id}, \mathsf{verk})$.
- $\mathsf{Decrypt}(\mathsf{pp}, \mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})}, \mathsf{ct}_{\mathtt{id},\mathtt{t}}) \to \mathsf{M}$: Parse
  - ▷ $\mathsf{pp} = \mathsf{MPK}$,
  - ▷ $\mathsf{dk}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})} = \mathsf{hk}^{(0)}_{\mathtt{id},\mathsf{T}_0(\mathtt{t})} = ((\mathsf{SK}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}))})_{\ell \in [0,L]})$,
  - ▷ $\mathsf{ct}_{\mathtt{id},\mathtt{t}} = ((\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}, \mathsf{verk})})_{\ell \in [0,L]}, \sigma, \mathsf{verk})$.

  Compute $\mathsf{Vrfy}(\mathsf{verk}, (\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}, \mathsf{verk})})_{\ell \in [0,L]}, \sigma)$. If the output is $\bot$, then output $\bot$. Otherwise, for $\ell \in [L]$, run
  - · $\mathsf{SK}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk})} \leftarrow \mathsf{GenSK}(\mathsf{MPK}, \mathsf{SK}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}))}, (\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk}))$,
  - · $\mathsf{M}_\ell \leftarrow \mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk})}, \mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk})})$,

  Compute $\mathsf{SK}_{(0 \| \mathtt{id}, \mathsf{verk})} \leftarrow \mathsf{GenSK}(\mathsf{MPK}, \mathsf{SK}_{0 \| \mathtt{id}}, (0 \| \mathtt{id}, \mathsf{verk}))$. Output
  - · $\mathsf{M} = \mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{(0 \| \mathtt{id}, \mathsf{verk})}, \mathsf{C}_{(0 \| \mathtt{id}, \mathsf{verk})}) \bigoplus_{\ell \in [L]} \mathsf{M}_\ell$.

Rest of the algorithms are the same as those of the CPA-secure construction.

Theorem 3. *If the underlying HIBE scheme with hierarchical depth* $L+2$ *satisfies (adaptive-identity) CCA security and the underlying OTS scheme satisfies strong unforgeability, then the above HKIBE scheme with hierarchical depth* L *also satisfies (adaptive-identity) CCA security. Specifically, if there exists an adversary* $\mathcal{A}$ *to break adaptive-identity CCA security of our HKIBE scheme with advantage* $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$*, then there exists a reduction algorithm* $\mathcal{B}$ *to break adaptive-identity CCA security of the underlying HIBE scheme with advantage* $\mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+2,\mathcal{B}}(\lambda) \geq \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)/\Theta(L)$ *or a reduction algorithm* $\mathcal{F}$ *to break strong unforgeability of the underlying OTS scheme with advantage* $\mathsf{Adv}^{\mathtt{OTS}}_{\Gamma,\mathcal{F}}(\lambda) \geq \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$*.*

Proof. First of all, we consider two types of adversaries $\mathcal{A}$:

- ▷ $\mathcal{A}$ makes at least one decryption query $(\mathtt{id}, \mathtt{t}, \mathsf{ct}_{\mathtt{id},\mathtt{t}})$ that includes valid $\mathsf{verk}^\star$, which is a *challenge verification key* generated at the beginning of the game.[15] Here, "valid" $\mathsf{verk}^\star$ means $\mathsf{verk}^\star$ such that it holds $\mathsf{Vrfy}(\mathsf{verk}^\star, (\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk}^\star)})_{\ell \in [0,L]}, \sigma) = \top$, where $\mathsf{ct}_{\mathtt{id},\mathtt{t}} = ((\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk}^\star)})_{\ell \in [0,L]}, \mathsf{verk}^\star, \sigma)$ is a ciphertext of the decryption query.
- ▷ $\mathcal{A}$ does not make any decryption query $(\mathtt{id}, \mathtt{t}, \mathsf{ct}_{\mathtt{id},\mathtt{t}})$ that includes valid $\mathsf{verk}^\star$.

If $\mathcal{A}$ is the former type, we can construct a reduction algorithm $\mathcal{F}$ against the underlying OTS scheme. Otherwise, i.e., if $\mathcal{A}$ is the latter type, we can construct a reduction algorithm $\mathcal{B}$ against the underlying HIBE scheme. The rest of the proof follows from the following Lemmas 1 and 2.

Lemma 1. *If there exists an adversary $\mathcal{A}$ to break adaptive-identity CCA security of our HKIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$ and $\mathcal{A}$ makes at least one decryption query that includes valid $\mathsf{verk}^\star$, then there exists a reduction algorithm $\mathcal{F}$ to break strong unforgeability of the underlying OTS scheme with advantage $\mathsf{Adv}^{\mathtt{OTS}}_{\Gamma,\mathcal{F}}(\lambda) \geq \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$.*

Proof. At first, $\mathcal{F}$ is given a challenge verification key $\mathsf{verk}^\star$ from an OTS challenger $\mathcal{C}$, and computes $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Init}(1^\lambda, L+2)$. Then, $\mathcal{F}$ initializes $\mathtt{HKList} = \emptyset$ and sends $\mathsf{pp} := \mathsf{MPK}$ to an HKIBE adversary $\mathcal{A}$. $\mathcal{F}$ can answer all helper-key generation queries, initial helper-key reveal queries, key-insulation queries, and decryption queries since $\mathcal{F}$ has MSK. We here explicitly describe the challenge query.

Challenge Query: Upon a query $(\mathtt{id}^\star, \mathtt{t}^\star, \mathsf{M}_0^\star, \mathsf{M}_1^\star)$ from $\mathcal{A}$, $\mathcal{F}$ samples $b \leftarrow_R \{0,1\}$ and $\widehat{\mathsf{M}}_\ell \leftarrow_R \mathcal{M}$ for $\ell \in [L]$ and runs

- $\mathsf{C}^\star_{(\ell \| \mathtt{id}^\star, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star), \mathsf{verk}^\star)} \leftarrow \mathsf{Enc}(\mathsf{MPK}, (\ell \| \mathtt{id}^\star, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star), \mathsf{verk}^\star), \widehat{\mathsf{M}}_\ell)$ for $\ell \in [L]$,
- $\mathsf{C}^\star_{(0 \| \mathtt{id}^\star, \mathsf{verk}^\star)} \leftarrow \mathsf{Enc}(\mathsf{MPK}, (0 \| \mathtt{id}^\star, \mathsf{verk}^\star), \mathsf{M}_b \bigoplus_{\ell \in [L]} \widehat{\mathsf{M}}_\ell)$.

$\mathcal{F}$ then makes a signature generation query $(\mathsf{C}^\star_{(\ell \| \mathtt{id}^\star, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star), \mathsf{verk}^\star)})_{\ell \in [0,L]}$ and receives $\sigma^\star$. $\mathcal{F}$ returns $((\mathsf{C}^\star_{(\ell \| \mathtt{id}^\star, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star), \mathsf{verk}^\star)})_{\ell \in [0,L]}, \mathsf{verk}^\star, \sigma^\star)$ to $\mathcal{A}$.

At some point, $\mathcal{A}$ makes a decryption query $(\mathtt{id}, \mathtt{t}, \mathsf{ct}_{\mathtt{id},\mathtt{t}})$ such that

$$\mathsf{Vrfy}(\mathsf{verk}^\star, (\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk}^\star)})_{\ell \in [0,L]}, \sigma) = \top,$$

where $\mathsf{ct}_{\mathtt{id},\mathtt{t}} = ((\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk}^\star)})_{\ell \in [0,L]}, \mathsf{verk}^\star, \sigma)$. $\mathcal{F}$ then outputs $((\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk}^\star)})_{\ell \in [0,L]}, \sigma)$ as a forgery and terminates the game. Due to the restriction on decryption query, it holds

$$((\mathsf{C}_{(\ell \| \mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk}^\star)})_{\ell \in [0,L]}, \mathsf{verk}^\star, \sigma) \neq ((\mathsf{C}^\star_{(\ell \| \mathtt{id}^\star, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star), \mathsf{verk}^\star)})_{\ell \in [0,L]}, \mathsf{verk}^\star, \sigma^\star).$$

Hence, $\mathcal{F}$ breaks strong unforgeability of the underlying OTS scheme, and we have $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathtt{OTS}}_{\Gamma,\mathcal{F}}(\lambda)$ if $\mathcal{A}$ queries at least one decryption query that includes valid $\mathsf{verk}^\star$. □

Lemma 2. *If there exists an adversary $\mathcal{A}$ to break adaptive-identity CCA security of our HKIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)$ and $\mathcal{A}$ does not make any decryption query that includes valid $\mathsf{verk}^\star$, then there exists a reduction algorithm $\mathcal{B}$ to break adaptive-identity CCA security of the underlying HIBE scheme with advantage $\mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+2,\mathcal{B}}(\lambda) \geq \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda)/\Theta(L)$.*

[15] To be precise, it should be generated at the challenge phase. However, the original security game and the game where $\mathsf{verk}^\star$ is generated at the beginning of the game are identical from the viewpoint of $\mathcal{A}_{\ell^\star}$. Therefore, we here consider the latter game.

Proof. We can prove this theorem in the same manner as Theorem 2. We divide $\mathcal{A}$'s attack strategy into $L+1$ types (with $\Theta(L)$ reduction loss), and show the proof against $\mathcal{A}$ of the Type-$\ell^\star$ strategy (denoted by $\mathcal{A}_{\ell^\star}$) for fixed $\ell^\star$. Let $\mathcal{B}_{\ell^\star}$ be a reduction algorithm against the underlying HIBE scheme. We show how to construct $\mathcal{B}_{\ell^\star}$ by using $\mathcal{A}_{\ell^\star}$ that does not make any decryption query that includes valid $\mathsf{verk}^\star$.

At first, $\mathcal{B}_{\ell^\star}$ is given an HIBE's master public key MPK from an HIBE challenger $\mathcal{C}$, and computes $(\mathsf{sigk}^\star, \mathsf{verk}^\star) \leftarrow \mathsf{SSetup}(1^\lambda)$. Then, $\mathcal{B}_{\ell^\star}$ initializes $\mathtt{HKList} = \emptyset$ and sends $\mathsf{pp} := \mathsf{MPK}$ to an HKIBE adversary $\mathcal{A}_{\ell^\star}$. $\mathcal{B}_{\ell^\star}$ can answer all queries except for decryption and challenge queries in the same way as in the proof of Theorem 2. Therefore, we here describe how to answer decryption queries (that does not include valid $\mathsf{verk}^\star$) and challenge query.

Decryption Query: Upon a query $(\mathtt{id}, \mathtt{t}, \mathsf{ct}_{\mathtt{id},\mathtt{t}})$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ checks the following two conditions:

- $\triangleright$ $(\mathtt{id}, \cdot) \notin \mathtt{HKList}$,
- $\triangleright$ $\mathsf{Vrfy}(\mathsf{verk}, (\mathsf{C}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}),\mathsf{verk})})_{\ell\in[0,L]}, \sigma) = \bot$,

where $\mathsf{ct}_{\mathtt{id},\mathtt{t}} = ((\mathsf{C}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}),\mathsf{verk})})_{\ell\in[0,L]}, \mathsf{verk}, \sigma)$. If at least one condition holds, $\mathcal{B}_{\ell^\star}$ returns $\bot$. Otherwise, $\mathcal{B}_{\ell^\star}$ makes HIBE secret key queries on $(0\|\mathtt{id}, \mathsf{verk})$ and $(\ell\|\mathtt{id}, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}), \mathsf{verk})$ for $\ell \in [L]$, and obtains $\mathsf{SK}_{(0\|\mathtt{id},\mathsf{verk})}$ and $\mathsf{SK}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}),\mathsf{verk})}$ for $\ell \in [L]$. From the assumption that $\mathcal{A}_{\ell^\star}$ never makes any decryption queries that include valid $\mathsf{verk}^\star$, $\mathsf{verk} \neq \mathsf{verk}^\star$ holds. Therefore, $\mathcal{B}_{\ell^\star}$ can get the corresponding HIBE secret keys even if $(\mathtt{id}, \mathtt{t}) = (\mathtt{id}^\star, \mathtt{t}^\star)$ for any decryption queries. $\mathcal{B}_{\ell^\star}$ then runs

- $\cdot$ $\mathsf{M}_\ell \leftarrow \mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}),\mathsf{verk})}, \mathsf{C}_{(\ell\|\mathtt{id},\mathsf{T}_{[\ell-1,0]}(\mathtt{t}),\mathsf{verk})})$ for $\ell \in [L]$,
- $\cdot$ $\mathsf{M} := \mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{(0\|\mathtt{id},\mathsf{verk})}, \mathsf{C}_{(0\|\mathtt{id},\mathsf{verk})}) \bigoplus_{\ell\in[L]} \mathsf{M}_\ell$.

$\mathcal{B}_{\ell^\star}$ returns M to $\mathcal{A}_{\ell^\star}$.

Challenge Query: Upon a query $(\mathtt{id}^\star, \mathtt{t}^\star, \mathsf{M}_0^\star, \mathsf{M}_1^\star)$ from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ samples $\widehat{\mathsf{M}}_\ell \leftarrow_R \mathcal{M}$ for $\ell \in [0, L] \setminus \{\ell^\star\}$ and makes an HIBE challenge query on $((\ell^\star\|\mathtt{id}^\star, \mathsf{T}_{[\ell^\star-1,0]}(\mathtt{t}^\star), \mathsf{verk}^\star), \mathsf{M}_0^\star \bigoplus_{\ell\in[0,L]\setminus\{\ell^\star\}} \widehat{\mathsf{M}}_\ell, \mathsf{M}_1^\star \bigoplus_{\ell\in[0,L]\setminus\{\ell^\star\}} \widehat{\mathsf{M}}_\ell)$, and receives an HIBE challenge ciphertext $\mathsf{C}_{(\ell^\star\|\mathtt{id}^\star,\mathsf{T}_{[\ell^\star-1,0]}(\mathtt{t}^\star),\mathsf{verk}^\star)}$. Then, $\mathcal{B}_{\ell^\star}$ runs

- – $\mathsf{C}_{(\ell\|\mathtt{id}^\star,\mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star),\mathsf{verk}^\star)} \leftarrow \mathsf{Enc}(\mathsf{MPK}, (\ell\|\mathtt{id}^\star, \mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star), \mathsf{verk}^\star), \widehat{\mathsf{M}}_\ell)$ for $\ell \in [0, L] \setminus \{\ell^\star\}$,
- – $\sigma^\star \leftarrow \mathsf{Sign}(\mathsf{sigk}^\star, (\mathsf{C}_{(\ell\|\mathtt{id}^\star,\mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star),\mathsf{verk}^\star)})_{\ell\in[0,L]})$.

Finally, $\mathcal{B}_{\ell^\star}$ returns $((\mathsf{C}_{(\ell\|\mathtt{id}^\star,\mathsf{T}_{[\ell-1,0]}(\mathtt{t}^\star),\mathsf{verk}^\star)})_{\ell\in[0,L]}, \sigma^\star, \mathsf{verk}^\star)$ to $\mathcal{A}_{\ell^\star}$ as an HKIBE challenge ciphertext.

Based on the same observation as in Theorem 2, the challenge ciphertext is properly distributed by implicitly setting

- – $\mathsf{M}_\ell = \widehat{\mathsf{M}}_\ell$ for $\ell \in [0, L] \setminus \{\ell^\star\}$,
- – $\mathsf{M}_{\ell^\star} = \mathsf{M}_b^\star \bigoplus_{\ell\in[0,L]\setminus\{\ell^\star\}} \widehat{\mathsf{M}}_\ell$ for $b \in \{0, 1\}$.

After $\mathcal{B}_{\ell^\star}$ receives b' from $\mathcal{A}_{\ell^\star}$, $\mathcal{B}_{\ell^\star}$ returns $\beta' := b'$ as its own guess to $\mathcal{C}$.

As we already observed, $\mathcal{B}_{\ell^\star}$ perfectly simulates the adaptive security game against $\mathcal{A}_{\ell^\star}$. Since the probability that β' is a correct guess is the same as that of b', $\mathcal{B}_{\ell^\star}$'s advantage is $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}_{\ell^\star}}(\lambda) = \mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+2,\mathcal{B}_{\ell^\star}}(\lambda)$ if $\mathcal{A}_{\ell^\star}$ does not query any decryption query that includes valid $\mathsf{verk}^\star$. Thus, we set $\mathcal{B} := (\mathcal{B}_0, \dots, \mathcal{B}_L)$ and have $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda) = \sum_{\ell^\star\in[0,L]} \mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}_{\ell^\star}}(\lambda) \leq \sum_{\ell^\star\in[0,L]} \mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+2,\mathcal{B}_{\ell^\star}}(\lambda) = (L+1) \cdot \mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+2,\mathcal{B}}(\lambda)$. □

Proof of Theorem 3. Taken together, we have $\mathsf{Adv}^{\mathtt{HKIBE}}_{\Pi,L,\mathcal{A}}(\lambda) \leq (L+1) \cdot \mathsf{Adv}^{\mathtt{HIBE}}_{\Sigma,L+1,\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathtt{OTS}}_{\Gamma,\mathcal{F}}(\lambda)$. □

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

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.

[AKA$^+$19] Michael P Andersen, Sam Kumar, Moustafa AbdelBaky, Gabe Fierro, John Kolb, Hyung-Sin Kim, David E. Culler, and Raluca Ada Popa. WAVE: A decentralized authorization framework with transitive delegation. In *28th USENIX Security Symposium, USENIX Security'19*, pages 1375–1392, Santa Clara, CA, August 2019. USENIX Association.

[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.

[BCHK07] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen ciphertext security from identity based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.

[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.

[BLSV18] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous IBE, leakage resilience and circular security from new assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 535–564, Cham, 2018. Springer International Publishing.

[BP06] Mihir Bellare and Adriana Palacio. Protecting against key-exposure: strongly key-insulated encryption with optimal threshold. *Appl. Algebra Eng. Commun. Comput.*, 16(6):379–396, 2006.

[BW06] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer Berlin Heidelberg, 2006.

[BWY11] Mihir Bellare, Brent Waters, and Scott Yilek. Identity-based encryption secure against selective opening attack. In Yuval Ishai, editor, *Theory of Cryptography, TCC 2011*, volume 6597 of *LNCS*, pages 235–252. Springer Berlin Heidelberg, 2011.

[CDRW10] Sherman S.M. Chow, Yevgeniy Dodis, Yannis Rouselakis, and Brent Waters. Practical leakage-resilient identity-based encryption from simple assumptions. In *ACM Conference on Computer and Communications Security, CCS 2010*, CCS '10, pages 152–161, New York, NY, USA, 2010. ACM.

[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.

[CHK07] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. *Journal of Cryptology*, 20(3):265–294, Jul 2007.

[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.

[Cis14] The internet of things reference model. Technical report, Cisco, 2014.

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

[DG17a] 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.

[DG17b] Nico Döttling and Sanjam Garg. Identity-based encryption from the Diffie-Hellman assumption. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference*, volume 10401 of *Lecture Notes in Computer Science*, pages 537–569. Springer, 2017.

[DK05] Yevgeniy Dodis and Jonathan Katz. Chosen-ciphertext security of multiple encryption. In Joe Kilian, editor, *Theory of Cryptography*, volume 3378, pages 188–209. Springer Berlin Heidelberg, 2005.

[DKXY02] Yevgeniy Dodis, Jonathan Katz, Shouhuai Xu, and Moti Yung. Key-insulated public key cryptosystems. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques*, volume 2332 of *Lecture Notes in Computer Science*, pages 65–82. Springer, 2002.

[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.

[ETW20] Keita Emura, Atsushi Takayasu, and Yohei Watanabe. Adaptively secure revocable hierarchical ibe from k-linear assumption. *IACR Cryptology ePrint Archive*, 2020:886, 2020.

[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.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS ’06, pages 89–98, New York, NY, USA, 2006. Association for Computing Machinery.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer Berlin Heidelberg, 2002.

[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.

[HHSI05] Yumiko Hanaoka, Goichiro Hanaoka, Junji Shikata, and Hideki Imai. Identity-based hierarchical strongly key-insulated encryption and its application. In Bimal K. Roy, editor, *Advances in Cryptology - ASIACRYPT 2005, 11th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings*, volume 3788 of *Lecture Notes in Computer Science*, pages 495–514. Springer, 2005.

[HL02] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, pages 466–481, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[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.

[JR13] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security*, volume 8269 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2013.

[KHA$^+$19] Sam Kumar, Yuncong Hu, Michael P Andersen, Raluca Ada Popa, and David E. Culler. JEDI: Many-to-many end-to-end encryption and key delegation for IoT. In *28th USENIX Security Symposium, USENIX Security 19*, pages 1519–1536, Santa Clara, CA, August 2019. USENIX Association.

[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.

[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*, pages 153–183, Cham, 2020. Springer International Publishing.

[LRW11] Allison Lewko, Yannis Rouselakis, and Brent Waters. Achieving leakage resilience through dual system encryption. In Yuval Ishai, editor, *Theory of Cryptography*, volume 6597 of *Lecture Notes in Computer Science*, pages 70–88. Springer Berlin Heidelberg, 2011.

[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.

[ML19] 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.

[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.

[SE15] 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.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer Berlin Heidelberg, 2005.

[SW18] Junji Shikata and Yohei Watanabe. Identity-based encryption with hierarchical key-insulation in the standard model. *Designs, Codes and Cryptography*, Jun 2018.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. 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 114–127. Springer, 2005.

[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.

[WLC+08] Jian Weng, Shengli Liu, Kefei Chen, Dong Zheng, and Weidong Qiu. Identity-based threshold key-insulated encryption without random oracles. In Tal Malkin, editor, *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, Proceedings*, volume 4964 of *Lecture Notes in Computer Science*, pages 203–220. Springer, 2008.

[WLCM06] Jian Weng, Shengli Liu, Kefei Chen, and Changshe Ma. Identity-based parallel key-insulated encryption without random oracles: Security notions and construction. In Rana Barua and Tanja Lange, editors, *Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Proceedings*, volume 4329 of *Lecture Notes in Computer Science*, pages 409–423. Springer, 2006.

[WS16] Yohei Watanabe and Junji Shikata. Identity-based hierarchical key-insulated encryption without random oracles. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Proceedings, Part I*, volume 9614 of *Lecture Notes in Computer Science*, pages 255–279. Springer, 2016.

[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.

A Overview of the Bug in the Security Proof in [SW18]

The SW18 scheme [SW18] is flawed due to improper handling of dual system encryption [Wat09]. We start from the basic concept of the dual system encryption and observe the bug in the security proof in [SW18]. Note that the earlier version [WS16] also contains the same bug. We follow the notations and terminologies used in the main body.

Dual System Encryption. Dual system encryption is one of the well-known techniques to prove adaptive security of plain HIBE, and utilizes two kinds of distributions for ciphertexts and secret keys. One is distributions that appear in a construction, called the *normal distribution*, and the other is distributions that only appear in the security proof, called the *semi-functional distribution*. Although normal secret keys can decrypt both normal and semi-functional ciphertexts, semi-functional secret keys cannot decrypt semi-functional ciphertexts. During the security proof, we first change a challenge ciphertext to be semi-functional, then change *all* secret keys queried by an adversary to be semi-functional one by one. Essentially, the reason why the changes succeed is that, in plain HIBE, the adversary is not allowed to query the challenge identity or its ancestors. Since the adversary completely loses decryption capability after the changes, it is easy to replace the underlying plaintext of the challenge ciphertext with a random one without being noticed by the adversary.

The Overview of the Bug. To the authors' credit, the SW18 scheme is the same as our first construction instantiated by [RS14]. It means that the flaw is due to the proof methodology they employed, not their construction. The proof of [SW18] employs dual system encryption in a naïve manner; the challenge ciphertext is first changed to be semi-functional, then all helper keys and decryption keys are changed to be semi-functional one by one. However, the latter changes failed in the sense that the changed keys do not distribute properly as the authors expected or an adversary is able to detect the changes. What is essential here is that the HKIBE adversary is allowed to make key-insulation queries on the challenge identity $\mathtt{id}^\star$, whereas the HIBE adversary does not make any secret-key reveal queries on $\mathtt{ID}^\star$. We explain the details below.

HIBE Proof Using Dual System Encryption. If the adversary is allowed to make a query on the prefix of the challenge identity, the simulation fails since the randomness of the secret key for the query would be correlated to the randomness of the challenge ciphertext. Nevertheless, such a query is not allowed during the security game. Therefore, the dual system encryption goes through since the reduction algorithm does not need to create and reveal information on $\mathtt{ID}^\star$ except for the challenge ciphertext $\mathsf{C}_{\mathtt{ID}^\star}$; the randomness for $\mathsf{C}_{\mathtt{ID}^\star}$ is independent of randomness for any secret keys from the viewpoint of the adversary.

Proof of [SW18]. The HKIBE challenge ciphertext $\mathsf{ct}_{\mathtt{id}^\star,\mathtt{t}^\star}$ can be regarded as a ciphertext for $(\mathtt{id}^\star, \mathsf{T}_{[L-1,0]}(\mathtt{t}^\star))$ in the HIBE scheme proposed by [RS14]. The randomness of the challenge ciphertext $\mathsf{ct}_{\mathtt{id}^\star,\mathtt{t}^\star} := \mathsf{C}_{(\mathtt{id}^\star,\mathsf{T}_{[L-1,0]}(\mathtt{t}^\star))}$ is correlated to the randomness of level-ℓ helper keys $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ for the key-insulation query on $(\mathtt{id}^\star, \mathtt{t}, \ell)$ such that $\ell > \ell^\star$. This is due to the form of $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$: roughly speaking, the level-ℓ helper key $\mathsf{hk}^{(\ell)}_{\mathtt{id},\mathsf{T}_\ell(\mathtt{t})}$ is the HIBE secret key of [RS14] for a hierarchical identity $(\mathtt{id}, \mathsf{T}_{[L-1,\ell]}(\mathtt{t}))$ masked with a random group element. The adversary is allowed to make the key-insulation query $(\mathtt{id}^\star, \mathtt{t}^\star, \ell\ (> \ell^\star))$, and hence can obtain the HIBE secret key of [RS14] for a hierarchical identity $(\mathtt{id}^\star, \mathsf{T}_{[L-1,\ell]}(\mathtt{t}^\star)) \in \mathsf{prefix}^+((\mathtt{id}^\star, \mathsf{T}_{[L-1,0]}(\mathtt{t}^\star)))$ (with a random mask). As mentioned above, in the standard HIBE game, a query on such a hierarchical identity, i.e., the prefix of the challenge identity, is not allowed and spoils dual system encryption. The authors seemed to expect that the random mask of $\mathsf{hk}^{(\ell)}_{\mathtt{id}^\star,\mathsf{T}_\ell(\mathtt{t}^\star)}$ would help to resolve the issue, however, it does not; the query still leads to the correlation. Thus, the simulation fails when the adversary obtains both $\mathsf{ct}_{\mathtt{id}^\star,\mathtt{t}^\star}$ and $\mathsf{hk}^{(\ell)}_{\mathtt{id}^\star,\mathsf{T}_\ell(\mathtt{t}^\star)}$ for $\ell \in [\ell^\star + 1, L]$.